Available online at www.sciencedirect.com

ScienceDirect

Procedia Technology 16 (2014) 11 – 21

Procedia Technology

CENTERIS 2014 - Conference on ENTERprise Information Systems / ProjMAN 2014 - International Conference on Project MANagement / HCIST 2014 - International Conference on Health and Social Care Information Systems and Technologies

Towards a process to guide Big Data based Decision Support Systems for Business Processes

Alejandro Vera-Baquero[a], Ricardo Colomo-Palacios[b,*], Owen Molloy [c],

[a]Universidad Carlos III de Madrid, Av. Universidad 30, Leganes (Madrid) 28911, Spain
[b]Østfold University College, B R A Veien 4, Halden 1783, Norway
[b]National University of Ireland, Galway, Ireland

Abstract

Process performance improvement initiatives generally require the application of both knowledge management techniques and analysis tools to assist business users in decision making. Decision support systems (DSS) are a valuable asset to measure process performance; however, they require a vast amount of process performance data in order to support a valuable analysis with highest precision and accuracy. Moreover, this analysis needs to be attained in a timely manner in order to respond quickly to non-compliant situations. Existing process performance improvement initiatives lack of the appropriate methods and tools to give full support to business users. Contrarily to this, they are focused on a purely methodology perspective. We introduce a Business Process Improvement methodology for overcoming this limitation by integrating process improvement with big data based DSSs.

Keywords: Business Process Improvement; Business Performance Management, Big Data; Decision Support Systems.

* Corresponding author. Tel.: + 47 6921 5000; fax: + 47 6921 5002.
E-mail address: ricardo.colomo-palacios@hiof.no

doi:10.1016/j.protcy.2014.10.063

1. Introduction

Analysis of performance data with the aim of improving the performance and efficiency of enterprise business systems is an important part of running a competitive business. Decision Support Systems (DSS) are a valuable asset for analysts since they transform performance data into useful information, and in turn, such information is transformed into knowledge in order to support decision making. However, traditional BI and DSS tools require something more than the use of mere historical data and rudimentary analysis tools to be able to predict future actions, identifying trends or discovering new business opportunities [1]. Reducing the time needed to react to non-compliant situations can be a key factor in maintaining competitiveness. Real-time, low latency monitoring and analysing of business events for decision making is key, but difficult to achieve [2]. The difficulties are intensified by those processes and supply chains (typically, the most interesting cases) which entail dealing with the integration of enterprise execution data across organizational boundaries. Such processes usually flow across heterogeneous systems such as business process execution language (BPEL) engines, ERP systems, document management systems, etc. The heterogeneity of these supporting systems make the collection, integration and analysis of high volume business event data extremely difficult [3]. Big data scenarios are making the process even more complicated. With the amount of information available coming from a variety of sources in rapid way and with panoply of formats, the landscape of DSS has changed from its roots. In [2] we introduced a big-data based DSS that provides visibility and overall business performance information on distributed process. This DSS approach has the capability to enable business users to access performance analytics data efficiently in a timely fashion, availing of performance measurements on an acceptable response time basis. Integrating this approach with the methodology presented herein, we aim to put real BPI technology in hands of business users, thus leading analysts to gain a better understanding and control of their processes within and across organizations, while keeping aligned the control of their business operations with the process improvement activities. In short, this aim is to assist business users in sustaining a comprehensive process improvement program. In what follows, first the approach is presented and in second term a case study on the application of the process is depicted. Final section includes main conclusions and future work.

2. The approach

The methodology consists of five phases (see Fig. 1), in what follows phases are depicted.

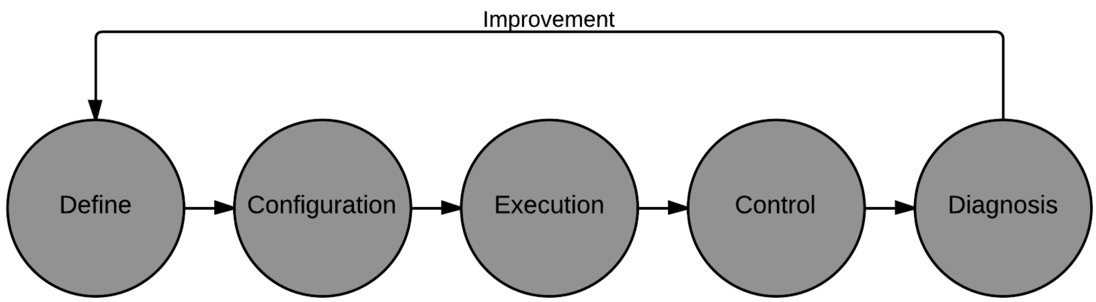

Fig. 1. Business process improvement lifecycle.

2.1 Define

The define phase deals with the identification of the distributed business process model along the large and complex supply chains. This phase consists in discovering and defining the process that we aim to improve. Likewise, the purpose of this phase is not only to identify and represent the business process that has a significant value for the organization, but also to have clear insight into the strategic management of the enterprise and a good understanding of the business goals being pursued. This will help the analyst to identifying the critical processes or activities that must be monitored. For identification of the process models, we use a method based on the tabular application development (TAD) methodology [4]. During this phase, a sequence of tasks must be undertaken in order to obtain a representation of the business process thereof. The define phase involves four steps: 1) the first step determines the scope and boundaries of the global business process (cross-organizational), 2) the second step identifies operational flows within each single organization; including interactions between operational units (inter-departmental), 3) the third step identifies the level of detail that the global business process will be broken down into (level of sub-activities), and 4) the last step deals with the development of process and activity tables.

2.1.1 *Define Phase – Scope and boundaries identification*.

This step consists in identifying the scope and boundaries of the global business process, and defining the global business process itself. In large and complex supply chains, there are a considerable number of business entities that are involved in the business process, such as Manufacturing, Sales, Stock, Logistic, Accounting, etc. The determination of these participants is crucial for establishing the boundaries of sub-processes and discovering key interactions between enterprises (cross-functional) or departments (inter-departmental), hereinafter *business nodes*.

2.1.2 *Define Phase – Definition of sub-processes, activities and sub-activities*.

In this step we have to iterate over each organizational node that has been identified in the previous step. The aim of this process is trying to discover sub-processes, activities and sub-activities associated to the global process identified in the previous step. As already stated, this BPI methodology is sustained by the big-data based DSS system widely discuss in [2]. This IT solution presents capabilities to monitor and query the structural and behavioural properties of business processes. Hence, it is required to gather the relevant properties that are part of the structure of the processes and activities, i.e. inputs and outputs, payloads, resources, costs, etc. Likewise, it is important to give especial emphasis on the input, outputs and payloads of processes and activities, as this information will be essential at further stages for establishing the link between inter-related processes.

2.1.3 *Define Phase – Determination level of detail within business processes*.

It is worth highlighting that "a process may itself be composed of a number of different sub-processes or activities which in turn may be decomposed into a set of smaller related tasks" [1]. There is no globally accepted limit on the number of levels, and depending on the nature of the business process and the specific requirements on process improvement endeavours, it may be necessary to monitor both high level and low level processes. This number of levels must be identified in this step. The greater the number of nested levels, the more cumbersome is the deployment of the DSS, and the more complex is the monitoring and analysis of the performance information. Therefore, it is important to determine the trade-off between the deployment costs, and the final value of such data. If the performance information of an activity or sub-activities at a given level is neither crucial nor relevant, then it might be better to leave them out of the analysis. Additionally, each business node may have its own level of detail per process or activity. According to [2], every business node (business analytics service units, hereinafter BASU) can perform the analysis of their own processes in isolation.

2.1.4 *Define Phase – Development of model tables*.

Once all sub-processes and activities have been identified, the next step is to model the business process in a tabular form. This methodology follows a business process model representation using tables because they are useful for representing the sequence of events, clear and easy to manage for business users [4], and simplifies the deployment of the DSS system in further stages. In this step we must create a table per business node, where each table is organized as follows: the first column defines the global business process definition of the business node. Consequently, this process is a sub-process of the cross-organizational process defined in the first step. The second column presents the activities grouped by processes; the third column represents the nested level of the activity by making a reference to the parent activity. The fourth and last column lists a set of properties in the form of key value pairs. Table 1 is a very simplified representation of the business process model. It has been designed to be as simple as possible on purpose. The execution sequence of instances will be determined by the DSS in runtime by making use of the information provided in such a model table, and the behavioural information will be calculated at the execution phase. The following table describes a sample of a business process for a single node A.

Table 1. Sample process model table for business node A

Process	Activity	Activity Parent	Properties
1#P_1	1#A_1		$Prop_1$
	2#A_2	A_1	$Prop_1$,$Prop_2$,$Prop_3$
	3#A_3	A_1	$Prop_1$,$Prop_2$
2#P_2	4#A_4		$Prop_1$
	5#A_5	A_4	$Prop_1$,$Prop_4$

2.2 Configuration

This phase intends to prepare the analytical environment for receiving structural event data from the operational systems that will feed the DSS for later analysis. Therefore, this phase is critical for the global success of the performance analysis, and equally important in the successful implementation of the DSS. In this stage we aim to identify software boundaries and inter-departmental processes within business nodes. Likewise, the selection of event data format, and the determination of instance correlations are also undertaken. Finally, we implement software listeners along with a selection metrics and their threshold values. The internal steps of this phase are outlined below.

2.2.1 Configuration Phase – Business nodes provisioning and software boundaries identification.

In this step we must provision a BASU component [2] per business node identified in the *Define* phase. The number of nodes may vary depending on three main factors: 1) the nature of the business process that we intend to analyse, 2) the performance of the DSS, or 3) security issues due to the data sharing between the BASU unit and the GBAS (Global Analytic Business Service) component (see [2] for further details). The DSS described in [2] allows individual companies in a supply chain to own and manage their own data. Provided data sharing was not an issue, or if a single secure data store was acceptable to all process owners, we can provide one BASU unit per business node. Otherwise, it is possible to breakdown a business node into smallest business units, and provisioning a unique BASU component per unit. This approach is also valid for performance reasons. Afterwards, and as part of the business nodes provisioning step, it is necessary to load the process model tables into every corresponding BASU unit. Once we have provisioned all business nodes, we must identify the software boundaries within each business node. This will give us an insight about the software requirements on source systems when implementing the listener in a further step. Furthermore, these software boundaries are normally linked to inter-departmental sub-processes. Therefore, the use of the model tables developed in the Define phase, are very useful to discover technological requirements for those processes that flow across heterogeneous systems.

2.2.2 *Configuration Phase – Selection of event data format.*

The event format data that will feed the DSS must be decided in this step. There are several alternatives that are discussed in [5], being XES, MXML and BPAF the most popular and accepted event format for process mining. This selection will depend on the business analysts; whether they consider to be useful or not to maintain interoperability of the event logs with other process mining tools and techniques besides the DSS. Within the DSS context, the legacy listener software may emit the event information to different endpoints depending on the message format provided. Currently, the platform supports a variety of widely adopted formats for representing event logs such as XES, MXML [5] [6] or even extended BPAF [1][6]. Every BASU unit transforms and correlates their own events by querying the event repository for previous instances. The DSS event correlation algorithm uses the event data provided in the format message, and thus this correlation data is crucial for the accuracy and quality of the performance data.

2.2.3 *Configuration Phase – Determine event correlation data.*

This step consists in determining which part of the message payload will be used to correlate instances. The term *instance correlation* refers to the unique identification of an event for a particular process instance or activity during execution. For example, for an order process, the *order number* may be used to match the start and end of the event sequence in the timeline. The event correlation is on the execution critical path, and they must occur timely with their own process definitions. Without the ability to correlate events, it is not possible to generate metrics or Key

Performance Indicators (KPI) per process instance or activity [7]. Furthermore, if the correlation data is wrongly chosen, the metrics would be incorrect, leading to a poor accuracy and loss of quality on analytical data. In this phase we must look close into the business process model table and identify the relationships between processes. The common properties along the business process will reveal good candidates for using their values as correlation data.

Table 2 – Correlation properties identification on the model table.

Process	Activity	Activity Parent	Properties
1#P_1	1#A_1		$\boxed{\mathbf{Prop_1}}$
	2#A_2	A_1	$\boxed{\mathbf{Prop_1}}$,$Prop_2$,$Prop_3$
	3#A_3	A_1	$\boxed{\mathbf{Prop_1}}$,$Prop_2$
2#P_2	4#A_4		$\boxed{\mathbf{Prop_1}}$
	5#A_5	A_4	$\boxed{\mathbf{Prop_1}}$,$Prop_4$

2.2.4 *Configuration Phase – Implement listeners.*

At this stage, we already have all the information required for implementing the software listeners that will capture business events from operational systems. Thus, the next step consists in building the software that will be capable to collect the execution event data of instances. Accordingly to the event format selected in the step 2, the event data must contain at least the mandatory entries stated in Table 3.

Table 3 – Event structure data.

Field	Description	Optional
EventId	Unique identifier for the event per business node.	No
Source	BASU unit.	No
ProcessDefinitionId	Definition of the process identified in model table.	No
ProcesName	Name of the process.	Yes
ActivityDefinitionId	Definition of the activity identified in model table.	Yes
ActivityName	Name of the activity.	Yes
ActivityParent	Parent of the current sub-activity.	Yes
StateTransition*	State transition for the current event. This is highly dependent of the message format.	No
Correlation[]	Set of key/value pairs used for correlation.	No
Payload[]	Set of key/value pairs that represent the structural properties of the process or activity.	Yes

2.2.5 *Configuration Phase – Selection of metrics and KPIs.*

The selection of metrics and Key Performance Indicators (KPI) are accomplished in this step. These metrics are essential to build a concrete understanding of what needs to be monitored and analysed. Within a Business Activity Monitoring (BAM) context, the construction of metrics and KPIs is intended to be performed at minimum latency, and this can be a data-intensive process in big data based DSS systems with BAM capabilities, such as [2]. Hence, the metrics and KPIs must be selected with caution. Once the metrics are activated in the DSS, we may or not establish thresholds per process or activity. This depends whether there already exists or not in the DSS historical information where the expected execution time of a process or instance could be inferred. In such case, the thresholds might be set in the BAM component to generate alerts, and thus detecting non-compliant situations. At following it is outlined the structural metrics that the DSS can deal with:

- **Running cases**: number of instances executed for a given process or activity.
- **Successful cases**: number of instances for a given process or activity that completed their execution successfully.
- **Failed cases**: number of instances for a given process or activity that finalized their execution with a failure state.

- **Aborted cases**: number of instances for a given process or activity that did not complete their execution.

This methodology also defines and uses the following behavioural metrics proposed in [8] :

- **Turnaround**: Measures the gross execution time of a process instance or activity.
- **Wait time**: Measures the elapsed time between the entrance of a process or activity in the system and the assignment of the process or activity to a user prior to the start of its execution.
- **Change-over time**: Measures the elapsed time between the assignment of the process or activity to a user and the start of the execution of the process or activity.
- **Processing time**: Measures the net execution time of a process instance or activity.
- **Suspend time**: Measures the time an execution of a process or activity is suspended.

Likewise, this methodology incorporates the *performance* dimension that is defined as a quality factor in [9]. The following measures refer to the performance dimension, and we adapt them to this methodology as KPIs that can be inferred from the metrics defined above. Below it is outlined the most relevant ones.

2.2.5.1 *Cycle-time*

Time is a universal and commonly used measure of performance. It is defined as total time needed by a process or activity instance to transform a set of inputs into defined outputs [9], i.e. the total amount of time elapsed until task completion. This KPI is automatically derived from the "Turnaround" metrics defined in [8], and it is provided by the DSS.

$T(a) = DD(a) + PD(a)$

a = activity.
T(a) = cycle Time duration of an activity.
DD(a) = Delay Duration of an activity.
PD(a) = Process Duration of an activity (processing time).

$DD(a) = CH(a) + WT(a) + ST(a)$

a = activity
DD(a) = Delay Duration of an activity.
CH(a) = Change over time of a process or activity.
WT(a) = Waiting time of a process or activity.
ST(a) = Suspended time of a process or activity.

$OF : Min T(a)$

OF = Objective Function.

2.2.5.2 *Time Efficiency*

This KPI is derived from the *Time Efficiency* quality factor defined in QEF. Activity *Time Efficiency* measures "how an activity execution is successful in avoiding wasted time". This KPI is the "mean of Time Efficiency in different instances of an activity execution". Formula for *Time Efficiency* KPI calculation is defined as follows:

$ET(a) = \frac{PT(a)}{T(a)} x100$

a = process or activity
ET(a) = Time of Efficiency of a process or activity
T(a) = cycle time duration of a process or activity
PT(a) = Planned Time duration of an activity. This is a big data based function that is inferred by the historical registry of the DSS.

$OF : E(a) \geq 100$

OF = Objective Function

2.3 Execution

This phase involves the execution of the operational systems. During this phase the listeners and the configured DSS become operational. The execution phase starts to capture the operational data and send business event data to the DSS. It is in this phase, when we must ensure that the configuration has been set up correctly, i.e. the incoming events recreate the business process designed in the first phase, and the metrics are generated accordingly to expected values. This trial period is very useful in complex business processes, such as very large supply chains. Before moving to the next phase, this trial phase should be completed successfully.

2.4 Control

At this stage, the business users are encouraged to monitor and analyse the outcomes of the DSS. The DSS pursues to avail and represent the analytical data from three different perspectives: 1) Historical Analysis: this drives the analysis of the event logs to provide business users with a powerful understanding of what happened in the past, 2) Business Activity Monitoring: this serves for evaluating what happens at present, and 3) Predictive Analysis: this will give analysts the ability to predict the behaviour of process instances in the future [6]. Figure 2 illustrates the different dimensions on where the analysis can be focused on in this phase.

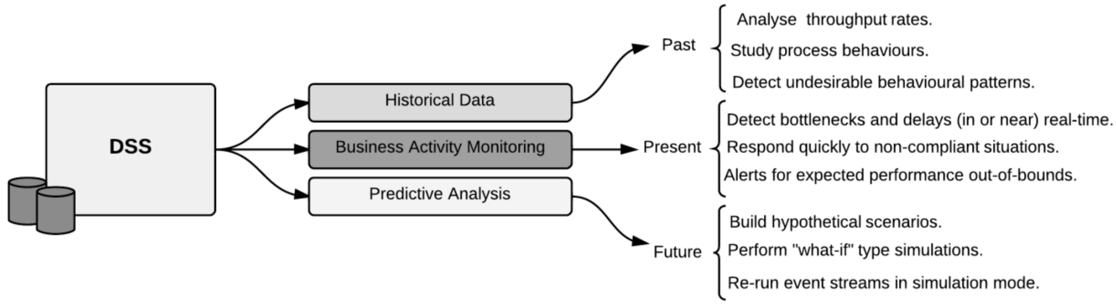

Fig. 2. Business process analytics on different dimensions.

2.5 Diagnosis

This last and final phase aims to identify deficiencies and weaknesses on the business processes identified in the *Define* phase. Business analysts may exploit the DSS capabilities such as visualization to identify hot-spots, or re-run event streams in simulation mode in order to perform root cause analysis, among others. Once the weaknesses are found, they must be eliminated from the operational systems. In such a case, the business process is re-designed and re-deployed on the operational environment, and the improvement lifecycle starts over again on a continuous refinement basis.

3. Case Study

We present a case study with the aim of testing the methodology using a big data based DSS described in [2]. The case study consists in analysing and improving the efficiency and security of the roads network in England. The study focuses on the motorways and major trunk roads, as these are the roads with highest traffic flow rates, and are the most interesting cases to identify hot-spots, safety and congestion on the roads. For attaining this study, we have used a real-life data set published by the Highways Agency. This data set is publicly available and fully accessible in [10]. It provides average speed, journey times and traffic flow information on all motorways and 'A' roads, known as the Strategic Road Network, in England.

2.6 Define

3.1.1 *Define Phase - Scope and boundaries identification*.

In this phase we identify six business nodes that correspond to the different areas in the road network. These areas are North West, North East, Midlands, East, South West and South East. Whereas the amount of data is presumably to be huge, we aim to breakdown the analysis per areas for performance and managerial reasons. Thereby, each business node will manage the data of its own area locally.

3.1.2 *Define Phase - Definition of sub-processes, activities and sub-activities*.

In this study case we aim to analyse the journeys of every vehicle per day along the road network. We define a global process as a specific route that links a source city to a destination across different areas. Consequently, we define a sub-process as a specific route that links source and destination, but only within the limits of determined area.

3.1.3 *Define Phase - Determination level of detail within business processes*.

The analysis is intended to be performed on the motorways and main roads of England. Hence, a sub-process level is sufficient for the purpose of this study.

3.1.4 *Define Phase - Development of model tables*.

We selected a variety of journeys that connect two cities by using different routes. For constructing the model table we used a road map for establishing the routes along with the properties of every road link. This information is supplied in the data set (refer to the data set published information in [10] for further details). The Table 4 illustrates a sample of one the process models developed.

Table 4 – Birmingham-Staffordshire process model for business node BASU-ML

Process	Activity	Activity Parent	Properties *
BirmStaf01#Journey [Birmingham - Staffordshire]	LM1015		PN, JD, TP, FR, AS,LL, LD
	LM1017		PN, JD, TP, FR, AS,LL, LD
	LM1019		PN, JD, TP, FR, AS,LL, LD
	LM1021		PN, JD, TP, FR, AS,LL, LD
	…		…

**PN=Plate Number, JD=Journey Date, TP=Time Period, FR=Flow Rate, AS=Average Speed, LL=Link Length, LD=Link Description*

2.7 Configuration

3.1.5 *Configuration Phase - Business nodes provisioning and software boundaries identification*.

We deployed six BASU nodes in a test environment for evaluating the approach. The business nodes deployed are: BASU-SW (South West), BASU-SE (South East), BASU-EA (East), BASU-ML(Midlands), BASU-NW (North West), BASU-NE (North East). Afterwards, in every BASU unit we loaded the process models (journeys) developed in the previous phase.

3.1.6 *Configuration Phase - Selection of event data format*.

We selected exBPAF as event format since we do not require integration with other process mining. Furthermore, exBPAF does not require format conversion on the DSS since it already deals with BPAF internally.

3.1.7 *Configuration Phase - Determine event correlation data*.

This phase is critical to recreate successfully the vehicle journeys. For the purpose of this case study, and assuming that a vehicle cannot drive along the same journey more than once a day, the correlation data to be used is the plate number and the day of journey. This information will identify uniquely the process along the sequence of events.

3.1.8 *Configuration Phase - Implement listeners*.

For the implementation of the listeners we simulated an Automatic Number Plate Recognition (ANPR) systems and in-vehicle GPS. The journey time, traffic flow (number of vehicles per road), and the rate of accidents, have been inferred from real information publicly available. The events have been generated by using pseudo-random numbers from a normal distribution based on the values provided in the dataset.

3.1.9 *Configuration Phase - Selection of metrics and KPIs*.

The set of metrics and KPIs selected for the purpose of this case study are specified below. The DSS-standard metrics are outlined in the Table 5 for representing behavioural measures, and Table 6 for the structural ones.

Table 5 – DSS-Standard behavioural measures.

DSS-Standard Measure	Description
Throughput time	Total amount of time for a vehicle to travel the road link.
Change-Over time	Time elapsed by joining a motorway from the slip road.
Processing time	Effective amount of time for a vehicle to travel the road link.
Waiting time	This measure has no sense in this study case. We start to track the vehicle when they start the journey.
Suspended time	This measure has no sense in this case study since we are not interesting in tracking journey

	interruptions. Only breakdowns and crashes will be reported once the vehicles start their journeys.

Table 6 – DSS-Standard structural measures.

DSS-Standard Measure	Description
Running cases	# of vehicles that started a journey through the road network.
Successful cases	# of vehicles that completed their journeys successfully.
Failed cases	# of vehicles involved in an accident during journey, and never reached their destinations.
Aborted cases	# of vehicles that did not complete their journeys due to breakdowns.

Regarding to the KPIs selection, we used a behavioural KPI (congestion) for measuring and identifying bottlenecks on key roads (in or near) time.

<u>Behavioural KPIs</u>

Congestion: This KPI uses the throughput measure and set a threshold value in those roads that are susceptible to experience some congestion at peak times. This threshold value is the mean of the journey time of the road for a given time, plus an increment which is determined by its normal distribution function. When the threshold is reached, an alert is fired on the DSS.

<u>Structural KPIs</u>

Reliability: This KPI measures the rate of successful cases in respect with to all running cases.

$R(p) = SC(p)/(RC(p) - AC(p))$ *R(p) = Reliability KPI for the process or activity p*
SC(p) = Number of successful cases for the process or activity p
RC(p) = Number of running cases for the process or activity p
AC(p) = Number of aborted cases for the process or activity p

Safety: This KPI measures the rate of failed cases (accidents) in respect with to all running cases.

$S(p) = FC(p)/(RC(p) - AC(p))$ *S(p) = Safety KPI for the process or activity p*
FC(p) = Number of failed cases for the process or activity p
RC(p) = Number of running cases for the process or activity p
AC(p) = Number of aborted cases for the process or activity p

2.8 Execution

The evaluation has been accomplished successfully in a test environment that follows the infrastructure depicted on the Figure 3. A vast amount of event data was generated by simulating the traffic flow experienced during Jan-Jun of 2013.

2.9 Control

We successfully experienced that the outcomes of the DSS were those expected. The execution results, measures and KPIs did not present any statistical significance in respect with the official values publicly available.

2.10 Diagnosis

This phase is out of scope in this paper since authors are using a public data set as input.

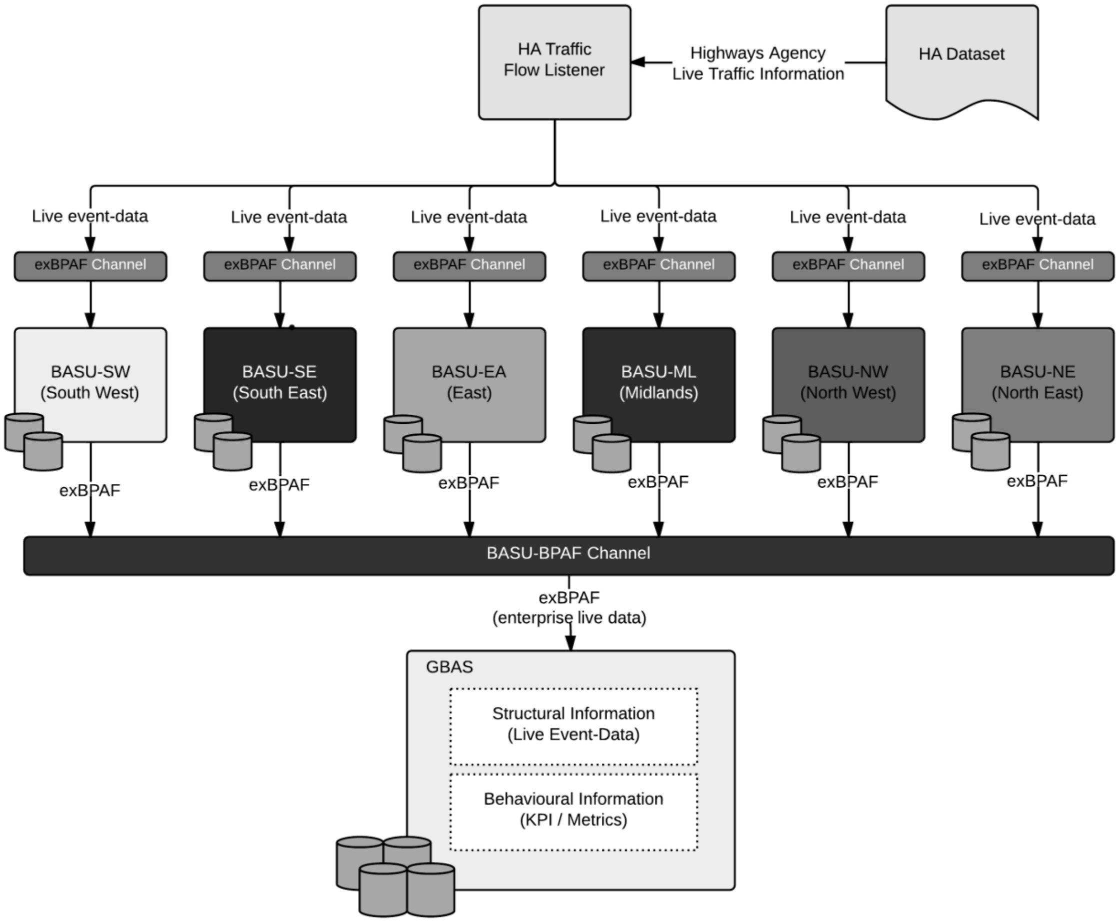

Fig. 3. DSS infrastructure.

4. Conclusions

This paper has presented a process devoted to guide Big Data based DSS in the field of business processes. The proposed methodology integrates Big Data based DSS with business process improvement activities with the aim of bringing real BPI technology to business users. A previous research work in the area of Big Data and business process analytics has been harnessed on this purpose. The outcome of this former work is a Big Data based IT solution that gives analysts an insight into business process performance in a timely fashion. This system has been adopted by the process in the form of DSS implementation as key-driven tool for supporting the BPI activities.

In the absence of suitable frameworks and tools for supporting process performance improvement initiatives, the authors propose a comprehensive BPI methodology that leverages the big data based DSS aforementioned to assist analysts in sustaining a full process improvement program.

Apart from the description of the process itself and based on previous works, authors presented a case study conducted in the field of traffic flow management. The methodology has come into practice by applying their principles in this area. Even though, the main research objective of the approach is focused on the improvement of large and complex distributed business processes that cross multiple organizations, the purpose of the process is well covered and successfully attained in the case study. This could be achieved by modelling traffic flows as business processes. This demonstrates the ability of the DSS to be agnostic to any business domain. Future endeavours will be devoted to apply the process designed in a wide range of organizations and functional scenarios to test its validity and guarantee its applicability.

References

[1] A. Vera-Baquero and O. Molloy, "A Framework to Support Business Process Analytics," 2012, pp. 321–332.

[2] A. Vera-Baquero, R. Colomo-Palacios, and O. Molloy, "Business Process Analytics Using a Big Data Approach," *IT Professional*, vol. 15, no. 6, pp. 29–35, 2013.

[3] A. Vera-Baquero and O. Molloy, "Integration of Event Data from Heterogeneous Systems to Support Business Process Analysis," in *Knowledge Discovery, Knowledge Engineering and Knowledge Management*, A. Fred, J. L. G. Dietz, K. Liu, and J. Filipe, Eds. Springer Berlin Heidelberg, 2013, pp. 440–454.

[4] N. Damij, T. Damij, J. Grad, and F. Jelenc, "A methodology for business process improvement and IS development," *Inf. Softw. Technol.*, vol. 50, no. 11, pp. 1127–1141, Oct. 2008.

[5] J. Becker, M. Matzner, O. Müller, and M. Walter, "A Review of Event Formats as Enablers of Event-Driven BPM," in *Business Process Management Workshops*, F. Daniel, K. Barkaoui, and S. Dustdar, Eds. Springer Berlin Heidelberg, 2012, pp. 433–445.

[6] M. zur Muehlen and K. D. Swenson, "BPAF: A Standard for the Interchange of Process Analytics Data," in *Business Process Management Workshops*, M. zur Muehlen and J. Su, Eds. Springer Berlin Heidelberg, 2011, pp. 170–181.

[7] O. Molloy and C. Sheridan, "A Framework for the use of Business Activity Monitoring in Process Improvement," in *E-Strategies for Resource Management Systems: Planning and Implementation*, E. Alkhalifa, Ed. IGI Global, 2010, pp. 21–46.

[8] M. zur Mühlen and R. Shapiro, "Business Process Analytics," in *Handbook on Business Process Management 2*, J. vom Brocke and M. Rosemann, Eds. Springer Berlin Heidelberg, 2010, pp. 137–157.

[9] F. Heidari and P. Loucopoulos, "Quality evaluation framework (QEF): Modeling and evaluating quality of business processes," *Int. J. Account. Inf. Syst.*, 2013.

[10] Highways Agency, "Live traffic information from the Highways Agency - Dataset," *data.gov.uk*. [Online]. Available: http://data.gov.uk/dataset/live-traffic-information-from-the-highways-agency-road-network. [Accessed: 01-Apr-2014].