A Systematic Method for Performance Analysis of SOA Applications

Marzieh Asgarnezhad, Ramin Nasiri, and Abdollah Shahidi

Abstract—The successful implementation of Service-Oriented Architecture (SOA) is not confined to Information Technology systems and required changes of the whole enterprise. In order to adapt IT and business, the enterprise requires adequate and measurable methods. The adoption of SOA creates new problem with regard to measuring and analysis the performance. In fact the enterprise should investigate to what extent the development of services will increase the value of business. It is required for every business to measure the extent of SOA adaptation with the goals of enterprise. Moreover, precise performance metrics and their combination with the advanced evaluation methodologies as a solution should be defined. The aim of this paper is to present a systematic methodology for designing a measurement system at the technical and business levels, so that: (1) it will determine measurement metrics precisely (2) the results will be analysed by mapping identified metrics to the measurement tools.

Keywords—Service-oriented architecture, metrics, performance, evaluation.

I. INTRODUCTION

SOA defines an architecture that is related to but still distinct from its predecessors [1]. SOA is an architectural approach to define and provision the IT infrastructure that allows different applications to data exchange and participate in business processes regardless of the operating systems and programming languages underlying those applications [2]. It is said what has not been measured can not be improved. As the enterprises move to SOA they encounter, the challenge of turning the abstract advantages to tangible and measurable ones [3]. If SOA is a tool to achieve a goal, then what are the goals? How do enterprises evaluate the development and performance of SOA in the field of business?

Contrary to common belief, performance evaluation of SOA is a complicated approach. An evaluation technique and selecting a Well-defined metric are two key steps in all performance evaluation projects [4]. Well-defined metric here means a metric that is precise and practical enough to effectively diagnose the cause of the performance problem [5]. Like an artist, each analyst has a unique style. Given the same problem, two analysts may choose different performance

Marzieh Asgarnezhad is with Islamic Azad University, Kashan Branch, Iran (e-mail: asgarnezhad77@yahoo.com).

Ramin Nasiri (e-mail: r_nasiri@iauctb.ac.ir) and Abdollah Shahidi (e-mail: a_shahidi@iauctb.ac.ir) are with Islamic Azad University, central Tehran branch, Iran.

metrics and evaluation methodologies [4]. However, current works on performance evaluation are not precise and comprehensive enough to be applied in effective diagnosis especially regarding the dynamism of SOA [5].

The aim of this paper is to focus on Performance evaluation of SOA and measurement SOA adaptation with enterprise goals. The proposed method evaluates SOA in two technical and business levels. To analyse in the technical level are used Process mining techniques. The idea of process mining is to discover, monitor and improve real processes by extracting knowledge from event logs. Thus process mining requires the availability of an event log. Events logs may originate from all kinds of systems ranging from enterprise information systems to embedded systems. [6]

Therefore SOA metrics is combined with advanced evaluation tools in order to improve the performance evaluation and analysis of the SOA adaptation. To achieve this aim, the present researchers discuss precise measurement metrics, measurement strategy (the hierarchy of measurement and the management process), methods and tools of measurement, their relations and application in the form of designing a measurement and management system.

The paper continues with: giving a brief overview on the related work in section II, defining a set of metrics for measuring SOA performance in section III and describing the proposed framework to measure and analysis in section IV. To show the practicability and usefulness of the metrics, researchers apply the metrics to process of payment of driving fines in section V. Finally, researchers conclude the paper and point out future research directions in Section VI.

II. RELATED WORKS

In regard to evaluating SOA approaches have been presented: (1) Approaches that argue in favour of the introduction of a SOA and are mostly presented by consulting companies [7][8]. These approaches often lack scientific methodologies. (2) Approaches which primarily focus on IT aspects of SOA evaluation. These approaches define and evaluate quality attributes which may refer to single applications [9] or to the entire IT architecture [10][11].

Another work is to Present the goal/question/metrics method to identify evaluation metrics and determining the status of SOA in an enterprise architecture (EA) context [12].

III. MEASURING METRICS

As the enterprises struggle to progress, their success will be impossible without effective enterprise metrics. But what is to be measured? And are all measurements important? Table I summarizes the proposed measurement metrics framework in terms of metric name, abbreviation, metrics type, and description. Some of SOA measurements should be done similar to other common IT measurements such as budget and time, availability and system response time. The measurement metrics are interrelated and should be controlled at the start of SOA adoption in the enterprise. For example calculating ROI[1] precisely without controlling the costs from the beginning is impossible, or in order to determine the value of services the following factors should be considered:

- the Number of Services that are available for Reuse (NSR)
- the Degree of Reuse (DR)
- the Complexity of service (C)

The formula which determines the service value looks like the following formula. And if this formula is applied to the entire domains of the enterprise, the amount of the reuse of services will directly depend on the saved cost.

$$Value = (NSR*DR) * C$$

IV. METHODOLOGY

In this section the researchers are trying to propose their own method. As Fig. 1 depicts, the proposed framework is based on four major columns: *Application, ProM Import Framework[2], ProM Framework[3] and Measuring Metrics.* It uses *ProM* and *ProM Import* to analyse the system. Before using these tools one should define measuring metrics in a very clear and precise manner. It is very important to have metrics that are well defined and meaningful. ProM needs data which can be extracted from the logs created by the designated Application. To fulfil this, some changes should be done in the Application.
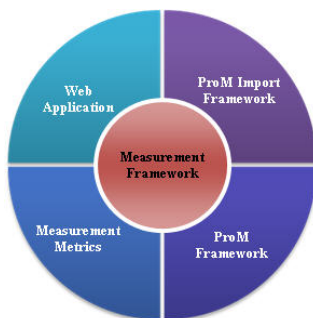

Fig. 1 The proposed measurement framework

The measurement is done in two technical and business levels so that the SOA metrics determined in the section III is combined with advanced evaluation tools in order to improve the performance evaluation and analysis of the SOA

adaptation. To analysis at technical level, as Fig. 2 shows, all event logs produced by executing the web application are transferred to the ProM Import framework using the CPN tools plug-in and finally SOA is analysed at the technical level by mapping the determined technical metrics to the ProM.
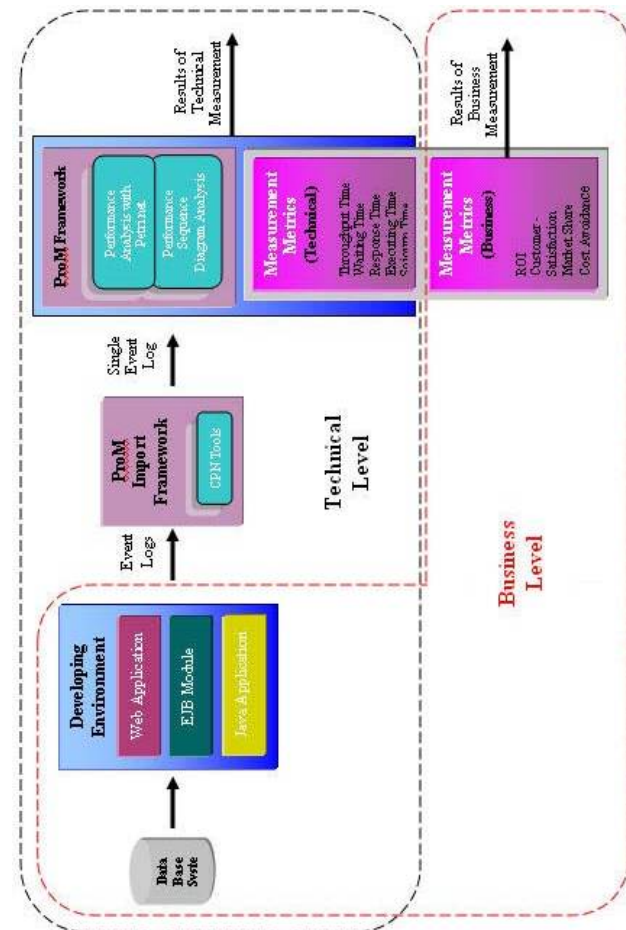

Fig. 2 The visual description of the methodology in two technical and business levels

An event log consists of several instances or cases, each of which may be made up of several audit trail entries. [6] Each audit trail entry records task name, event type, originator and time stamp. This information is defined by the MXML[4] format which is a generic XML-based format suitable for representing and storing event log data. The structure of an MXML document is depicted in Fig. 3 [16].

Information systems typically support logging capabilities that register what has been executed in the organization. These produced logs usually contain data about cases (i.e. process instances) that have been executed in the organization, the times at which the tasks were executed, the persons or systems that performed these tasks, and other kinds of data.

---

[1] Return of Investment

[2] Process Mining Import Framework

[3] Process Mining Framework

[4] Mining-XML

TABLE I
MEASURING METRICS [13][14][15]

Metric	Abbr.	Type	Description
Service Response Time	SRT	Technical	Time when consumer finishes sending request to the service – Time when consumer starts receiving response from the service
Service Turnaround Time	STA	Technical	$STA = \sum_{i=1}^{m} STR_i + \sum_{j=1}^{n} TH_j$
Throughput	TP(SRV)	Technical	Number of Completed Service Requests / Unit of Time
	TP(BP)	Technical	Number of Completed Business Process Requests/ Unit of Time
Throughput Ratio		Technical	TP / Required TP *(Numerator is the currently measured TP which can be can be either TP(SRV) or TP(BP). Denominator is the required TP that is defined in the SLA.)*
Service Response Time Ratio		Technical	Sum of SRT / Required Sum of SRT, *n; number of interactions in a service. Numerator is the currently measured summation of the n SRTs. Denominator is the required summation of the n SRTs that is defined in the SLA.*
Throughput time		Technical	The throughput time of the process instances.
Waiting time		Technical	time that passes from the (full) enabling of a transition until its firing, i.e. time that a token spends in the place waiting for a transition (to which the place is an input place) to fire and consume the token. (in the Petri net)
			The time between the moment at which the activity (service) is scheduled and the moment at which execution of the activity is started. (Time between a schedule and a start event of the activity).
Synchronization time		Technical	The time that a token spends in a place, waiting for the transition to be fully enabled. (in term of Petri net)
Think Time	TH	Technical	Time between the end of a response generated by a service and the beginning of an end user's response
Think Time Total	TH_{Total}	Technical	Total time spent for thinking within one turnaround
Sojourn time		Technical	The total time a token spends in a place during a visit (in the Petri net) (Waiting time + Synchronization time).
			Time between the scheduling of an activity (service) and the time it finishes execution (Time between a schedule and a complete event).
Execution time		Technical	Time in which an activity (service) is actually executed. Which is the time between the moment at which the activity is started and the time at which it is completed, without possible time spend in a state of suspension. (Time between a start and a complete event of an activity, without the time spend in between all suspend and resume pairs that occurred in between).
Market Share		Business	Market share is the portion or percentage of sales of a particular product or service in a given region that are controlled by a company. *For example, if there are 100 widgets sold in a country and company A sells 43 of them, then company A has a 43% market share.*
Time-to-Market	TTM	Business	The length of time it takes to get a service from idea to marketplace. *There are many definitions for the start and the end of the TTM period. The start of period might be when the product concept is approved or when the project is fully staffed. The end of the period is when engineering transfers the product to manufacturing. Also The end of the period can be when the first copy of the new product is shipped or when a customer buys it or will often be defined in terms of reaching a certain production volume, such as a million units per month.*
Integration Time Savings		Business	Since the integration cost for each ongoing or future project can only be estimated, a standard reuse factor can be applied the service build cost. Eighty percent is the typical number used in these situations.
Number of Services Created	NSC	IT Pervasiveness	The actual number of new services created.
Number of Services Reused	NSR	IT Pervasiveness	The number of reusable services is the actual number of new services created, or, existing services abstracted, that are potentially reusable from system to system.

Service Reuse Ratio		IT	Number of Services Reused / The total number of services
ROI	SoROI	Financial	SOA ROI (%) = SOA ROI ($) All SOA-Related Investments SOA ROI ($) = Cost Savings/Efficiencies Achieved - All SOA-Related Investments
			Total SOA - related investment = all the costs related to building the existing set of services + the SOA software purchases such as ESB, RegRep, service management tools, etc.
Cost Savings/Avoidance		Financial	Service Cost Avoidance = Service Build Cost - Project's Service Integration Cost
			Service Build Cost = Initial Service Build Cost + Cost of all Subsequent Changes
Reduction in project and maintenance costs		Financial	To calculate the entire project's cost avoidance amount, simply add the cost avoidance for all the services being leveraged. To forecast the total potential cost avoidance at any point of time, multiply the number of times each service is envisioned to be leveraged by its build cost and add it all together.
Service Availability/Uptime		Quality Management	Service availability is the percentage of time a service is usable by the end user.
Lines of code		Quality	Lines of code (both consumer-side and producer-side)
Recovery time		Timeliness	Recovery time for each service
Degree of Service Reuse	DR	Pervasiveness Project	The degree of reuse from system to system is the number of times you actually reuse the services. We look at this number as a percentage.
Revenue per Service		Corporate	This number captures both the value an organization delivers, and the productivity achieved based on the value created by the service base *(Not all services will generate revenue)*
Service Vitality Index	SoVI	Corporate	Vitality Index is the amount of revenue from new services over the last 12 months as a proportion of total service revenue.
Number of new services generated and used as a percentage of total services		Management	Number of New Services Generated and Used as a Percentage of Total Services - Not only does this drive the total cost of service development up, but it also reduces the average revenue per service, indicating poor service development productivity.
Mean Time to Service Development	MTTSD	Management	MTTSD provides a statistical measure, along with a range of certainty, of the average time to stand up a service. Reducing MTTSD is a key benefit of SOA governance and indicates business agility.
Mean Time to Service Change	MTTSC	Management	MTTSC can point out those services that have been poorly created and costing the organization in terms of effort and lost opportunity costs.
Cost of not using or stopping a service		Project	A well-designed SOA implementation has low shutdown or switching costs.
Service Complexity as measured through cyclomatic complexity	C	Service Development	The cyclomatic complexity of a service is the single measure that will determine if your service is testable and maintainable. Studies have shown that services with cyclomatic complexity greater than 50 are not testable.
			The complexity of the services is the number of functions or object points that make up the service.
Service Value	V	IT Pervasiveness	Value = (NSR*DR) * C

These logs are the starting point for process mining, and are usually called event logs. The ProM framework is based on interpreting event log data in the MXML[5] format and it is an open-source tool to support the development of process mining plug-ins. [17][18]
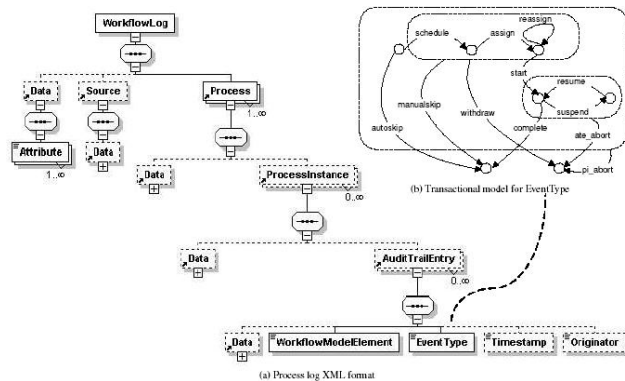

Fig. 3 Schema of the MXML format

ProM framework enables rapid development of new algorithms and techniques by means of plug-ins [6]. A plug-in is basically the implementation of an algorithm used in the process mining area. Such plug-ins can be added to the framework relatively easily. This paper does not tend to cover all measuring metrics because firstly many metrics are not too obvious to define thus can not be measured easily, secondly such a job needs a lot of time and practice. So as a proof of correctness and practicability of the framework, this paper uses two plug-ins regarding measuring performance issue. One can use other existing plug-ins or develop his/her own plug-ins to measure different metrics.

The analysis of SOA at business level goes on regarding the determined business metrics in the proposed framework. Therefore, as Fig. 2 shows, applying tools for analysis process will be different. Those metrics such as some of business metrics that are not easy to define and implement could be measured manually.

A. Two plug-ins regarding Performance Analysis

Researchers use two plug-ins, *Performance Analysis with Petri net* and *Performance Sequence Diagram Analysis* to analysis performance in the technical level. This plug-ins focuses on analysing time-related aspects of the process instances [19].

1. Performance Analysis with Petri net

The focus of the plug-in is to provide key performance indicators, which can be summoned in an intuitive way. Here is the list of Performance metrics which this plug-in provides:

- Process metrics:
  - Average Throughput time
  - Minimum Throughput time

  - Maximum Throughput time
- Place metrics:
  - Waiting time
  - Synchronization time
  - Sojourn time
- Activity metrics:
  - Waiting time
  - Execution time
  - Sojourn time

In addition to the main log produced by CPN tools plug-in, the mined model of Petri net as input is also required. For this, the event log is replayed in the Petri net model of the use case generated using the alpha algorithm. [20] Therefore, at the first step the mined model is imported and at the second one the imported model and the main log are analysed in the analysis plug-in.

Also the plug-in can answer the following questions [19]:

- What are the routing probabilities for each split/join task?
- What is the average/minimum/maximum throughput time of cases?
- What is the average service time for each task?
- How much time was spent between any two tasks in the process model?
- Where are the bottlenecks in the processes? How severe are these bottlenecks?

2. Performance Sequence Diagram Analysis

This plug-in can answer the above questions. Filter options can be use to filter the log so that only cases with throughput time superior to average throughput time are kept. [19]

- How often each case happened in the log?
- Where are the bottlenecks in the processes?
- How severe are these bottlenecks?

Also the following questions are answered by applying the analysis plug-in *Performance Sequence Diagram Analysis* in combination with the *Performance Analysis with Petri net*. The procedure to do so have two steps: (1) to produce PNML model with the process instances with throughput times superior to average throughput time, (2) to apply *Performance Analysis with Petri net* to discover the critical sub-paths for these cases.

- Which paths take too much time on average?
- How many cases follow these routings?
- What are the critical sub-paths for these routes?

V. A CASE STUDY: DRIVING FINES' SYSTEM

In this section, researchers apply the designed framework to measure some metrics in the *driving fines' system* to show the practicability and usefulness of the framework. This paper first briefly explains the process of *payment of driving fines* and then describes the environment for measuring the system.

A. Scenario and Environment Description

After a fine is entered in the system, it is recorded in the system and then the bill will be sent to the driver in term of

fine type. The recorded fines are checked and in case the bill is not paid within one month, the driver will receive a reminder. When the bill is paid, the case is archived. The driver goes to the bank to pay the bill. The bank system sends the payment number back to the system and the complete information regarding the fine is archived in the system.

The case study has been implemented in the NetBeans IDE equipped by Web & Java EE 5 and the database has been implemented in MySQL. EJB technology can provide business methods as web service. Both of EJB and Web Service Application have developed in the Java EE servers. In the case study it have been tried to use combination of these software components of netbeans to get the goals of SOA. Moreover as Fig. 4 shows the event logs consists of task name, event type, time stamp and originator. This information is used for performance evaluation.

Fig. 4 Overview of an event log in the case study

B. Measuring Metrics

In this section, researchers apply the selected metrics; Throughput time, Waiting time, Synchronization time and Sojourn time and show the value of the metrics computation.

As Fig. 5 shows, the central panel specifies the bottlenecks and the routing probabilities. The colour of the places in the Petri net indicates where in the process the jobs of this use case spend most time. In the context of our example, "Place3" with red margin is not bottleneck but it has medium waiting time. Table II shows values of the metrics computation.

On the other hand since the results indicate that the cases take on average 0.92 *hours* to be completed, it would be interesting to analyse what happens for the cases that take longer than that. The procedure to do so has the following steps:

1. *Performance Analysis with Petri net plug-in*

2. *Performance Sequence Diagram Analysis* on the whole event log.

Fig. 6 shows result of the Analysis.

3. To filter the log so that only cases with throughput time superior to 0.92 hours are kept.

4. *Performance Analysis with Petri net plug-in* with PNML model but this time it is linked to the filtered cases.
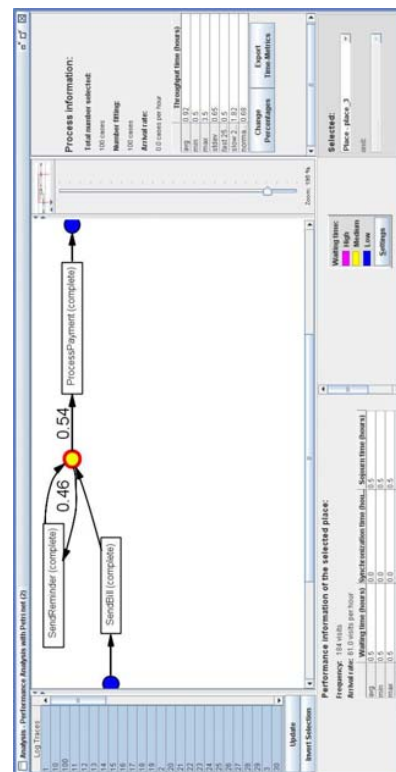

Fig. 5 Screenshot of the analysis plug-in Performance Analysis with Petri net

TABLE II
VALUES OF THE METRICS COMPUTATION (TIMES MEASURED IN HOURS)

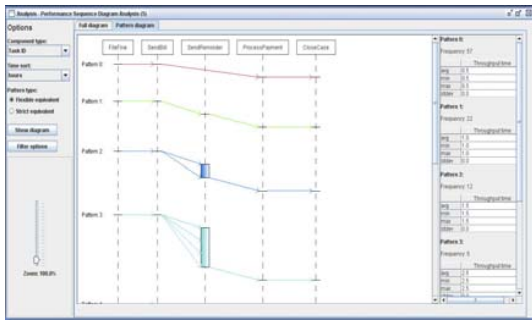| (Process metric)			SendReminder & ProcessPayment (Activity metrics)			Place3 (Place metrics)		
Average Throughput time	Minimum Throughput time	Maximum Throughput time	Waiting time	Execution time	Sojourn time	Waiting time	Synchronization time	Sojourn time
0/92	0/5	3/5	-	-	0/5	0/5	0/0	0/5

Fig. 6 Screenshot of the analysis plug-in Performance Sequence Diagram Analysis

VI. CONCLUSION

This paper proposed a systematic method for designing a measurement system for performance analysis of OSA. At the first step the measurement framework including the precise definition of measurement metrics was determined. Then, in order to apply the proposed method in this framework, the metrics where classified into technical, business and etc. categories. At the second step the analysis method was presented in the technical and business levels. To fulfil this aim, the SOA metrics framework was combined with advanced evaluation tools to improve the performance evaluation and SOA adaptation analysis.

Since many metrics are not easy to determine, thus they can not be measured easily. Therefore as a proof of correctness and practicability of the framework, researchers use two plug-ins of Prom framework that regarding measuring performance issue.

REFERENCES

[1] Erl, T., Service-Oriented Architecture: Concepts, Technology, and Design, Prentice Hall, 2005.
[2] Newcomer, E. and Lomow, G., Understanding SOA with Web Services, Addison Wesley Professional, 2004.
[3] Hubbard, D. W., How to Measure Anything Finding the Value of ''Intangibles'' in Business, Wiley, 2007.
[4] Jain, R., the art of computer systems performance analysis: Techniques, for Experimental Design, measurement, Simulation, and Modeling, Wiley, 1984.
[5] Her, J. S., Choi, S. W., Oh, S. H., and Kim, S. D., A Framework for Measuring Performance in Service-Oriented Architecture, Proceedings of the NWeSP 2007, pp. 55-60, 2007.
[6] Song, M. and van der Aalst, W. M. P., Towards Comprehensive Support for Organizational Mining, Decision Support Systems, v.46 n.1,pp.300-317, 2008.
[7] DiMare, J., Service oriented architecture: A practical guide to measuring return on investment, http://www-935.ibm.com/services/de/bcs/pdf/2007/soa_practical_guide.pdf, 2006.
[8] Barnes, M., Applied SOA: Measuring Business Value, http://www.gartner.com/teleconferences/attributes/attr_150040_115.pdf, 2006.
[9] Johnson, P., et al., Extended Influence Diagrams for Enterprise Architecture Analysis, in the Proceedings of the EDOC 2006, pp.3-12, 2006.
[10] Vasconcelos, A., et al.: An information system architectural framework for enterprise application integration, in the Proceedings of the HICSS 2004, pp. 225-233, 2004.
[11] Khaddaj, S., Horgan, G., The Evaluation of Software Quality Factors in Very Large Information Systems, Electronic Journal of Information Systems Evaluation, pp. 43–48, 2004.
[12] Aier, S., Ahrens, M., Stutz, M., Bub, U., Deriving SOA Evaluation Metrics in an Enterprise Architecture Context, in the Proceedings of the ICSOC 2007, pp. 224 – 233, 2009.
[13] Smith, J., 10 Measures for Successful SOA Implementations, http://soa.sys-con.com/node/631831, 2008.
[14] McKendrick, J., Justifying SOA: 12 key metrics to keep tabs on, http://blogs.zdnet.com/service-oriented/?p=1950, 2009.
[15] http://www.processmining.org/
[16] A.K. Alves de Medeiros and C.W. Gunther, Process Mining: Using CPN Tools to Create Test Logs for Mining Algorithms, www.daimi.au.dk/CPnets/workshop05/cpn/.../AnaKarlaAlvesdeMedeiros.pdf, 2005.
[17] Medeiros, A. K. Alves de, and Weijters, A. J. M. M., ProM Framework Tutorial, http://prom.win.tue.nl/research/wiki/_media/tutorial/promtutorialv2.pdf?id=tutorials&cache=cache, 2008.
[18] Günther, C. W., van der Aalst, W. M. P., A Generic Import Framework For Process Event Logs, Proceedings of the Business Process Management Workshops Workshop on Business Process Intelligence (BPI 2006), volume 4103, pp. 81-92, Springer LNCS, 2006.
[19] Van der Aalst, W. M. P., Stein, S., Zoeller, S. and et. al, Semantics Utilized for Process management within and between Enterprises, www.ip-super.org/res/Deliverables/M12/D2.2.pdf, 2007.
[20] Kannan, V., van der Aalst, W.M.P. and Voorhoeve, M., Formal Modeling and Analysis by Simulation of Data Paths in Digital Document Printers, http://is.tm.tue.nl/staff/wvdaalst/publications/p538.pdf, 2008.