Before you invest: An illustrated framework to compare conceptual designs for an enterprise information system

Arif, M, Kulonda, D, Proctor, M and Williams, K

Information Knowledge Systems Management 4 (2004) 119–135
IOS Press

Before You Invest: An Illustrated Framework to Compare Conceptual Designs for an Enterprise Information System

Mohammed Arif[a,*], Dennis J. Kulonda[b], Michael Proctor[b] and Kent Williams[b]
[a]*The British University in Dubai P.O.Box 502216 Dubai, UAE*
[b]*Department of Industrial Engineering and Management Systems, University of Central Florida, 4000 Central Florida Blvd, Orlando, FL, USA*

ABSTRACT: Post-implementation analysis on Enterprise Resource Planning (ERP) systems has drawn attention to many structural shortcomings. Yet, no framework exists to compare the different structural features of the ERP system. This paper develops a framework to compare different enterprise-wide systems at the conceptual design level using size, coupling and architectural complexity as criteria. Since, metrics used to measure these criteria are subjected to individual interpretation, a statistical technique using repeated measures design is used to validate the results of multiple evaluators. The framework was applied to the comparison of two enterprise-wide system implementations at the conceptual design level. One was a typical ERP, and the other was a document- based system. A conceptual model was developed for the two methodologies using Unified Modeling Language (UML). Ten evaluators, all graduate students with the knowledge of UML were given the conceptual models of both systems and were instructed to apply the metrics. The evaluators performed the evaluations separately and were under no time restriction. Their results were used in the repeated measures design. Based on the results, TDM was smaller in size, more loosely coupled and less complex as compared to the ERP model. The framework successfully demonstrated that it can differentiate between two different implementations on the basis of their size, module coupling and architectural complexity. This framework presents a quantifiable technique that helps in informed decision making prior to a major financial commitment.

KEYWORDS: ERP, systems design, software metrics, UML

INTRODUCTION

We live in an era of global competition. In order to sustain a competitive edge, enterprises must adopt appropriate improvement schemes. Information Technology has been an undeniable source of productivity gains through the 1990's. Through effectiveness and efficiency of operation, Information Technology has also enabled market growth. Two widely used tools have been Business Process Re-engineering (BPR), and Enterprise Resource Planning (ERP) [Ng et. al., 1999]. Although evolving from different sources, these two have since become inter-wound in the search for operational excellence. According to Ng et. al., (1999), Business Process Re-engineering (BPR) has provided the vision through which companies have kept pace with change and Enterprise Resource Planning (ERP) has been one

*Corresponding author. Tel.: +1 971 4 391 3626; Fax: +1 971 4 366 4698; E-mail: mohammed.arif@buid.ac.ae or marif@carthage.edu.

of the chief enablers (Grover and Malhotra, 1997). Hammer and Champy (1993) define BPR as "the fundamental rethinking and radical redesign of business processes to achieve dramatic improvements in critical, contemporary measures of performance, such as cost, quality, service, and speed". In contrast, ERP systems are prepackaged software driven solutions developed to integrate business functions. They do so through transaction processing systems that cut through all functions, and update a common corporate database.

ERP software includes prepackaged best business practices in many different industries in the form of blueprints. The business blueprint defined in the ERP acts as a starting point for the BPR effort. The business blueprint concentrates on the following four key areas necessary for understanding business: events, tasks or function, organization, and communication. These areas define who must do what, when, and how. Events are the driving force behind a business process, prompting one or more tasks to take place. By connecting events and tasks, the analysis of even very complex business processes become possible (Curran and Kellar, 1998). However, this ability to handle complexity comes at a significant cost (Kulonda, 1999).

The information system in an ERP based solution is modeled according to established business strategies and processes. Difficulty arises when change in business activities necessitates change in the supporting information system. With an ERP system, implementing a change in the supporting information system is cumbersome and expensive. For example, communication links may need to be re-defined and once the links are re-defined; there are no guarantees as to how long they will be valid due to frequently changing market strategy (Harrold, 2000 and Larsen and Myers, 1999). A Larsen and Myers (1999) case study of a financial company indicated that the company had to abandon its entire ERP system because the system was unable to accommodate process changes imposed on it due to a merger. Further, the time required to implement an ERP system can range anywhere from 12 to 130 months (The Conference Board, 1998 and Kulonda, 1999). Unfortunately, in today's technology driven market place, this reaction time is unacceptable. This long implementation time can be attributed to the required business process re-engineering and the large size of the ERP system (Kulonda, 1999).

Kulonda (1999), Larsen and Myers (1999), and Agarwal et. al., (2000) also point to the inflexible nature of the ERP system. The ERP software is divided into different modules. But, during the implementation, these modules have to be connected to facilitate the information flow in the system. Once the system is implemented completely, the coupling resulting from these inter-module connections makes the system rigid and inflexible to an extent that it can almost be regarded as a monolithic system. ERP even though fraught with all these shortcomings, still performs a valuable function of integrating the information flow in an organization and eliminating redundancies as well as discrepancies which may result in major financial losses.

This research uncovered an alternative approach to ERP development which employs a document-based approach to design and construction of enterprise-wide systems. Jones (1999) articulates the philosophy of this approach, and a methodology for applying its axioms and concepts. This methodology hereafter referred to as The Document Methodology (TDM) is intricate. It recognizes documents as data containers or objects that both record and drive the processes of the company. The methodology is also comprehensive. It encompasses knowledge documents and management documents as well as the transaction documents that are processed in the data-based packaged ERP systems. Because of these fundamental differences in structure, an enterprise-wide system using TDM is very different than developing one from a packaged ERP approach. The resulting systems solutions may vary in terms of size, changeability and complexity. Jones (1999) never compared TDM to ERP. This paper describes an approach to compare these alternative platforms and choose between them.

The choice of the enterprise information system is necessarily made at the conceptual design stage of the project. The information available to compare alternatives at this level of detail is limited. Certainly, cost can be estimated based upon the size and scope of the effort. Benefit estimation is fraught with difficulties as most of the benefits result from reengineering, rather than emanating from the software itself. Issues like the size, the architectural complexity, and the changeability of the software are treated as intangibles, if considered at all. To overcome these decision problems, this paper develops a framework to evaluate enterprise information system solutions at the conceptual design level that can be used in conjunction with the traditional cost-benefit approaches and will address the information age concerns for the size, the architectural complexity, and the changeability. This framework will be used to compare the traditional ERP system approach with one suggested by TDM. This framework can also be used to compare any two system implementation approaches.

To develop conceptual designs using the ERP approach and TDM, a test case will be used. This test case is described in the next section. In Section 3 the selection of software metrics for the comparison of the conceptual designs is made. Section 4 describes the development of the solutions using the two approaches. Section 5 describes an experiment used to validate the results obtained after the application of the metrics. Section 6 summarizes the results of the experiment. Section 7 analyzes the results of the experiment. Section 8, summarizes the research and discusses the results and their implications.

A TEST CASE

For the purpose of this research, operations of a mobile home manufacturer, hereafter referred to as Mobile Homes Inc., were modeled. One of the major aspects considered during the selection of the test case was the diversity of document formats needed to run the operations in this plant. There are CAD drawings from engineering, an accounting database, pictures describing the process, and memos in the document format.

Mobile Homes Inc. has about 200 employees and is divided into seven departments; sales, purchasing, engineering, production, quality control, service, and human resources. The entire operations of the company are divided into fifteen processes described in Table 1. The entire operation involves 101 documents. These documents are either generated or updated during one or mores of these processes. Some parts of the operations are automated and involve databases. Other non-database, non-electronic documents are produced manually on paper.

METRICS

Based on the concerns identified in Section 1, additional metrics beyond traditional cost-benefit measures were developed that addressed the system size, the changeability and the scalability. The technical size of the system hereafter referred to as the system size translates into the implementation effort during the implementation. Difficulties associated with the changeability and the scalability results due to the monolithic nature of the software. Both the effort needed and the monolithic nature has been detrimental to the implementation process as has been highlighted in Section 1.

System Size

Traditionally there are three types of metrics used in software engineering that estimate the size of an information system. They are Lines of Code (LOC) metrics, Case Based Reasoning (CBR) metrics, and functions based metrics.

Table 1
Processes and Objectives for Mobile Homes Inc.

Process	Objective
Ordering process	This process involves customer's order acceptance and internal review to provide him/her with an estimate of cost and delivery date.
Engineering review	This process contains the preparation of drawings and PE approval.
Purchasing review	This process involves checking the inventory status, the material cost and expected lead time in case some materials have to be ordered.
Production review	In this process the current production schedule is reviewed and an expected production date is communicated.
Order confirmation	Orders are confirmed when the customer agrees to the delivery date and cost. The sales department informs the engineering, the purchasing and the production department to start the production.
Production	This process includes the reporting and compilation of quality control data during the production of the house.
Post-production inspection	This process includes the reporting and compilation of the quality control data gathered after the house leaves the production floor and is awaiting shipment in the staging area.
Production change order	This process includes any changes requested by the customer after the start of the production process. The impact on cost and schedule of the change is assessed and the customer is informed about the changes. If the customer agrees then the changes are made and the database is updated.
Accounts receivable (house payment)	This process covers the receipt of the house payment from the customer and the updating of the accounts after the payment has been received.
Hiring	This process involves capturing information for resume review and interviewing process of candidates.
Production injuries	In case of any production injuries the information regarding the type and cause of the injury, physicians report on the type of the injury, and report filed with OSHA is captured in the information system.
Timesheet	This process involves capturing information in terms of the hours worked and dollars earned.
Accounts receivable (service)	When some service is performed on the house, the accounts have to be updated after the payment is received for the service. This whole information is captured in this process.
Accounts payable	The suppliers have to be paid for the raw material supplied to the manufacturer. The financial transaction is captured in the accounts payable process.
Budgeting	Every year the budget is finalized based on the sales forecast, expected expenses and expected profits. All these transactions are covered in the budgeting process.

LOC based metrics COCOMO (Boehm, 1981), SLIM (Putman, 1978) and Meta Model (Bailey and Basili, 1987) measure the size of the system based on the lines of code. As it is not possible to count the lines of code until the code has actually been written, this metric is not suitable for predictive estimation purposes (Moser et. al., 1999). For this research, a metric is needed that can estimate the effort needed based on the conceptual system design only. This stage precedes the coding stage in the project. Therefore, none of the LOC based metrics can be used in this research.

CBR metrics ANGEL (Shepperd and Schofield, 1997), and ESTOR (Mukhopadhyay et. al., 1992) identify specific project, or cases, in the database that are similar to the project to be predicted (Stensrud, 2001). CBR approaches can only be used if a similar past case is stored in database memory. Since this research considers only one model, and no past estimate is available, it is not possible to use these CBR metrics.

Functions based software metrics use a measure of functionality delivered by the application as a normalization value. One of the function based metric methods is referred to as the function point method. Albrecht (1979) first suggested the function point method. Function points are derived using an empirical relationship based on countable measures of software's information domain and assessments of software complexity. Function points are calculated by counting the following five information characteristics (Jones, 1991): 1] Number of user inputs; 2] Number of user outputs; 3] Number of user inquiries; 4] Number of files, and 5] Number of external interfaces. Each parameter is assigned a weight based on whether it is perceived as simple, average, or complex by the modeler. A Total Count (TC) is obtained by multiplying the count by the weight. For a detailed explanation of function points refer to IFP (1994).

A key development which makes the function point method a viable means of estimating information system size is the Unified Modeling Language (UML). UML is a technique that can be used to model an information system with such detail that function points can be determined without having to actually code the system (Schow, 2001). Thus for our research the function point method was the only means by which the size metric could be estimated at the conceptual design stage. Therefore, this metric is used to compare the respective sizes of the competing systems.

Changeability and Scalability

As indicated above, ERP systems differ from TDM systems in terms of changeability and scalability. Metrics need to capture the effects of those differences. Typically, the ERP systems are monolithic in nature resulting in a close coupling between the modules, whereas the TDM systems loosely coupled. Further, the all encompassing nature of an enterprise information system (whether ERP or TDM) generates architectural complexity. The two metrics in the literature found appropriate for capturing these differences are module coupling (Dhama, 1995) and module architectural complexity (Card and Glass, 1990).

Module Coupling

As was highlighted in the previous sections, one of the major reasons for ERP failure is its rigid structure. The modules used in an ERP system are very tightly coupled. An alternative to ERP should be less rigid. The coupling metric (Dhama, 1995) measures the rigidity of a modular system. It measures the coupling of each module with the rest of the system; the lower the score, the looser the coupling between the modules of the system. Dhama (1995) describes a good system as one with a higher score for coupling metrics because this lowers the possibility that a change in one module will cascade to a module coupled to it. This metric considers the following eight parameters of the system: 1] Number of Input Data Parameters; 2] Number of Input Control Parameters; 3] Number of Output Data Parameters; 4] Number of Output Control Parameters; 5] Number of Global Variables Used as Data; 6] Number of Global Variables Used as Control; 7] Number of Modules Called (Fan-Out), and 8] of Modules Calling the Module (Fan-In). For a detailed description refer to Dhama (1995).

For the purpose of this research, comparison of the average coupling metrics [mcavg] of all the modules in a system will be made. To achieve a loosely coupled system, one with a lower average coupling metric is desired.

Module Architecture Complexity (*MAC*)

MAC (Card and Glass, 1990) measures the architectural complexity of each module and its contribution to the architectural complexity of the whole system. MAC for a module is a sum of data complexity and system complexity of the module. Data complexity provides an indication of the complexity in the internal interface for a module and is a function of number of input and output variables passing through the system, as well as number of modules called. System complexity provides an indication of the contribution of the module to the complexity of the whole system, and is a function of number of modules called.

For this research, the average MAC for the two systems will be compared. A lower value of average MAC will indicate an architecturally less complex system.

Previous sections discussed the issues with the ERP system. Based on the issues highlighted in the literature three metrics were selected as possible criterion for evaluating enterprise wide systems at the conceptual design. The next section develops two enterprise wide solutions for comparison at the conceptual design level.

ALTERNATIVE SOLUTIONS FOR THE TEST CASE

The following subsections briefly describe the development of the alternative solutions for the test case using two different methodologies. For this research the solutions were developed by one of the authors of this paper. The process followed is described in the following sections.

TDM Based Solutions

Figure 1 displays the process followed to develop the conceptual design of the information flow model for the TDM method. The first step was to compile the list of all the documents. The documents were classified into one of the following five categories: 1] Knowledge Documents; 2] Transaction Documents; 3] Management Documents; 4] Capability Documents; and 5] External Documents (Jones, 1999). Detail descriptions of these documents in provided in Table 2. After classifying all documents in the above mentioned five categories, redundant documents were then eliminated. After the elimination of redundant documents, the input and output documents for all the processes were listed. All the documents were then connected in the order in which they contribute to the operations of the company. This was done in order to gain more insight into the contribution of the documents to the processes. Once all the connections and contributions of all the documents were established, the conceptual design of the information flow model using the TDM method was created.

The goal of TDM is to structure the information required to manage people after defining the role of each document in the enterprise by generating the following information: 1] Expectations about the performance of people and processes; 2] Metrics for evaluation; 3] Statistics of the existing performance and opportunities for improvement, and 4] Facilitation of continuous improvement for the system through the available statistics.

TDM also generates architectural specifications for the distributed heterogeneous computer network to support electronic documents and provides the framework to assemble integrated systems from existing and commercial off-the-shelf software. TDM connects business processes through the complete value chain by explicitly defining processes in a workflow system that permits limited access to documents in a document management system. Figure 2 illustrates the software requirements for Mobile Home Inc. Software chosen can be divided into four categories: office-ware, the CAD package, the database

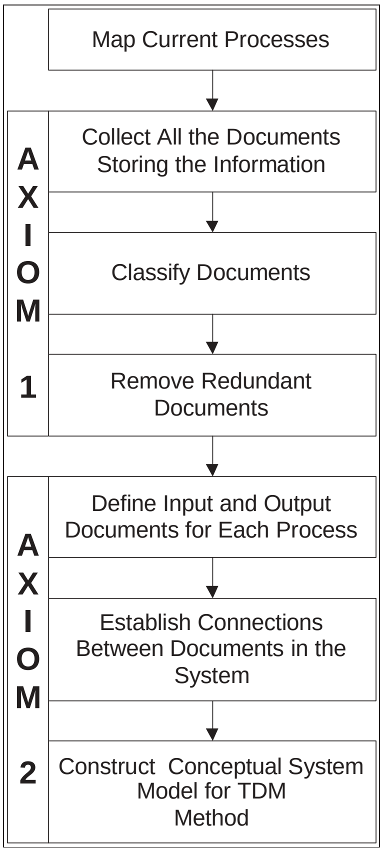

Fig. 1. Conceptual system design steps using the document methodology.

package, and the document management package. Office-ware includes packages for word processing, spreadsheet creation, email, and presentation creation. The CAD package is the software used by the engineering department for the creation of drawings, by manufacturing to fabricate houses, and by sales to show customers the layout of houses. The database package is used for creation of databases throughout the enterprise. All these software packages are connected using a document management system, Docushare^{TM} in this case. The roles played by Docushare^{TM} are as follows: 1] The first role it plays is that of the workflow system where the information regarding the flow of documents is defined, as well as the initial, intermediate, and final steps of the processes are defined in terms of document flow with each milestone in the process representing creation, updating, or archiving of a document; 2] The second role played by the document management system is that of a manager which provides the archiving and searching capabilities, report generating capabilities, user interfaces, and the format interchange capabilities between different formats of the documents; 3] The third role played by the document management system is that of a vault which contains the templates for different documents, the reference links, the document itself, and any alternate views of the document; and 4] The fourth role played by the document management system is the metadata storage of the documents which includes a dictionary of the documents, an index with key words, and abstracts of the documents.

TDM provides a framework to facilitate the implementation of future changes. The change process is accomplished using the following four steps: 1] Planning continuous improvements using the insight

Table 2
Types of documents and their functionality

Documents	Function
Knowledge document	While all the documents contain knowledge, knowledge documents contain the specific knowledge that the enterprise uses to create unique value for its customers in the form of products and services. Product drawings, routings, process instructions, quality testing instructions are examples of knowledge documents.
Transaction documents	Transaction documents contain the knowledge that initiates processes and records the performance of the processes. Work orders, time cards, production counts are examples of transaction documents.
Management documents	Management documents contain the strategic direction and operational facts to direct a business process. Production plans, budgets, staffing plans, advertising campaign plans are examples of management documents.
Capability documents	Capability documents identify the productive capacity of the enterprise machines, people, facilities, external resources, etc. Capacity plans, vendor blanket agreements are examples of capability documents.
External documents	External documents contain information about the enterprise's market environment and are created and maintained outside of the enterprise. Industry forecasts, pending legislation, commodity futures price reports are examples of external documents.

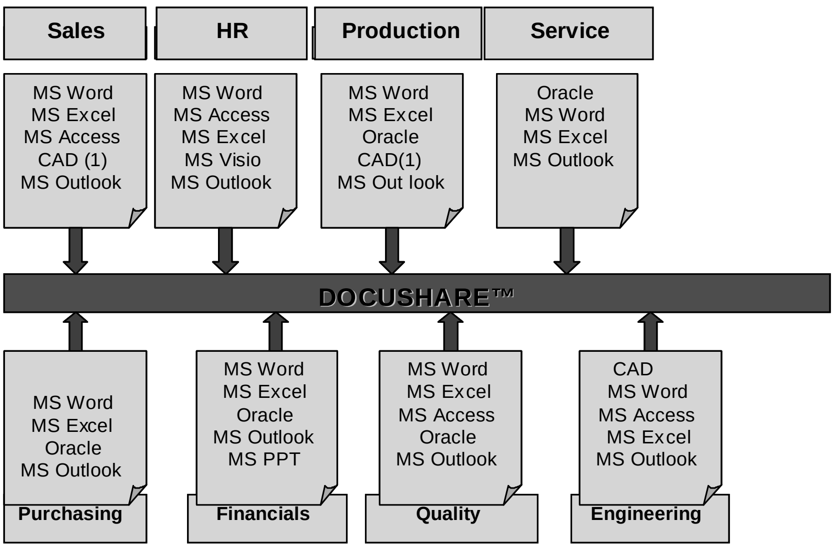

Fig. 2. Software links for the TDM based system.

provided by the documents; 2] Analyzing flow, variation and bottlenecks using the document process models which define the business environment; 3] Developing rapid prototypes with COTS using the document object models, and 4] Deploying explicit definition of processes with document technology allowing real-time auditing to provide summaries that indicate how to reduce process variation, complexity, and cycle times. These four steps present a formal process of implementing change in an organization. Therefore the enterprise-wide system is capable of changing and improving continuously.

ERP Implementation

SAP R/3 was selected as the ERP system to be compared against the document-based system. The selection was made due to extensive documentation available in the public domain for the SAP. SAP R/3 is divided into three modules; logistics, financial accounting and human resources. Each module could be implemented as an independent system, or as part of a system containing other modules based on the requirements of the corporation. The functions performed by each module are as follows:

1. Logistics – Sales and distribution, materials management, production planning, project system, quality assurance, and plant maintenance.
2. Financial Accounting – Finance, assets accounting, investment management, and controlling.
3. Human Resources – Personnel administration and payroll, and personnel planning and development.

The development of the ERP based model was a five step process as shown in Fig. 3. The first step was the documentation of the current processes. The next step was the identification of the data elements in the process. To identify the data elements all the information available in the system was broken down to the data level. ERP systems have predefined repositories of data. Keller and Teufel (1998) define in detail, the repositories of information for the SAP R/3 package. Each of these repositories is part of one of the modules in the SAP R/3 package. Based on this information the repositories for different data elements were identified. Curran and Keller (1998) describe the different processes and the sequence of transactions in detail. This description is called the process blueprint. Based on these descriptions, the process flows, for Mobile Home Inc., were developed. Data elements were connected to capture information accessed, modified and saved during the process. After the connections were drawn a UML model was developed to represent the flow of information in the system.

EXPERIMENTAL DESIGN

Kemerer (1993) identifies two shortcomings of function points. One of the shortcomings is that the measurement of function points cannot be automated, i.e. the measures cannot be extracted automatically from design documents. The second issue is that function point analysis is prone to variation as a function of the individual differences of the people who perform them. The same argument can be made for the other two measures, the module coupling and the MAC. Therefore, it is necessary for the purpose of this research, that multiple evaluators calculate the metrics to take into consideration the reliability of the resultant measures. Data collected from all the evaluators will then be subjected to further analysis using an analysis of variance technique.

For the analysis, the two factors repeated measures analysis of variance design will be used. One factor was the type of the system, either typical ERP or TDM and the other factor, the evaluator. Repeated measures design treats evaluators like any main effect. This isolates the variability associated with the effect, and consequently, variability in individual differences do not inflate the error variance in the design (Myers, 1967). Thus, conclusions can be drawn about the differences in the two systems after eliminating the effects of the personal biases of the evaluators. The experimental setup is defined in detail in following sub-sections.

Objectives

Some factors in the three system structure metrics are subjected to individual interpretations (Kemerer, 1993). Therefore, it is necessary, for the purpose of this research, that multiple evaluators should

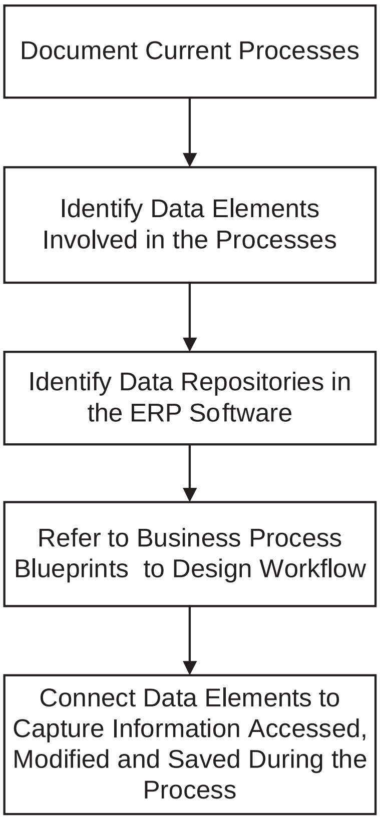

Fig. 3. Modeling process for ERP (SAP R/3).

calculate the metrics. Data collected from all the evaluators will be subjected to further analysis. For analysis, the two factors repeated measures design will be used. One of the factors will be the type of design (TDM or ERP), and the other factor will be the evaluator. Since in a repeated measures design the evaluators are treated like any main effect, the variability associated with the effect can be isolated, and as a consequence, the error variance in this design is not inflated by variability due to individual differences (Myer, 1967). Thus, conclusions can be drawn about the differences in the two systems after eliminating the effects of personal biases of evaluators.

Since the model generated using the two methodologies were large in size and would take a long time to evaluate, it was decided to select seven out of fifteen processes from both the methodologies. The same seven processes were selected for both systems. The reason for selection of these processes was their utilizing multiple formats of documents e.g. database, spreadsheet, CAD drawings, and memos. The eight processes not selected were more administrative in nature and involved database application only.

The data collected from evaluators is used in repeated measures design. The evaluation will be performed at 95% confidence level. If the p-value for the within subject effect is less than 0.05, it indicates that there is a difference in the magnitudes of the metrics. To evaluate the difference the means of the magnitudes for the metrics will be taken. The difference in magnitude will indicate the difference in quality. The system with a higher function point will be bigger in size. The system with lower average module coupling will be more loosely coupled. The system with a higher value of the average module complexity will be more complex.

Evaluators

Ideally certified instructors should have been used for this experiment. However, due to lack of funding, and lack of willingness on part of certified evaluators to work free of charge, we had to use graduate students in the college of engineering and computer science at the University of Central Florida (UCF). It should be kept in mind that certification from International Function Point Institute (IFPI) does not guarantee same results for measurement. All it will do is reduce the variability. Evaluators for this experiment were selected from a group of volunteers familiar with UML and the three software metrics. Out of the evaluators eight were males and two were females. All of them had used UML in at-least one of their classes. They were provided food and beverages after the evaluation. Each evaluator performed the evaluation separately. There was no time limit. The evaluators took anywhere from four to nine hours to complete their evaluation.

Material Supplied

Each evaluator was given two packages one package contained the UML representation (Use Case Diagrams and Sequence Diagrams) of the ERP based solution, and the other package contained the UML representation (Use Case Diagrams and Sequence Diagrams) of the TDM based solution. The solutions were developed using methodologies described in Sections 4.1 and 4.2. One package contained the TDM model, and the other contained the ERP model. As mentioned in the previous section, only seven out of fifteen processes performed by the information system were selected. For further clarity a graphical representation of module boundaries were clearly demarcated. It was ensured that the same processes were performed in both the TDM, and ERP segments of the experiment. Evaluators were not aware of the identity of the packages. To review these packages please contact the leading author of this paper.

Procedure

Each evaluator had to evaluate both designs. If evaluator felt that there was a possibility of multiple interpretations in a scenario, they were asked to assume whatever they felt was correct, as long as they were uniform in their assumptions in both the designs.

It was realized that there would be a learning curve associated with the evaluation process. Therefore, the order in which the designs are evaluated might have an impact on the outcome. To neutralize this effect, the experiment was counterbalanced. Half of the evaluators evaluated TDM first, and the other half evaluated ERP first. The overall system in the case of ERP was divided into the three modules logistics, human resources and financial accounting. The TDM model was divided into seven modules based on the processes performed. The modules were ordering process, production review, engineering review, purchasing review, hiring, order confirmation, and production timesheet. The evaluators' responses contained function points, module complexity, and module coupling for the two systems. From their responses, average module complexity, and average module coupling for the two systems were calculated.

RESULTS

The data obtained is tabulated in Table 3, and as can be seen had a high degree of variability. There are several factors that contribute to this variability. In function point one has to count the number of user

Table 3
Evaluation result

	Function point		Average module coupling		Average module complexity	
	JDM	ERP	JDM	ERP	JDM	ERP
Evaluator 1	807.3	1020.6	0.041571	0.02503	9.248571	22.28667
Evaluator 2	992.25	1277.1	0.030486	0.029057	23.11429	30.33333
Evaluator 3	1096.2	1331.1	0.043679	0.020667	11.71429	29.19
Evaluator 4	1046.25	1189.35	0.044597	0.025	13.64286	15.55333
Evaluator 5	893.7	1067.85	0.017984	0.007947	25.89286	51.23333
Evaluator 6	1077.3	1314.9	0.040423	0.025417	15.92857	14.16667
Evaluator 7	1058.4	1301.4	0.038521	0.027333	15.5	15.23333
Evaluator 8	842.4	1047.6	0.046257	0.04005	8.81	27.65333
Evaluator 9	1377	1506.6	0.03532	0.0175	19.60714	21.33333
Evaluator 10	1394.55	1512	0.0347	0.016667	20.31429	22.03333

Table 4
Summary of metric measurement

	Mean		Standard deviation	
	TDM	ERP	TDM	ERP
FP	1058.50	1256.90	199.354	175.944
AMC	0.02	0.04	0.009	0.008
MAC	16.38	24.90	5.028	6.493

inputs, user outputs, user inquiries, files, and external interfaces. Owing to the large size of two cases the final tally of these variables was different for different evaluators. Apart from the count, the evaluators were also required to assign a weighting factor based on their perception of the complexity. Evaluators chose different weighting factors based on their perception of the system complexity. However, this variability was expected from this experiment due to reasons mentioned in Kemerer (1993). It is due to this variability that we use repeated measures design in the following sub-section to validate the results.

ANALYSIS

For analysis, the two factors repeated measures design was used. Significant variability was observed in the results. This variability was due to the individual assumptions made by the evaluators. The repeated measures design analysis was performed for an alpha of 0.05. The means and standard deviation for the measurements from surveys is tabulated in Table 4.

After analyzing the data for the between subjects evaluation, the p value was found to be 0. Since the significance of the within group comparisons is 0 (< 0.05) it indicates that at 95% confidence level there is a difference between the function points of TDM vs. ERP. To find which has a lower function point measure, the mean of the ten evaluator's readings was taken. The means are represented in Fig. 4. The figure indicates that the TDM system has a function point score of 1058.50 as compared to the function point score of 1258.90 for the ERP system. Therefore, it can be concluded that TDM is a smaller system in size than the ERP, for this test case. Therefore, the programming effort needed to implement the TDM will be less than that required for ERP. Therefore the TDM system is less complex. This factor will be further confirmed after measuring the MAC.

After analyzing the data for the between subjects evaluation of average module coupling, the p value was found to be 0. Since the significance of the within group comparisons is 0 (< 0.05), it indicates that at 95% confidence level there is a difference between the couplings of the TDM and the ERP. To find

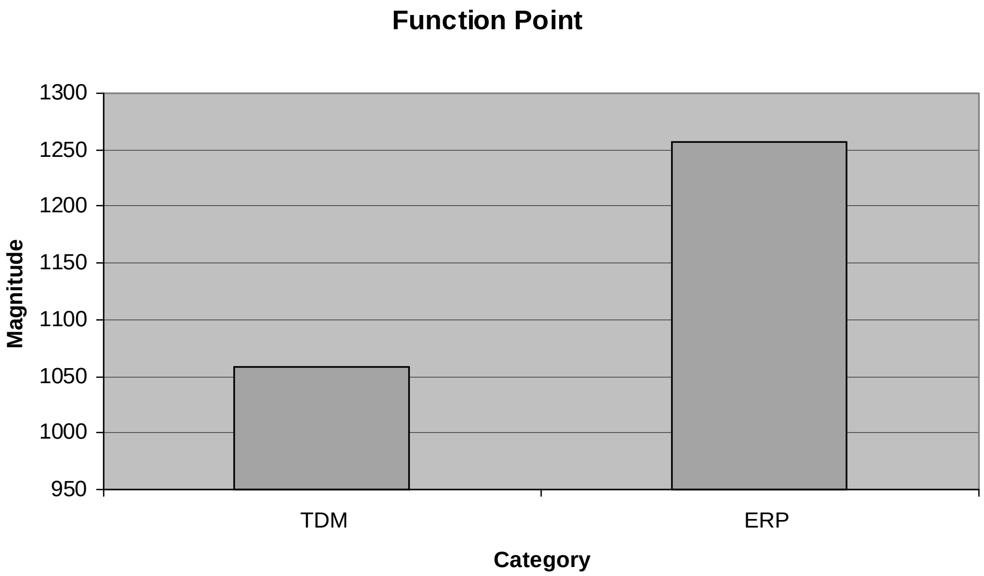

Fig. 4. Comparison of function point.

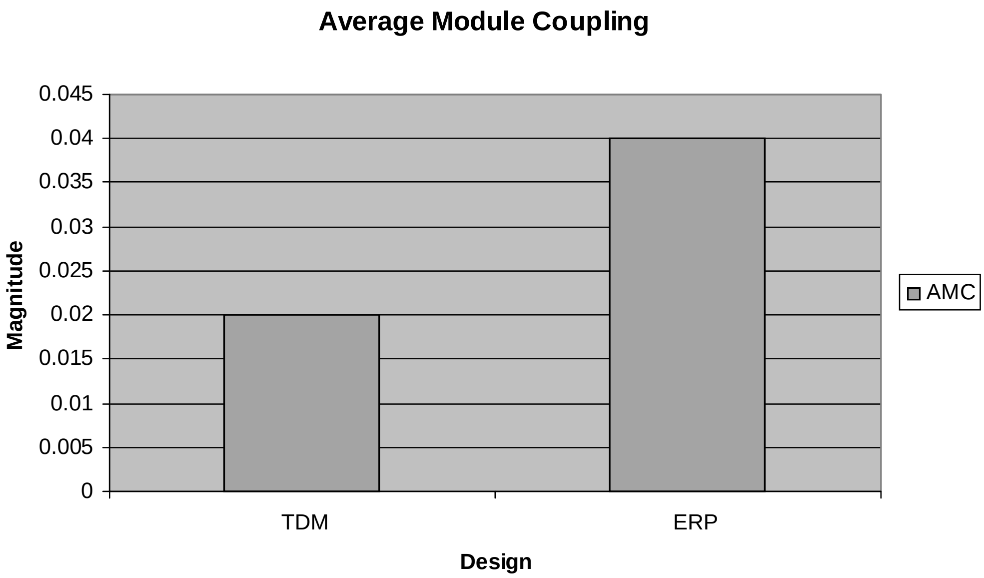

Fig. 5. Comparison of average module coupling.

which design has a lower coupling measure, the mean of the ten evaluators' readings was taken. The means are represented in Fig. 5.

Dhama (1995) describes a good system as one with a lower score for coupling metrics because this lowers the possibility that a change in one module will cascade to another module coupled to it. TDM has a lower value of coupling. A higher value indicates a loosely coupled system. Since TDM has an average module coupling score of 0.02 as compared to an average module coupling score of 0.04 for ERP. Therefore, for the test case TDM is a more loosely coupled system than the ERP.

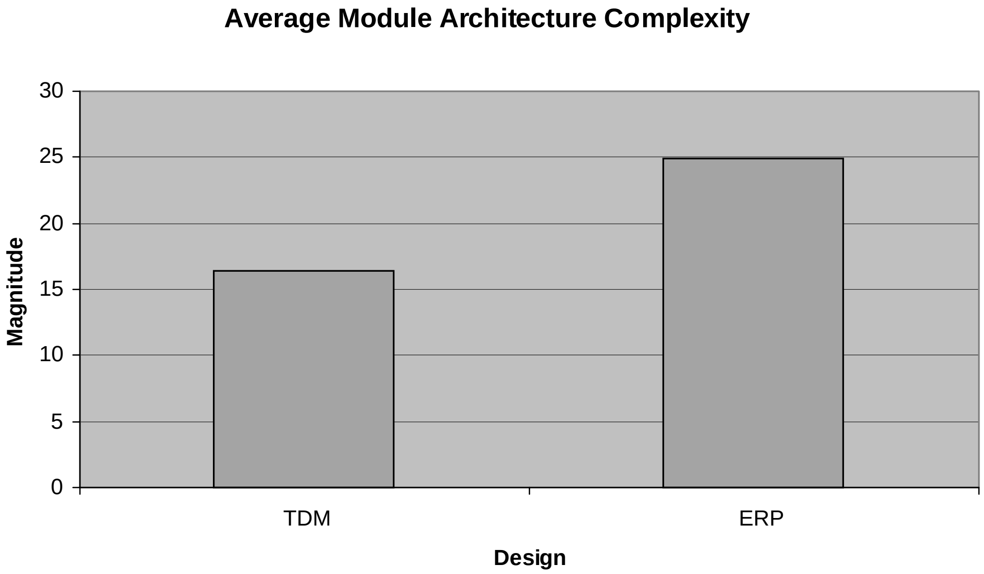

Fig. 6. Comparison of average module complexity.

After analyzing the data for the between subjects evaluation of average module architecture complexity, the p value was found to be 0. Since the significance of the within group comparisons is 0.019($<$ 0.05), it indicates that at 95% confidence level there is a difference between the couplings of the TDM and the ERP. To find which design has a lower complexity measure, the mean of the ten evaluators' readings was taken. The means are represented in Fig. 6.

The mean of the average module complexity for TDM is 16.38 compared to 24.90 for the ERP. A lower value indicates a less complex system. Therefore, it can be concluded that TDM is a less complex system for the test case than the ERP.

DISCUSSION

The literature review highlighted a number of issues associated with the ERP based systems: exorbitant cost, long implementation time, inability to change. The literature review also pointed towards the possible causes associated with these challenges. Some causes were complicated data structure, forced business process re-engineering to meet database requirements, a large monolithic nature, and tightly coupled modules.

Numerous case studies highlighting the challenges with the ERP system exist in the literature. This paper developed a framework for the evaluation of two or more enterprise-wide information systems at the conceptual design level using size, architectural complexity, and module coupling. Three metrics were selected to compare the systems based on these criteria.

Function point was the metric used to measure the sizes through two implementations. Module coupling was used to compare the systems for scalability and changeability. Module coupling assesses the cascading effect of changing an entry in one module. A system with a lower value of the average module coupling is a more flexible system. In this system, a change in one module will have a lower cascading effect on the other modules. The third metric, Module Architecture Complexity (MAC) measures the architectural complexity of each module and the contribution of this module to

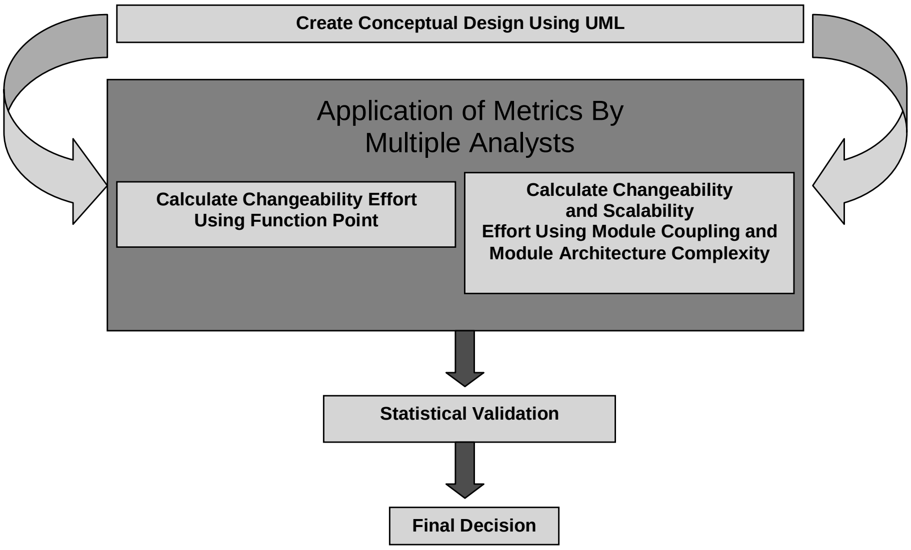

Fig. 7. Evaluation framework summary.

the architecture complexity of the system. This metric is an indication of the effort needed to modify a module. A module with a higher value of MAC is a more complex module and hence difficult to modify. Using these three metrics we can better estimate the effort and the associated cost with implementing as well as incorporating changes in enterprise-wide information systems. Kemerer (1993) identifies two shortcomings of the function points. One of the shortcomings is that the measurement of function points cannot be automated, i.e. the measures cannot be extracted automatically from design documents. The second issue is that function point analysis is prone to vary as a function of the individual differences of the people who perform them. Making the same argument for other two metrics, it was found necessary that multiple evaluators calculate the metrics and ensure reliability of resultant measures. Data collected from all evaluators was then subjected to further analysis using an analysis of variance technique called the two-factor-repeated measures design. One factor was the type of the system, and the other factor the evaluator. Since, in a repeated measures design the evaluators are treated like any main effect, the variability associated with the effect can be isolated, and, as a consequence, the error variance in this design is not inflated by variability due to individual differences (Myers, 1967). It should be pointed out that use of novice evaluators certainly resulted in an increase in variability, however due to the nature of these metrics removing this variation is impossible. It can certainly be reduced if we use IFPI certified evaluators, but owing to financial constraints it was not possible to hire certified evaluators for this research.

To test the framework two types of enterprise-wide information system methodologies were compared for an enterprise wide system. The first one was the event-driven standard ERP based methodology and the other was a document-based methodology referred to in this paper as the TDM. Mobile Homes Inc. was used as a test case to develop conceptual designs using the two methodologies. UML use case, activity, and sequence diagrams were constructed for each of the two methods. Mobile Homes Inc. has $100 million annual revenues and 200 employees. The breadth of the enterprise compared includes seven departments, which included sales, purchasing, engineering, production, quality control, service, and human resources. The overall operations are partitioned into fifteen processes. For evaluation purposes only seven out of fifteen processes were selected. The reason for selection of these processes was their utilizing multiple formats of documents e.g. database, spreadsheet, CAD drawings, and memos. The eight processes not selected were more administrative in nature and involved database application only. After analyzing the results of the experiment it was found that TDM was a smaller, less complex, and more decoupled compared to ERP. However, it should be kept in mind that this research only focuses on developing a framework for comparison and validating it through implementation in one case. Therefore, we are not claiming that TDM is better than ERP in all cases. In order to validate that claim several cases coming from a wide spectrum of industries will have to be analyzed.

This paper has developed a framework (Fig. 7) that will help evaluate the different systems over a much wider spectrum and not just concentrate on the project costs. Function point, module architectural complexity, and module coupling metrics were used to provide more insight into the issues that have been responsible for ERP implementation problems. Using repeated measures design, possibility of individual evaluator bias associated with these metrics is eliminated and the process has been made more objective. Performing this evaluation will definitely increase the time and effort needed at the conceptual design level but, on the other hand, it will provide the decision-makers with more information to base their decisions. This framework cannot be used in isolation. This framework has to be tied to efficiency improvement, process improvement and return on investment based metrics for a comprehensive evaluation. However, significant progress can be made by using these measure in lieu of simply considering these structure issues as intangibles, as is the practice currently.

REFERENCES

Agarwal, R., Raha, A.R., and Ghosh, B., (2000), "Our Experience and Learning in ERP Implementation", Software Engineering Notes 25(2), 31–34.

Albrecht, A.J., (1979), "Measuring Application Development Productivity", Proceedings IBM Application Development Symposium, Monterey, CA, October, 83–92.

APICS Dictionary, (1999), 9TH edition, American Production and Inventory Control Society, Falls Church, VA.

Arthur, L.J., (1985), "Measuring Programmer Productivity and Software Quality," Wiley-Interscience.

Bailey, J.W. and Basili, V.R., (1987), "A Meta-model for Software Development Resource Expenditure", 8th International Conference on Information Systems, 160–175.

Boehm, B.W., (1981), Software Engineering Economics, Englewood Cliffs, NJ: Prentice Hall.

Card, D.N. and Glass, R.L., (1990), "Measuring Software Design Quality", Prentice Hall, Upper Saddle River, NJ.

Curran, T. and Keller, G., (1998), "SAP R/3 Business Blueprint", Prentice Hall, Upper Saddle River, NJ.

Dhama, H., (1995), "Quantitative Models of Cohesion and Coupling in Software", Journal of Systems Software, 29, 65–74.

Grover, V. and Malhotra, M.K., (1997), "Business Process Re-engineering: A tutorial on the Concept, Evolution, Method, Technology and Application", Journal of Operations Management, 15, 193–213.

Hammer, M. and Champy, J., (1993), "Re-engineering the Corporation: A Manifesto for Business Evolution," Nicholas Brealcy, London.

Harrold, D., (2000), "Enterprise Integration Requires Understanding the Plant Floor", Control Engineering, February, 46–54.

International Function Point User Group (IFP), (1994), "Function Point Practices Manual", Release 4.

Jones, C., (1991), "Applied Software Measurement," McGraw Hill.

Jones, J. I., (1999), "The Document Methodology", Priority Process Associates, Rochester MI.

Keller, G. and Teufel, T., (1998), "SAP R/3 Process Oriented Implementation," Addison Wesley, Reading, MA.

Kemerer, C.F., (1993), "Reliability of Function Points Measurement," CACM, 36(2), 85–97.

Kulonda, D.J., (1999), "Investing in Technology Based Infrastructure: An Assessment of SAP", IE Solutions, May, Phoenix AZ.

Larsen, M.A., and Myers, M.D., (1999), "When Success Turns into Failure: a Package Driven Business Process Re-engineering Project in the Financial Service Industry", The Journal of Strategic Information Systems, 8(4), 395–417.

Moser, S., Henderson-Sellers, B. and Misic, V.B., (1999), "Cost Estimation Based on Business Models", The Journal of Systems and Software, 49, 33–42.

Mukhopadhyay, T., Vicinanza, S.S. and Prietula, M.J., (1992), "Examining the Feasibility of a Case-based Reasoning Model for Software Effort Estimation", MIS Quarterly, June, 155–171.

Myers, J.L., (1967), "Fundamentals of Experimental Design," Allyn and Bacon, New York, NY.

Ng, J.K.C., Ip, W.H. and Lee, T.C., (1999), "A Paradigm for ERP and BPR Integration", International Journal of Production Research, 37(9) 2093–2108.

Putman, L.H, (1978), "A General Empirical Solution to the Macro Software Sizing Problem", IEEE Transactions on Software Engineering, April, SE-4, 345–361.

Schow, G.D., (2001), "Conceptual Modeling of Applied Research Projects," Unpublished Ph.D. Dissertation, Department of Industrial Engineering and Management Systems, University of Central Florida, Orlando, Florida.

Shepperd, M.J., and Schofield, C., (1997), Estimating Software Project Effort Using Anologies", IEEE Transactions on Software Engineering, 21(2), 126–137.

Stensrud, E., (2001), "Alternative Approaches to Effort Prediction of ERP Projects", Information and Software Technology, 43(7), 413–423.

The Conference Board, (1998), "SAP Implementation: Strategies and Results," The Conference Board Inc, New York, NY.