

1

Towards Web-based Productivity Analysis and Reporting

Michael Mwebaza¹, Emmanuel Mnzava², Joyce Nakatumba³, Peter Nabende⁴
1,2,4{mmwebaza, memmanuel, pnabende}@cit.mak.ac.ug
1,2,4Faculty of Computing and Information Technology, Makerere University,
1,2,4P.O.Box 7062, Kampala, Uganda.

3jnakatumba@must.ac.ug
3Institute of Computer Science, Mbarara University,
3P.O. Box 1410, Mbarara, Uganda.

Abstract

In this paper, we propose a Web-based Productivity Analysis and Reporting Tool (W-PART) for applications that necessitate gathering productivity data from remote sites for a given business entity. W-PART is aimed at reducing on productivity data entry and analysis workload from a central input point. Other benefits that we expect to realize from the tool include reduction of data loss and time taken entering productivity data. The tool is generally aimed for utilization by any given organization in a 'developing area', but relies on availability of the Internet in any form. We report on the current status in the development of the tool and discuss its software implementation prospects. For the purpose of discussing different aspects concerning the tool, we use banks as a case study. We avail general implementation requirements for the tool which we expect should lead to a variety of options for implementation.

1. Introduction

Productivity and profit margins are considered one of the most important performance indicators for Managers or CEOs in their strategic decision-making (Stevens, 1998). The measurement of productivity for a given business entity is obtained through what

is commonly referred to as Productivity Analysis. Productivity analysis comprises measurement, interpretation, and evaluation of the measurement results (Rao et al., 2005). Before the onset of computers, productivity analysis was carried out manually, but later in the 20th century, many companies started using computerized techniques mainly for the measurement function. Of recent, computerized techniques for the Interpretation and Evaluation functions have also been proposed and implemented, mostly in the form of expert systems (Rao et al., 2005) and decision support systems. This paper is concerned mainly with the capturing and analysis of productivity data or information in the Measurement function of productivity analysis. Currently, spreadsheet technology stands out as the prevalent tool for productivity measurement. A common procedure is as follows: data for productivity measurement is captured in a spreadsheet from different sources within an organization or outside the organization and the spreadsheets are emailed or manually submitted to an analyst who tallies the various records and merges them into one, plots graphs, and writes some basic spreadsheet logic to enhance the functionality of the results for management review. For some business entities, this may be a sufficient procedure for productivity measurement. However, when a business entity is spread over various geographic sites (for brevity, we hence use the phrase “remote sites” for such sites) and there is need to access productivity information from the remote sites on a frequent basis, the just described procedure that utilizes spreadsheet technology will not scale well. There are also other problems that are associated with using spreadsheet technology regardless of the nature of the productivity information requirements: there is a high possibility of data loss or loss of data integrity as the data is copied from one spreadsheet to another; portability of the data is also difficult since technical specifications such as formulas for referencing data from other spreadsheets will return errors if they are copied from one location to another without the source spreadsheet being copied as well.

Based on the limitations of using spreadsheet technology for productivity measurement, and the requirements for accessing data from various external sites of a business entity, we propose: first, a model for the automated distributed capture of productivity measurement data that can be used within an organization – this constitutes the preliminary work associated with this paper; later, we propose extended work on implementing this model for automated capture and analysis of productivity measurement data from various remote sites of an organization. We take as our case study, financial institutions, specifically banks in a developing country. However, the ideas presented in this paper can be adapted to many organizations where productivity analysis and reporting is essential. In section two, we present related work, in the third section, we propose an approach for developing the tool, in the fourth section we describe the different requirements and propose abstract

structures for the tool, in the fifth section, we show the architecture of the proposed tool and propose implementation requirements for the different components. The sixth section concludes the paper with pointers to future work.

2. Related Work

The need to develop web-based distributed data-entry, analysis, and reporting systems has been realized for various applications and by different organizations. Based on the requirements of data entry, analysis and reporting, different approaches have been used. Recently, Choe and Yoo (2008) proposed a secure multi-agent architecture for accessing healthcare information through the Web from multiple heterogeneous data sources. Choe and Yoo (2008) use a multi-agent architecture that supports authorized information access and secure exchange of information based on Web-services. Because security is one of the main features required in their application, Choe and Yoo use a Role-based access control system (RBAC) and an eXtensible Markup Language to enable user authentication, data integrity, and the selective encryption of patient information.

Chassiakos and Sakellaropoulos (2008) proposed a system based on the technologies of integrated databases within Web applications for managing construction information. Chassiakos and Sakellaropoulos (2008) design for relational databases and use dynamic data-driven Web applications that enable users to access data and perform certain transactions.

For our case study, some requirements differ from those presented the related work. It is therefore necessary to identify a suitable development approach for W-PART. However, there are cases where approaches such as those in Choe and Yoo (2008) could benefit the development of our proposed tool considering similar requirements for W-PART.

3. W-PART development approach

A variety of approaches have been proposed for developing web-based applications depending on the application requirements. Some of the common Web development methodologies include: Intranet Design Methodology (IDM) (Lee, 1998), Website based development methodology (Howcroft and Carroll, 2000), Web Application Extension (WAE) (Conallen, 2000), Internet Commerce Development Methodology (ICDM) (Standing, 2002), Object Process Methodology (OPM/Web) (Berger et al., 2002) and Event driven Method Chain (EMC) (Rittgen, 2000).

One major requirement for the proposed W-PART, is to support distributed data entry. Figure 1 illustrates the kind information sources for W-PART. Most of the arrows represent bidirectional data flow. The figure illustrates access to a Web-based application server via the Internet from various sites and using various data input, output, and processing technologies. Another major requirement for W-PART is the

need to ensure secure transfer of productivity data. Currently, there exist a variety of Web-service technologies that can be adapted for this purpose. We expect to have additional requirements for the proposed application for which we are bound to use more than one approach for developing the proposed tool. However, we have observed that an object-oriented framework should be the basis upon which we should integrate recent and suitable techniques for meeting the expected requirements of the application.

Out of a survey of some of the major object-oriented methods for web application development, we have selected a generic object-oriented approach referred to as the Web Composition Component Model (Gellersen and Gaedke, 1999) for developing the proposed tool. The Web Composition Component model is characterized by the properties of object-orientated design including modularity, abstraction, and encapsulation. Moreover, for implementation purposes, it is possible to define the model as an XML-based implementation technology (Gellersen and Gaedke, 1999) that is expected to lead to a seamless and reversible development process.

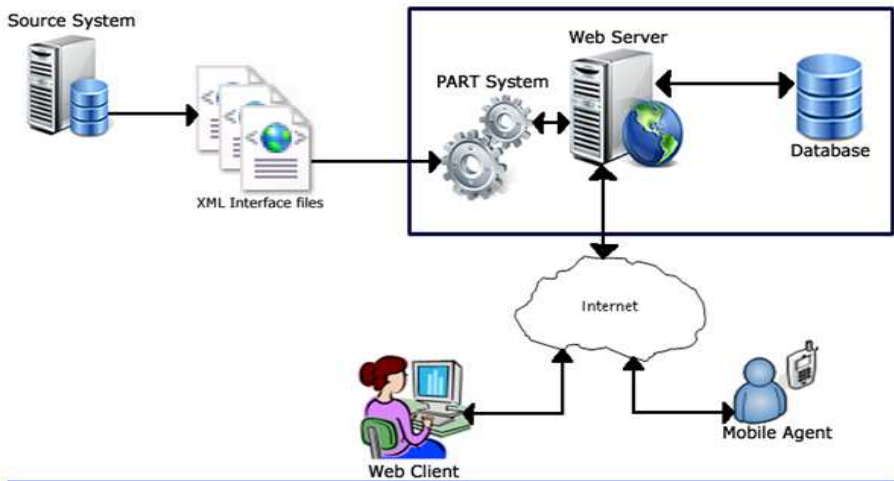


Figure 1: Illustration of Information access via the Internet from different data sources

4. W-Part system requirements

An organization's productivity can be viewed as a relationship between the organization's outputs and inputs during a given period (Lawlor, 1985; Murdick et al., 1990; Sumanth, 1984; Swierczek and Shrestha, 2003). For our case study, Swierczek and Shrestha (2003) provide a concise definition for productivity in the

context of banks while describing generic examples of common inputs and outputs that can be used in analyzing the productivity of a bank. From these examples, we can already draw the expected core functional requirements for our proposed Web-based Productivity Analysis and Reporting Tool (W-PART).

4.1. Input Requirements

According to Swierczek and Shrestha (2003), “the inputs of a bank are the resources consumed in providing products and services.” Generally, there are two categories of inputs: monetary and non-monetary inputs. Examples of monetary inputs include: labor expenses, tangible asset value, operating costs, etc. Examples of non-monetary inputs include: labor hours paid or worked, number of employees, total floor area of business premises, etc. We expect our proposed W-PART to capture data representing all the different types of input that are needed in estimating a bank’s productivity. The system therefore requires an interface which we propose to develop as an independent entity that accepts the different types of input in a ‘form’ that is required by the W-PART system for productivity analysis. By ‘form’ we mean that the interface for capturing the input data uses the same programming language (XML) as that proposed for use in W-PART and also formats the data in a way so as to enable precise estimation of a given productivity metric. Figure 2 illustrates the basic input structure for W-PART. For a proposed prototype of W-PART, we will assume that the measurements for each of the inputs will have been already determined by some functions outside the functional requirements for W-PART. As illustrated in Figure 1, access to data is expected to be from different sources using different electronic devices. We therefore expect W-PART to scale well with heterogeneous systems and hardware that will be used for providing productivity data. We also assume that an organization will be interested in estimating productivity at given times. So W-PART system should be able provide a time stamp for all input productivity data to enable accurate estimation of the organization’s productivity with regard to a particular time frame. It should also be noted that inputs required for estimating productivity can vary to a large extent. As the workings of many organizations change over time, it is expected that new inputs that are necessary for determining the productivity of an organization will also emerge. We therefore expect our proposed W-PART to evolve easily with organizational changes over time. This means that it should be easy to modify the input interface to incorporate capturing of any new inputs that can affect an organization’s productivity.

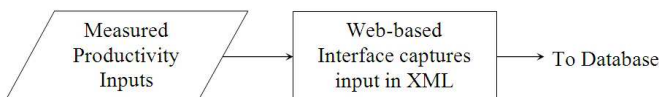


Figure 2: Illustration of basic Input structure for W-PART

4.2. Computational Requirements

W-PART is expected to handle computational procedures associated with various productivity analysis and reporting tasks. The tool is also expected to support the execution of primitive computations for example, summations of productivity values provided per clerk or per department for a given task within a given organization for a given timeframe for which productivity is to be estimated. We use the term ‘clerk’ to refer to any organizational members who are expected to use the proposed W-PART. Most importantly the tool should have procedures for determining the different metrics that are necessary for reporting on the organization’s productivity. Just like in the case of expecting new inputs over time for the input requirements, we also expect new productivity metrics to emerge. We therefore expect that W-PART can again be easily modified to support new computational procedures for estimating the new productivity metrics. To support this kind of feature, the system is expected to enable the definition of an arbitrary number of varying functions that represent the different metrics and also enable an easier way to factor in the new function while incurring very limited time and cost during modification. Figure 3 is an abstract representation of the computational structure for W-PART. In Figure 3 we have a mapping component that defines various mapping options based on

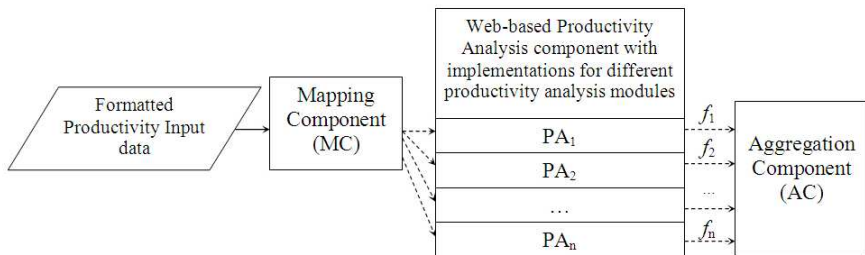


Figure 3: An abstract model for W-PART's computational structure. PA_i represents the i th Productivity Analysis module where $i = 1 \dots n$ for n PA modules. f_i represents the i th factor used by the Aggregation Component (AC) to factor in the metric provided by the i th PA module.

the type of input productivity data that it receives and maps the data to an appropriate one of n Productivity Analysis (PA) modules. Each of the PA modules implements a function for computing a particular productivity metric. Based on our proposed development approach, the PA modules should exist as separate objects within a Web-based Productivity Analysis component. The Aggregation Component (AC) implements a function for determining an estimate of an output based on parameters f_i ($i = 1 \dots n$) for PA metrics provided by any of the PA_i components. Such a function can be a summation of the multiples of the productivity metrics and their parameters for estimating a given productivity output:

For any proposed prototype of W-PART, we will assume that parameters required in the aggregation component exist and the only requirement for the W-PART system is to allow for their specification and modification.

4.3. Productivity reporting Requirements

The proposed W-PART system should transform results from the Productivity Analysis component into knowledge about an organization’s status. For example, the system should enable the determination of trends, patterns, and predictions. A user of the system should be able to obtain this information through a Web-based Output Interface. It is expected that such information may be of interest to other sites of the organization; therefore the ability to acquire this information from remote sites using the Web is expected to be of benefit. Figure 4 illustrates the output structure for the proposed W-PART system. In Figure 4, we expect the tool to utilize statistical functions that generate information in a manner that appropriately describes the productivity outputs of a given organization. For the W-PART we refer to such functions as Productivity Report Generation (PRG) functions. The user of the W-PART system will be interested in generating information about a particular productivity output with regard to the organization. W-PART should be capable of determining the particular PRG function to use for providing information that the user requires. We also envision that new outputs will emerge for measuring the productivity of a given organization. Just like in the requirements described in the preceding two subsections, we expect W-PART to allow for easy modification for integrating any new output productivity measures.

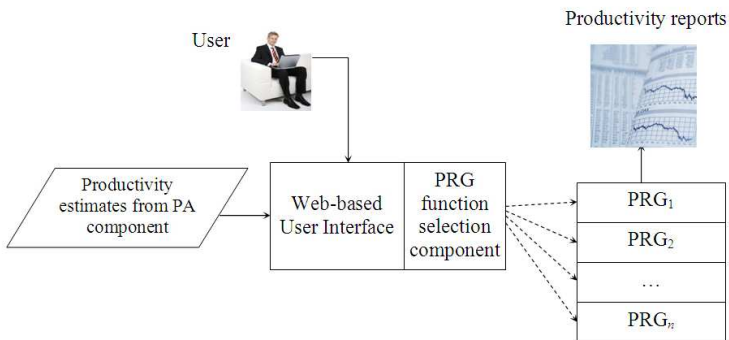


Figure 4: Illustration of output structure for W-PART. PRG_i (for $i = 1 \dots n$) represents the i th Productivity Report Generation function

5. W-Part Architecture and Implementation Requirements

Figure 5 illustrates the overall architecture for W-PART. The architecture is subdivided into three major top level components, a productivity data entry component, a productivity database management system, and a Productivity Analysis (PA) and Productivity Report Generation (PRG) component. In the following subsections we describe each of these components and propose how they should connect to each other to form a complete system that can be used for analyzing productivity data and generating reports out of the analyses. General implementation requirements for the components are also discussed.

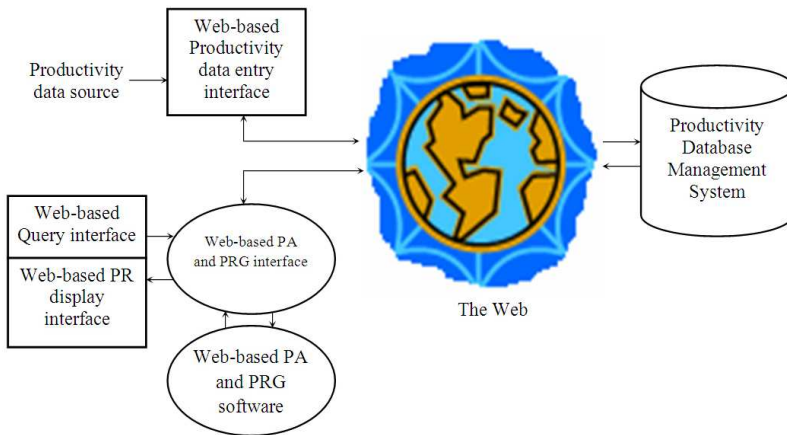


Figure 5: Proposed W-PART Architecture.

5.1. Web-based productivity data entry interface

Existing Internet standards such as SOAP and WSDL (Web Service Description Language) are expected to benefit interoperability between different software applications on the different data entry devices. We expect the data input to be structured or semi-structured in nature and for that we propose an XML-based data capture interface be implemented following a suitable XML implementation method. We propose to develop a user interface model that can be easily implemented on a variety of data entry devices. We intend to use the User Interface Markup Language (UIML) (Abrams et al., 1999) to enable an easy and automated specification of the data entry interfaces. Recently UIML has been adapted to enable the definition of, a data model, service model, and navigation model which allows data communication from one User Interface to another (Iñesta et al., 2009).

We also know that for the productivity analysis functions, we expect to perform computational operations, some of which could be complex. We therefore propose to use XForms to simplify the process of transfer of XML data from the Web-based data entry interfaces to the Productivity DBMS.

5.2. Productivity Database Management System

An abstract discussion of a DBMS for handling productivity data is given in this section. Depending on the application domain, the type of productivity data that should be handled by the system will vary. For some domains, the data required for productivity analysis will increase in complexity. A DBMS that scales well with the various requirements is proposed. Four major categories of DBMSs exist: File System DBMSs; Relational DBMSs, Object-oriented DBMSs; and Object-Relational DBMSs. Of these categories, the Object-Relational approach is becoming popular in recent applications and is mainly aimed at dealing with complex data requirements. We currently propose implementing an object relational DBMS that should allow for data input and access from heterogeneous interfaces. We propose to evaluate existing Object-Relational DBMS implementation approaches in future work to enable select one that is suitable for the application in this paper.

5.3. PA and PRG component

There are various web-based implementation options that we can consider for analyzing productivity data and generating reports. One main requirement, however, is that the online processing cost should be as minimal as possible and the productivity report generation and display should also easily be fast enough in places with minimal Internet bandwidth. Again as described in section 5.1, UIML is proposed for specifying output user interfaces for the variety of devices that may be used. The PA component will obviously comprise software designed to perform computations as described in section 4.2 and 4.3 above. The software used at this stage is expected to interface with the DBMS based on a given query or set of queries, and use the specified implemented functionality to make computations on the data. At this point, we consider using a Web-based object-oriented programming language that will ensure lighter implementations of the PA and PRG functionalities, but also ensure faster online execution of these functionalities. For accessing productivity data from the DBMS, we have a number of options for implementing database connectivity to the DBMS. For the PA functions, there also exist a variety of Web-based programming languages that can be employed including Java technologies and C#. For the PRG component, we could either use one of the Web-based programming languages, but depending on the complexity of report generation functionality, there is a variety of Web-based statistical software that could be easily adapted.

6. Conclusion and Future Work

A model of a Web-based Productivity Analysis and Reporting tool has been proposed. This model is expected to open up different avenues for implementing the proposed W-PART system depending on available Web implementation technology. We have also made available an implementation proposal for a prototype that can be adapted in future. It should also be interesting to investigate the extent of applicability of the proposed W-PART system on different application domains.

REFERENCES

- ABRAMS, M., PHANOURIOU, C., BATONGBACAL, A., WILLIAMS, S.M., AND SHUSTER, J.E.
- BERGER, I.R., DORI, D., KATZ, S. 2002. *OPM/Web – Object-Process Methodology for Developing Web Applications*. *Annals of Software Engineering* 13(1-4), 141-161.
- CHASSIAKOS, A.P. AND SAKELLAROPOULOS, S.P. 2008. *A Web-based System for Managing Construction Information*. In *Advances in Engineering Software*, 39(11):865-876.
- CHOE, J. and YOO, S. 2008. *Web-based Secure Access from Multiple Patient Repositories*. In *International Journal of Medical Informatics*, 77 (4): 242-248.
- CONALLEN, J. 2000. *Building Web Applications with UML*. Addison Wesley, Boston, MA.
- GELLERSEN, H-W. AND GAEDKE, M. (1999). *Object-oriented Web Application Development*. *IEEE Internet Computing*, 3(1): 60-68, IEEE Educational Activities Department.
- HOWCROFT, D., AND CARROLL, J. 2000. *A Proposed Methodology for Web Development*. In *Proceedings of the European Conference on Information Systems, Vienna*, 290-297.
- IÑESTA, L., AQUINO, N., AND SÁNCHEZ, J. (2009). *Framework and authoring tool for an extension of the UML Language*. In *Advances in Engineering Software*, 40(2009): 1287-1296.
- LAWLOR, A. (1985). *Productivity Improvement Manual*. Westerport, CT: Quorum Books.
- LEE, S.C. 1998. *IDM: A Methodology for Intranet Design*. In *Proceedings of the International Conference on Information Systems, Helsinki, Finland*. Association for Information Systems, 51-67.
- MURDICK, R.G., RENDER, B., AND RUSSEL, R.S. (1990). *Service Operations Management*. Boston, Massachusetts: Allyn and Bacon.
- RAO, M.P., MILLER, D.M., AND LIN, B. 2002. *PET: An Expert System for Productivity Analysis*. *Expert Systems with Applications* 29, 300-309.
- ROCKVILLE, MD: FDA, 2002 “*General Principles of Software Validation: Final Guidance for Industry and FDA Staff*”
- SCHWABE, D. AND ROSSI, G. 1998. *An Object Oriented Approach to Web-Based Application Design*. Draft copy, unpublished.

STANDING, C. 2002. *Methodologies for Developing Web Applications*. *Information and Software Technology* 44 (2002), 151-159.

STEVENS, T. 1998. *Chief Among us*. Cover story: 27th Annual CEO Survey. *Industry Week* November 16, 24-30.

SUMANTH, D. J. (1984). *Productivity Engineering and Management Productivity Measurement, Evaluation, Planning, and Improvement in Manufacturing and Service Organizations*. New York: McGraw-Hill.

SWIERCZEK, F.W. AND SHRESTHA, P.K. (2003). *Information Technology and Productivity: A Comparison of Japanese and Asia-Pacific Banks*. In *Journal of High Technology Management Research*, 14 (2003): 269-288.