PoS PROCEEDINGS OF SCIENCE

A Novel Data Analysis Service Oriented Architecture Using R

Rich C. Lee[1]
National Taipei University of Technology
#1, Sec. 3, Chung-Hsiao E. Rd., Taipei, 10608, Taiwan, R.O.C..
E-mail: RichChihLee@gmail.com

Making quality business decisions relies on the efficiency of data analysis processes. There are many existing Business Intelligence (BI) software products providing rich features to simplify the data analysis processes, but serving statistical purpose as effective and flexible as GNU-R (R) is rare. R is a highly extensible language and runtime environment providing a variety of statistical and graphical features. In enterprise environments, software records business activities in various forms such as files, database, and streaming data. Currently analysts conduct the data analysis in an offline mode using statistical software; it means that the analysts are manually (1) extracting the desired data; (2) storing extracted data into files; (3) manipulating the software; (4) drawing analytical results, and (5) making the inferences. Automating the statistical procedures—by directly extracting the data—will expedite the decision-making sooner and less costly. It is a common practice that enterprises build applications under Service-Oriented Architecture (SOA) to achieve their operation excellence. This paper proposes a service-oriented statistics framework integrating R to improve the quality of decision-making by automatically extracting the business live data directly. It also presents how such a system was developed from a software engineering perspective to benefit business decision making.

The International Symposium on Grids and Clouds (ISGC) 2012
Academia Sinica, Taipei, Taiwan
February 26 – March 2, 2012

[1] Rich C. Lee Senior I/T Specialist, System & Technology Group, IBM Taiwan Corporation, 3F, No.7, Songren Road, Taipei, 11073, Taiwan (richlee@tw.ibm.com). He was the solution/business architect in Cloud Computing and High Performance Computing related fields in IBM. He was the solution/business architect of BusinessWare, a Business Process Management platform when he served Vitria. He joined Foxconn USA, in early 2000 and brought new desktop assembly model for massive production, built up global logistic networks, and was in charge of manufacture business process design and SAP rollout. He was the initiator of Taiwan Supply Chain Project in 1999 and lead Mobile Operator Solution team in 1997 when he served Compaq. He also worked for DEC in 1995 before the merger by Compaq.

1. Introduction

Enterprise business performance relies on the responsiveness and the quality of decision-making. Decision-making should be based on the rationale of knowledge which was accumulated and compiled from the information that was generated from business activities. The decision-makers have to develop a feasible approach to acquiring sufficient knowledge to make a confident enough decision to support the business needs [1]. In modern business management, decisions are usually derived from responsive reliable data analysis. Business managers should justify decisions on the basis of data analysis from statistical model-based Decision Support Systems (DSS) [2]. An enterprise strives to improve processing and reduce variation; it requires a disciplined, data-driven approach to reach business goals. Statistics can help enterprise describe how well the processes are performing in a quantitative way, transforming the business activity facts to valuable information, and later becoming useful knowledge.

Efficiently analyzing data, deriving information, and accumulating knowledge is still a goal for enterprise information environments. First, the analysts must know where the desired data are, ask IT professionals to retrieve that data, save it into files, store them under a shared folder over the network, and then develop statistical models. If the data are considerably large and frequently-updated, it will make the statistical process more difficult. If another analyst concerns about just a part of the data or the data are not fully covered, the IT professionals must repeat these resource-intensive tasks again for each of their requirements. The overlapped part of these data files is duplicate effort and waste. Another drawback is the poor reusability if the statistical procedures are not shared for similar analysis. In fact, enterprise’s knowledge does not just cover the information itself but also how the processes derived this information. These processes ensure the reproducibility of knowledge that is a part of the enterprise Intelligent Properties (IP). To facilitate the enterprise’s knowledge building processes, Business Intelligence (BI) System is used to rediscover the usefulness of the existing data. It requires that the Subject Matter Experts (SME) and the software product engineers closely work together to develop analytical models for further disclosure of the facts and implications of current business performance. This is rather expensive and complicated process; the enterprise usually outsources a consulting project to do the works. Unlike commercial BI software products, GNU-R is a flexible and extendable language and runtime environment for statistical computing and graphics; it has been widely used in many applications for years. The analysts who are planning to use GNU-R can find abundant resources over the Internet to ease their learning curves. Thus, GNU-R is considerably more cost-effective as a statistical engine automating the analytical processes than commercial BI software products. Therefore, it is worthwhile to develop a *Statistical Service Engine* solution using R to improve the processes of enterprise knowledge generation and the reusability of knowledge as well.

However, while developing such a *Statistical Service Engine* solution to meet time-to-market objective within budget, it is important to know why most software projects failed. Reasons include (1) unrealistic or inarticulate project goals, (2) inaccurate estimates of needed resources, (3) badly defined system requirements, (4) poor reporting of the project's status, (5) unmanaged risks, (6) poor communication, (7) use of immature technology, (8) inability to handle the project's complexity, (9) sloppy development practices, (10) poor project management, (11) stakeholder politics, and (12) commercial pressures [3]. The third factor—the badly defined system requirements—attracts the most concerns in development. The system requirements have two aspects, functional and non-functional. The functional requirements are about software features and specifications; while non-functional requirements are more about the quality of software to ensure that the software capability do meet business goals with less *Total Cost of Ownership* (TCO). Many software projects that failed to respond the business needs were because the software development did not considered the non-functional requirements in the first place.

Since the purpose of the *Statistical Service Engine* solution—playing a supportive role within the knowledge building chain—is to provide an effective way to improve the business performance visibility and the quality of decision-making, therefore, the reusability of knowledge in the form of statistical models is the key to success. GNU-R is a script-like statistical programming language; the analysts design scripts to present their business perspectives—the statistical inferences—to enterprise; a knowledge repository stores and shares these scripts with authorized persons and prevents the analysts from reinvent-the-wheel. These scripts are also building blocks for broader applications; the analysts can look up the repository first, and then use the selected scripts as templates or as a library doing further design. This reusability saves lots of design time and gives greater quality of the statistical models.

The major drawback of exporting the live data into files for analysis is that uses too much resource in terms of disk spaces and processing time. If the data are volatile and the analysis requires periodical re-evaluation within a short time window, the resource-consuming export task might harm major operating tasks. GNU-R has the ability to access databases directly from analysts' desktops to extract live data through *Open Database Connectivity* (ODBC), but this will cause a security breach because the database schemas must be disclosed to the analysts. Too many ODBC connections at the same time will exhaust the maximum connection number that the database server can provide. This certainly will potentially jeopardize the performance of databases to serve the primary business activities.

Some statistical models might consume more computing resources than a desktop computer can offer; if the analysis takes too much time, the reporting might lose meaning for mistiming. Therefore, the proposed solution must be running on a set of high performance distributed servers, sharing the database connectivity, and executing the GNU-R scripts under a unified reliable mechanism.

To meet the requirements of the proposed solution, the *Enterprise Service Bus* (ESB) is a proven software infrastructure which underpins a fully integrated and flexible end-to-end *Service-Oriented Architecture* (SOA). The ESB enables SOA by providing a connectivity layer between services [4]. The ESB combines event-driven and service oriented approaches to simplify the integration of business units, bridging heterogeneous platforms and environments. The ESB acts as an intermediary-layer to enable communications between different application processes. A service deployed onto an ESB can be triggered by a consumer or an event. It can be synchronous or asynchronous communication, facilitating interactions between one or many applications (One-to-One or Many-to-Many communications) [5]. The GNU-R scripts are applications running under ESB. Since the execution time of these scripts varies, the result may not be produced within a reason time period, so processing these scripts asynchronously is a feasible approach to the proposed solution.

Accumulation of knowledge is a complex and dynamic process; it needs re-analysis to enhance usability—a core term in human–computer interaction—which depends on user-friendly interfaces that encourage the participants to acquire and disseminate their findings from business activities. Another important consideration for technology selection for the proposed solution is the platform portability. Java *Portal* technology (JSR-168/268) is well-known for its portability and the usability. It lets the developers focus more on the business logic and less on the layout of the appearance. It also enables users to dynamically construct and reconstruct Web information applications at runtime to resolve urgent and unplanned business requirements [6].

This paper discloses the important parts of the design considerations about the proposed solution as follows: (1) for achieving software quality, it presents what non-functional requirements should be considered; (2) for the effective data extraction concern, it presents how GNU-R scripts get live data from databases or web applications; (3) for achieving reliability, it presents a robust design of software architecture including the solution components; (4) for achieving efficient inter-communication, it presents the message exchange scheme among the solution components; (5) for practical use of the solution, it presents an empirical case adapting the solution to analyse the performance of MES servers, and (6) the conclusion and future work.

2. Quality Attributes of Statistical Service Engine Solution

It is obvious and important that the *Statistical Job Engine* software must be of high quality in design because the business decisions depend on its reliable outputs. The software quality attributes are defined, measured, and evaluated; so that the software can meet the business holistic requirements after it is built. From a Software Engineering perspective, the software design requires both functional and non-functional requirements. The quality attributes discussed here are non-functional requirements that cover holistic aspects of the software including: (1) **functionality** about capability, security; (2) **usability** about human factors; (3) **reliability** about failure prediction, (4) **performance** about throughput, and (5) **supportability** about serviceability [7].

The proposed solution especially focuses on the reliability and the supportability attributes. The reliability attributes are about (1) **maturity**—the frequency of software faults, (2) **fault tolerance**—the ability of software to deal with the software faults or the infringement of its specified interfaces (e.g. protocols, incompatible data formats, compensated message routes) of service, and (3) **recoverability**—the capability to recover data affected in case of a failure and measured by the time and effort needed for it [8]; while the supportability attributes cover (1) **testability**—the ability of testing, (2) **extensibility**—the ability of adding new features, (3) **adaptability**—the ability of running on other platforms, (4) **maintainability**—the ability of easy-to-maintain the source codes, (5) **compatibility**—the ability of running on similar environment, (6) **configurability**—the flexibility of the software through configurations, (7) **serviceability**—the availability of the service, (8) **installability**—the ability of easy software installation, (9) **localizability**—the ability of support multiple languages, and (10) **portability**—the ability of porting the software on other platform [7]. **Table 1** shows the most important attributes of the proposed solution about reliability and supportability listed as follows:

Table 1: Quality Attribute Requirements

Quality Attribute		Requirement
Reliability	Maturity	Proven Technologies.
	Fault Tolerance	Job execution at all times.
	Recoverability	Job automatically resumed when failed.
Supportability	Testability	Feature self-tests at any time.
	Extensibility	Adding more computing resources at any time.
	Adaptability	Standard protocols of service integration.
	Maintainability	Providing management console.
	Compatibility	None
	Configurability	Providing management console.
	Serviceability	7x24 operations.

3. Statistical Service Engine

GNU-R provides a powerful programming language and a statistical environment; it is very extensible, with hundreds of add-on packages obtainable from CRAN (the "*Comprehensive R Archive Network*"), and providing a high quality of graphical output [9]. The *r-base-core* is a universe/math package described as a GNU R core of statistical computation and graphics system. GNU R script can access databases by the *r-cran-rodbc* package through ODBC. The package should be platform independent and provide access to any database for which a driver exists [10]. **Table 2** shows how R script accesses the database through ODBC.

Table 2: GNU-R Uses ODBC Example

```
myChannel <- odbcConnect("PerfRepo")
sqlSave(myChannel, PerfInfo, rownames="EventDate", verbose=TRUE)
sqlQuery(myChannel, paste("SELECT EventDate, CPU from PerfInfo",
"WHERE EventDate > '2011/11/1' ORDER BY EventDate"))
close(myChannel)
```

On many occasions, the source data for statistical analysis is sophisticated and requires complex processing to generate. Using Java *servlet* technology is a common practice to populate complex data from various sources and transform them into the desired format. The HTTP output format should be text/plain with tab delimiter for data columns. **Table 3** shows how R script gets the data from a web page.

Table 3: GNU-R Reads Data from HTTP

```
mySourceData <- readtable("http://sjp.foo.com/getPerfData")
```

Graphics help users to comprehend the trend and the meaning behind statistical figures intuitively. Normally the analyst uses a tool to visualize the statistical graphics. The proposed solution runs GNU-R scripts on behalf of the analyst and saves the generated graphics into an image file asynchronously so that the analyst can visualize the image file later. **Table 4** shows how R script saves graphics into an image file.

Table 4: GNU-R Saves Graphics into an Image File

```
setwd('/data/images')
reset <- par(no.readonly=TRUE)
par(mfrow=c(1,3))
plot(x[,1],x[,2],xlim=c(min(x[,1],x[,2]),max(x[,1],x[,2])),
ylim=c(min(x[,1],x[,2]),max(x[,1],x[,2])))
plot(x[,1],x[,3],xlim=c(min(x[,1],x[,3]),max(x[,1],x[,3])),
ylim=c(min(x[,1],x[,3]),max(x[,1],x[,3])))
plot(x[,2],x[,3],xlim=c(min(x[,2],x[,3]),max(x[,2],x[,3])),
ylim=c(min(x[,2],x[,3]),max(x[,2],x[,3])))
savePlot('myPlot',type="jpg")
par(reset)
```

A Java program has two approaches to execute GNU-R scripts (1) using *Runtime.getRuntime().exec()*, and (2) using *JRI* as the interface to GNU-R. It allows Java program to take control over R. **Table 5** shows how Java program interacts with R.

Table 5: JRI Interacts with GNU-R Example

```
Rengine re=new Rengine(args, false, new TextConsole());
REXP x;
re.eval("print(1:10/3)");
System.out.println(x=re.eval("iris"));
RVector v = x.asVector();
if (v.getNames()!=null) {
    System.out.println("has names:");
    for (Enumeration e=v.getNames().elements();
         e.hasMoreElements(); ) {
        System.out.println(e.nextElement());
    }
}
if (true) {
    System.out.println("Console:");
    re.startMainLoop();
} else {
    re.end();
    System.out.println("end");
}
```

4. The Architecture of Statistical Service Engine Solution

Based on the above mentioned nature of the statistical analysis; the scripts take different length of time and consume different computing resources according to their complexities.

Asynchronous messaging is the solution for dealing with such various time-consuming requests. Using the asynchronous messaging scheme, the request is done after submission, simply returns an acknowledge flag back to the caller process. The real response of the executed script will be reported some time later in a different communication session. The user does not wait for the script output at the request time but retrieves the output later after the called process has finished [11]. Asynchronous messaging relies on the message-oriented middleware (MOM). The MOM creates a “software bus” for integrating heterogeneous applications [12]. Messaging is distributed across applications running on different servers; its reliability and robustness is crucial to MOM. The ESB is an open standards, message based, distributed integration infrastructure that provides routing, invocation and mediation services to facilitate the interactions of disparate distributed applications and services in a secure and reliable manner [13]. **Figure 1** illustrates the system architecture of the Statistical Service Engine solution.

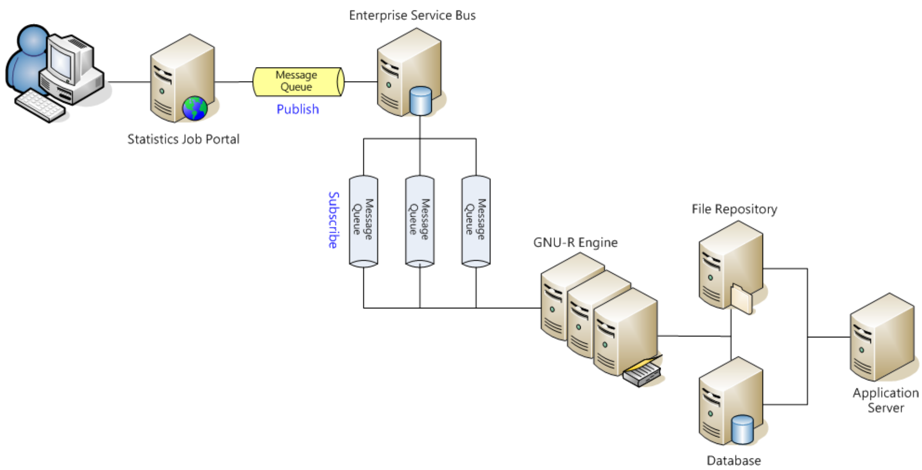

Figure 1: Statistical Service Engine Solution

The *Statistical Job Portal* server provides a number of predefined statistical procedures and ad-hoc analysis to users. The *Enterprise Service Bus* server receives messages, the statistical job requests, from the *Statistical Job Portal* server and dispatches them accordingly. *GNU-R Engine* is a set of blade servers executing the GNU-R scripts designated by the messages. The script may retrieve the data from the *Database* servers or the *File Repository* server. The *Application Server* is where the user processes populate the source data from the *Database* servers or the *File Repository* server. The *File Repository* server also stores the statistical results and for users to retrieve later. The statistical job request is depicted in the XML format; **Table 6** exhibits the most significant fields of the request.

Table 6: Statistical Job Request

Field	Description	Example
UserID	User Identification	richlee@tw.ibm.com
SubmitDateTime	Job Submitted Date/Time	20111127-065530

JobID	Statistical Job Identification	Statistics Summary: 00:Ad-hoc 01:CPU 02:MEM 03:I/O 04:Tnx Regression 11:CPU~Tnx 12: MEM 13: I/O~Tnx Moving Average: 21:CPU 22:MEM 23:I/O 24:Tnx Others
Ad-hoc Script	GNU-R Script	If JobID==00
SourceData	Statistics Source Data	If JobID==00, /DB0310/20111127.RData
ResultDestination	Statistics Result Destination	0: Web 1: Email 2: Both
ServerID	Server Identification	DB0310
DateRange	Date Range of Performance Counters	20110901~20111127

Figure 2 illustrates the UML *Activity Diagram* of the solution. There are two routes for processing the requests from (1) *predefined* and (2) *ad-hoc Statistical Job*. When a user selects a predefined job from the *Statistical Job Portal*, the response portal event sends the *Statistical Job Request* message to *Message Queue*. One of the *GNU-R Engine* servers gets the routed message from the ESB and does the following tasks:

(1) *Predefined Statistical Job*:

(1-1) Loads predefined GNU-R script.
(1-2) Replaces the parameters (i.e. *ResultDestinamtion*, *ServerID*, and *DateRange*, etc.).
(1-3) Spawns GNU-R interpreter ro run the prepared script.
(1-4) Retrieves data from *File Repository* or *Database* server.
(1-5) Populates the statistical result on *File Repository*.
(1-6) If *ResultDestination>0* emails the statistical result to the requested user.

(2) *Ad-hoc Statistical Job*:

(2-1) Loads the GNU-R script from the message received.
(2-2) Spawns the GNU-R interpreter to run the prepared script
(2-3) Retrieves the designated data from *File Repository* or *Database*.
(2-4) Populates the statistical result on *File Repository*.
(2-5) If *ResultDestination>0* emails the statistical result to the requested user.

The user can view the statistical result from the *Statistical Job Portal* under the [*My Job Results*] web page if *ResultDestination!=1*. The web page [*My Job Results*] lists down all the statistical results of previous submitted jobs with hyperlinks.

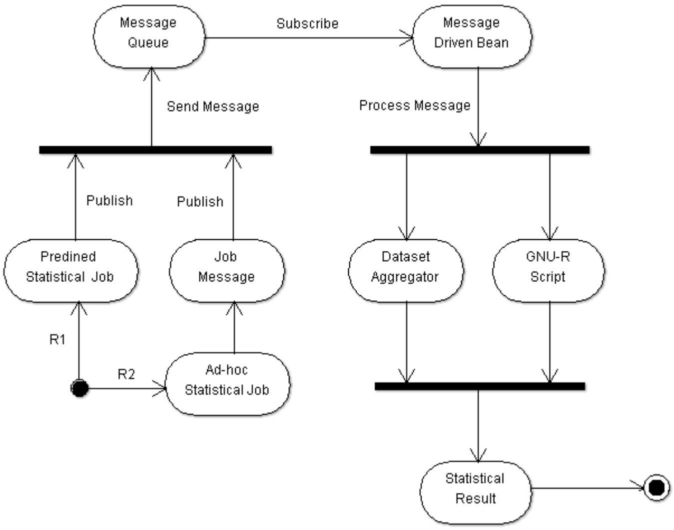

Figure 2: Activity Diagram of Statistical Service Engine Solution

5. Empirical Case

In the context of the *Computer-Integrated Manufacturing* (CIM) architecture, the *Manufacturing Execution System* (MES) provides information and controls the decision links between the *Production Planning System* at the higher level and the *Equipment Control System* at the lower level [14]. An MES usually has a number of application servers controlling various types of equipment and storing production information in databases. It is a common practice to deploy an agent on each MES server to collect system performance counters namely *CPU*, *MEM* and *I/O*; and capture these performance figures into a repository for further diagnostic use.

The MES administrator monitors performance counters and interprets the meaning of these figures to determine whether the system is healthy or not. However, when MES scales out and becomes more complex, it is impractical to determine the health of the system from a simple perspective of performance counters i.e. CPU utilization, Memory Paging and I/O Frequency. Such a simple perspective does not help the MES administrator to identify either the root cause when a problem occurs or how much the resources should be augmented to make the system more robust when high demand calls. To manage the system effectively, the MES administrator requires more comprehensive statistical perspectives against these performance counters and an automated-mechanism to simplify the procedures of statistical analysis. Therefore, the proposed solution was built to meet MES advanced performance analysis objectives. **Figure 3** illustrates an overview of MES performance statistical analysis mechanism which was designed on the *Statistical Service Engine* solution.

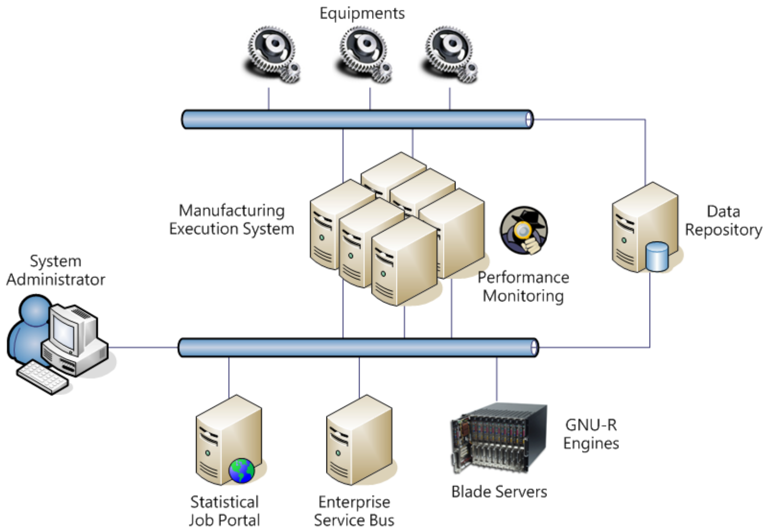

Figure 3: Integrated System Performance Analysis Solution

A performance counter collection service, *nmon*, is deployed on each MES server to collect performance related counters such as *CPU*, *MEM*, and *I/O*. The *nmon* tool is helpful in presenting all the important performance tuning information and capturing the performance data into a text file for latter analysis and drawing graphs for reports [15]. All daily *nmon* data files are stored in the *File Repository* server under separate sub-folders. The *File Repository* server also keeps MES performance counters within log files including the following fields: (1) *Time Period*, (2) *MES Server ID*, (3) *Transaction Counts*, and (4) average end-to-end *Response Time* periodically.

A predefined statistics job that is scheduled to prepare data sets and run the *Statistical Job Server* periodically sends job request messages to the ESB server. The blade servers subscribe to the job request messages. When job request message arrives at a blade server, a service program triggers a GNU-R script to parse the *nmon* data and the MES performance log files, transforming them into GNU-R datasets, and saving these datasets to a GNU-R Workspace (.Rdata) file under separate folder.

The MES administrator can submit a request of desired performance statistics perspective from the *Statistical Job Portal* web pages and get back the statistical results after the *GNU-R Engine* has executed the submitted GNU-R scripts. The submitted job request messages are sent to the ESB; the blade servers respond these requests when the XML messages arrive, and then execute the corresponding GNU-R scripts to populate the result files.

The MES administrator can view these results by comparing the job request information against performance counters from the Statistical Job Portal. **Table 7** shows some interesting perspectives to the MES administrator; these perspectives disclose the behavior of MES, which cannot be noticed by the conventional means—simply looking at the performance counters separately. **Figure 4** shows the linear regression of *CPU* utilization against *I/O* and *Memory Page Faults*. When production ramps up, more computing resources will be added into the MES. It is natural that the MES administrator wishes to compare the performance improvement before and after resources were added. **Figure 5 and 6** shows the performance behavior difference after two more CPUs were added into a server.

The Integrated-System-Performance Analysis solution - an implementation of Statistical Service Engine, helps MES administrators to monitor and comprehend MES servers' performance behaviors. Through regression analysis against MES transaction and computing resources, the MES administrator is able to forecast when to add more resources to respond to increasing production demand.

Table 7: An Integrated System Performance Analysis Solution

```
lm(formula = UMC$CPU.Total ~ UMC$IO.Avg + UMC$Paging.Faults)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.8726 -0.3320 -0.1451  0.4524  0.7583 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)  
(Intercept)       3.765e+01  1.192e+01   3.158    0.016 *
UMC$IO.Avg        6.132e-04  2.334e-04   2.628    0.034 *
UMC$Paging.Faults 1.066e-03  6.197e-04   1.720    0.129  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 0.6098 on 7 degrees of freedom
Multiple R-squared: 0.8063,   Adjusted R-squared: 0.751 
F-statistic: 14.57 on 2 and 7 DF,  p-value: 0.003197
```

Figure 4 CPU Utilization Analysis

```
Call:
lm(formula = CPU.Total ~ Page.Faults + IO.DiskSpeedAvg, data = UMC.Before)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.8726 -0.3320 -0.1451  0.4524  0.7583 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)  
(Intercept)     3.765e+01  1.192e+01   3.158    0.016 *
Page.Faults     1.066e-03  6.197e-04   1.720    0.129  
IO.DiskSpeedAvg 6.132e-04  2.334e-04   2.628    0.034 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 0.6098 on 7 degrees of freedom
Multiple R-squared: 0.8063,    Adjusted R-squared: 0.751 

F-statistic: 14.57 on 2 and 7 DF,  p-value: 0.003197
```

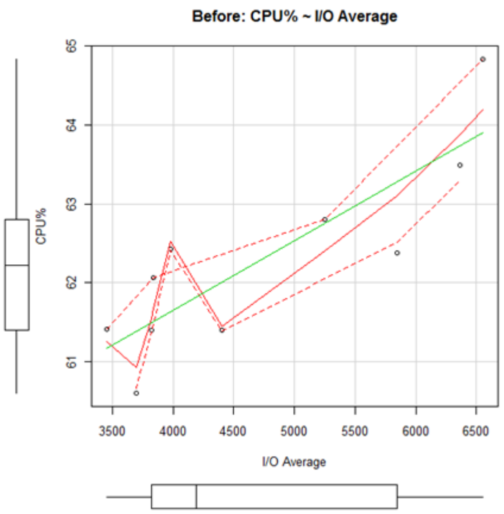

Figure 5 CPU Utilization vs I/O Rate (Before)

```
Call:
lm(formula = CPU.Total ~ Page.Faults + IO.DiskSpeedAvg, data = UMC.After)

Residuals:
    Min      1Q  Median      3Q     Max
-0.5215 -0.3119 -0.2145  0.1907  1.1906

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     4.607e+01  1.208e+01   3.813   0.0066 **
Page.Faults     2.767e-04  5.854e-04   0.473   0.6508
IO.DiskSpeedAvg 2.101e-03  8.243e-04   2.549   0.0381 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6155 on 7 degrees of freedom
Multiple R-squared: 0.4821,    Adjusted R-squared: 0.3341
F-statistic: 3.258 on 2 and 7 DF,  p-value: 0.09996
```

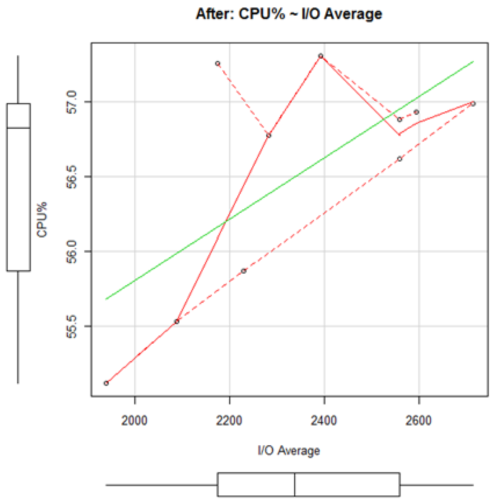

Figure 6 CPU Utilization vs I/O Rate (After)

6. Conclusion and Future Work

To increase business competitiveness requires continuous innovation and operational excellence. Re-examining the data of business activities, finding patterns and trends in various statistical perspectives will help an enterprise make rational decisions swiftly. Knowledge Management system offers a collaborative platform—to acquire, compile, disseminate, and reuse the knowledge—to elicit the creative and to improve operational efficiency as well. A more intense use of knowledge management platforms has both direct and indirect (innovation-mediated) with positive effects on enterprise performance [16]. To make knowledge management platforms successful, the users' perceived usefulness and user satisfaction are key. [17]. Based on useful statistical results, business competitiveness improvement will stimulate and inspire employees' further investigations by reusing these statistical procedures and the data. This reinforced process helps employees use knowledge management platforms more intensively and helps them make business decisions more rationally and swiftly. To realize this goal, a more convenient solution is needed to help analysts retrieve data, reuse statistical procedures, and disseminate the findings more easily. This paper answered such a need and presented a reliable and robust architecture on top of an Enterprise Service Bus. It also showed how GNU-R scripts access live data from databases, generate statistical graphics and save them to image files. From a software engineering perspective, this paper demonstrated how to define and measure the quality attributes of the solution to ensure the solution will meet business requirements during the development cycle.

The proposed solution is still very primitive but workable. More useful statistical models should be developed and categorized. The solution has more extensibility in the reuse and dissemination of statistical work products. To integrate with existing proven knowledge management platform products via Web Services seems to be a practical approach. The first step is to plan a knowledge sharing scheme, which is a part of the tasks of knowledge management, by configure appropriate taxonomy and document hierarchical folders for statistical results, so that other analysts can easily find the templates and reuse them to derive further inferences. The second is to apply Ajax technology to embed the statistical job submission form in the proposed solution. The third is to design an interface to check in the statistical results from the file repository to the knowledge management platform by Web Service calls. Fourthly since the computing resources (i.e. CPU, MEM, and I/O) consumption is determined by the complexity of statistical model and the volume of data, the GNU-R Engine must re-dispatch the job to another available message queue if resource-shortage failure occurs. To resolve that, designating a First-in-First-out (FIFO) sequential processing queue on a dedicated blade server to handle those earlier failed requests should be a feasible approach. Lastly, this paper urges employees to take advantage of existing data and transform them into useful knowledge to enhance the quality of decision making in their enterprises and let statistics become the core of strategy reasoning.

From the quality of business decisions perspective, the enterprises can be categorized by their levels of analytics capability: (1) **aspirational**—such enterprises are furthest away from progress, focusing more on efficiency or automation of existing processes and searching for ways of innovation; (2) **experienced**—such enterprises are with some analytic experience and are looking for ways to go beyond cost management; and (3) **transformed**—such enterprises are with substantial experience using analytics across a broad range of functions [18]. Rediscovering the business facts and forming the new strategies swiftly depends on the ubiquitous analytics of business activities. This paper presents a business analytics solution using R—a concept of leveraging the power of open-source software—to enhance the quality of decision making.

References

[1] B. P. Kumar, J. Selvam, V. Meenakshi, K. Kanthi, A. Suseela, and V. L. Kumar, "Business Decision Making, Management and Information Technology," *Ubiquity*, vol. 2007, p. 4, 2007.

[2] H. Arsham. (2011). *Statistical Thinking for Managerial Decisions (9 ed.)*. Available: http://home.ubalt.edu/ntsbarsh/Business-stat/opre504.htm

[3] R. N. Charette, "Why Software Fails," *IEEE spectrum*, vol. 42, p. 36, 2005.

[4] M. T. Schmidt, B. Hutchison, P. Lambros, and R. Phippen, "The Enterprise Service Bus: Making Service-Oriented Architecture Real," *IBM Systems Journal*, vol. 44, pp. 781-797, 2005.

[5] J. L. Maréchaux, "Combining Service-Oriented Architecture and Event-Driven Architecture Using an Enterprise Service Bus," *IBM Developer Works*, 2006.

[6] Y. Wei, Z. Sun, X. Chen, and F. Zhang, "A Service-Portlet Based Visual Paradigm for Personalized Convergence of Information Resources," *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*, pp. 119 - 124, 8-11 Aug. 2009 2009.

[7] L. Chung and J. do Prado Leite, "On Non-functional Requirements in Software Engineering," *Conceptual modeling: Foundations and applications*, pp. 363-379, 2009.

[8] B. Behkamal, M. Kahani, and M. K. Akbari, "Customizing ISO 9126 Quality Model for Evaluation of B2B Applications," *Information and software technology*, vol. 51, pp. 599-609, 2009.

[9] A. T. Yalta and R. Lucchetti, "The GNU/Linux Platform and Freedom Respecting Software for Economists," *Journal of Applied Econometrics*, vol. 23, pp. 279-286, 2008.

[10] a. f. t. O. M. L. Brian Ripley. (2011). *ODBC Database Access*. Available: http://cran.r-project.org/web/packages/RODBC/RODBC.pdf

[11] A. Friis - Christensen, N. Ostländer, M. Lutz, and L. Bernard, "Designing Service Architectures for Distributed Geoprocessing: Challenges and Future Directions," *Transactions in GIS*, vol. 11, pp. 799-818, 2007.

[12] C. F. Liao, Y. W. Jong, and L. C. Fu, "Toward a Message-Oriented Application Model and Its Middleware Support in Ubiquitous Environments," 2008, pp. 298-303.

[13] F. Menge, "Enterprise Service Bus," 2007.

[14] M. J. Sheng-Luen Chung, "Manufacturing Execution System (MES) for Semiconductor Manufacturing," *Systems, Man and Cybernetics, 2002 IEEE International Conference on*, vol. 4, p. 5, 6-9 Oct. 2002 2002.

[15] N. Griffiths. (2006). *nmon Performance: A Free Tool to Analyze AIX and Linux Performance*. Available: http://www.ibm.com/developerworks/aix/library/au-analyze_aix/

[16] A. Vaccaro, R. Parente, and F. M. Veloso, "Knowledge Management tools, Inter-organizational Relationships, Innovation and Firm Performance," *Technological Forecasting and Social Change,* vol. 77, pp. 1076-1089, 2010.

[17] U. R. Kulkarni, S. Ravindran, and R. Freeze, "A Knowledge Management Success Model: Theoretical Development and Empirical Validation," *Journal of Management Information Systems,* vol. 23, pp. 309-347, 2007.

[18] S. LaValle, E. Lesser, R. Shockley, M. S. Hopkins, and N. Kruschwitz, "Big Data, Analytics and the Path from Insights to Value," *MIT Sloan Management Review,* vol. 52, pp. 21-32, 2011.