ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
Indian Journal of Science and Technology, *Vol 8(2), 108–119, January 2015* DOI : 10.17485/ijst/2015/v8i2/57773

A New Architecture for Enterprise Resource Planning SystemsBased on a Combination of Event-based Software Architecture and Service-oriented Architecture

Esmail Amini[1] and Ali Ghaffari[2*]

[1]Department of Computer, Engineering, Science and Research Branch, Islamic Azad University, West Azerbaijan, Iran
[2]Department of Computer, Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran; a.ghaffari@iaut.ac.ir

Abstract

Enterprise Resource Planning (ERP) is a system through the application of which at organizational levels, companies can achieve their goal of homogenizing present functions in order to make a homogenous enterprise possible. Developers have been coping with a certain degree of complexity in the processes of designing and establishment of planning and organizational resource systems. Categorizing factors and relations with the purpose of transcending details and making holistic views possible is one of the methods for fighting the so-called complexity. This type of categorization is called software architecture. The purpose of this article is to provide the reader with a coherent and homogenous architecture for ERP systems via combining event-based and service-oriented architecture models. By means of introducing a modular framework for ERP systems, we divide systems into cores and independent modules, and then design a structure through combining event-based and service-oriented architectures for each section. Also in order to establish contact and information exchange both among sub-systems and between sub-systems and the core system, we have utilized an information exchange method based on XML. And simulating the so-called method in the Enterprise Architecture environment, we have improved the technological infrastructure and intractability in organizations to converge information technologies with business environments.

Keywords: Enterprise Resource Planning Systems, Event-Based Architecture, Service-Oriented Architecture, Software Architecture, Software Quality

1. Introduction

ERP is the theory through which companies can homogenize their sub-systems under the supervision and by the help of experts. Mega-organizations soon noticed that they need to adapt to changes in work conditions and technological developments and accordingly discovered the very importance of developing a homogenous piece of software via which it would become possible to fulfill needs at the various levels of management. The discovery has led to a tendency in companies for putting in use ERP software in order to respond to both internal and external demands. However, efforts as such would not bring forth proper improvements as to facilitate competition for large companies in the exponentially developing world today. But wherever it is possible to optimize knowledge transfer, particularly the organization or the employee will be responsible for the processes[1]. So the company rather than increasing investment on software packages, focus its investments on the use of integrated systems and involves the entire company in every day needs and parameters it needs, but do not answer them. Corporate systems are

*Author for correspondence

commercial software packages that integrate business processes and business information exchange across organizations enable, in fact, the data to be integrated by the software packages and by making their business processes to the full and to control them online[1,2]. Enterprise systems, including ERP software and related software packages are capable of advanced planning and scheduling, sales automation, customer relations management and product formation. One of the distinguishing features of enterprise systems integration is Integration without Borders i.e. data circulating in financial accounting system, human resource information, supply chain information, and customer information, Corporate computing systems that are functionally identical systems with different operations at different levels of providing organizations[3].

Today's organizations are complex creatures that describe the technical aspects of information systems as requiring the use of special architectural. Developments of software systems and software design in an attempt to win over the difficulties of designing and manufacturing ERP systems become larger and more complicated with time, in terms of the degree of variation in the number and complexity of factors and relationships in the systems or the faces that require grand approach to ERP systems have been. There are several ways to overcome the complexity of systems theory. Put one of these factors and relationships at various levels so that we can avoid the details of the macro we outlook[2,4]. That in some fields such as civil engineering, urban bridge maker, etc. It is a fusion of craft and become an architect. In fact, a solution to overcome the complexity of the system is the high level design.

Architecture has a Latin root, meaning a professor in the building. Architecture is a set of technical drawings where each map includes a description of certain aspects of the system. Architecture is a model used to describe various aspects of each module. The symptoms follow specific syntax rules and standards are literal. The models can be used to describe the components of a system and the relationships between them[5]. Architecture can be used to describe existing systems or systems that have been built or are to be used in the future. The architecture includes both structural and behavioral descriptions of a system. A set of technical drawings are applied to describe the architecture, which should be understood by designers and manufacturers, and fully in accordance with customer requirements. Architectural descriptions may be applicable. So part of the architectural design is called an action plan that includes a schedule of costs and resources required to organize the administrative architecture. Provide a technical description of a system architecture that reflects the structure of its components and the principles governing the relationship between the components of their design and evolution over time. The complexity of the specific needs such a large scale, long life and flexibility that can lead to necessary changes in the architecture of a system to be the architects of the design and help construction. Large prospective cohort and experienced architects are artists who seem to be able to also recognize that customer needs are expressed generally for a non technical scientific language and convert them to the standard fields of real creative thinking and provide their demands[7,8].

In this paper, our goal is to use the architectural style based on the event and the model of service-oriented architecture to introduce the structure and framework for ERP Systems which can improve the development of standardization and integration platforms. Additionally, increase the possibility of reuse and flexibility to maximize the intelligent services. Following methods and optimization procedures of our proposed architecture, Enterprise Architect modeling software architecture in the context of the models are done and the results were analyzed as the necessary changes were applied.

In part 2 of this article, we discuss the concept and objectives of ERP systems. Section (3) discusses the concept of software architecture and its necessity in software systems. Section (4) is an overview of related work in the field of ERP systems. In Section (5) the general structure of the proposed architecture for ERP systems and interior components and sub-systems and modules as well as how the interaction with the system based on the proposed architecture are introduced and examined. Section (6) concerns proposed architecture based on software quality attributes which are assessed, and finally in section (7) the result of the work presented in this paper and suggestions for further research are discussed.

2. Enterprise Resource Planning

In today's world the concept of mobility has changed, and managers must constantly search for ways to improve the section to be maintained in an organization. The use of information systems in an organization is to have all activities and tasks covered. Users can provide the

necessary information in a timely manner, which is a vital function in modern organizations. Without the increased capabilities of these systems to improve performance and achieve competitive advantage, it would be totally impossible decisions. The Enterprise Resource Planning system (ERP) is the latest management tool that enables the organizations to use available technology and unify multi-level information in order to provide users with information needed[3]. These systems could utilize the newest information technologies in the past decade as a result of the rapidly changing and evolution. Discrete and insular automation of work processes not only leads to uniformity, continuity and consistency in the flow of information as a valuable chain, it would also destroy the ability of the organizations to grow.

Discrete and insular automation business processes not only leads to uniform continuity and consistency of information flow in an organization's ability to grow, but it will also decrease the value chain and may even destroy it. Since the business environment is highly competitive in the current information age, information and knowledge is created, and systems/enterprises are forced to work together and cooperate as much as possible. To have a separate creation of such a mechanism is not possible without the help of an integrated system for all units of the organization in a financial office, warehouse or a production line, which is in control of their operations. Enterprise Resource Planning is the latest management tool that can utilize and organize information available in the organization, using information technology in all areas of the organization as a unified collection, making results available to users at various levels of the organization. ERP Enterprise organizations can use advanced applications such as Customer Relationship Management (CRM), Supply Chain Management (SCM) Electronic Procurement (EP) gain the so called technology[1,2,8].

3. Software Architecture

A general description of the technical architecture of software systems structure and behavior of a software system is a system which actually produces a structured software engineering where structural strength of the passage of time and the quality and size of these applications can be added. This is one of the major advantages in the use of the software architecture. Architecture is a set of technical drawings where each map includes a description of certain aspects of the system.

Architecture is a model used to describe various aspects of each model where the symptoms follow specific syntax rules and standards are literal. The models can be used to describe the components of a system and the relationships between them. The architecture includes both structural and behavioral descriptions of a system. A set of technical drawings applied to describe the architecture, should be understood by designers and manufacturers and be prepared strictly in accordance with customer requirements, architectural descriptions in order to be applicable[6,7].

Event-based architecture is to achieve scalability by isolating different parts of the computation in order to correct the style of the different processes that work together to form the message. This process also sends data directly to each part but does not control the messages sent to a specific subscriber. Each component of the data that is going to be multiplexed with the surroundings swallows the rest of the components in this class, and may be required to record some data. If you do this every time there was a data call, and receives data, component values record messages that want to manage the reception. It then gives a message to the object or the remaining subscribers and applicants[6,9].

Service Oriented Architecture is a strategic frame work of technology that allows all internal and external systems provide well-defined services, either giving or receiving. The general concept of service-oriented architecture focuses on defining professional infrastructure, rather than engaging the consumer with technical aspects of service-oriented architecture to implement a service call or service contract component that separates agreement. As a result of this separation an architecture where a connection between the consumer and the service modules do their work is very weak is easily reconfigured[7,10].

4. Previous Work

As the studies regarding the architecture of ERP systems to limited research in this area show, we stumble so with the knowledge of the capabilities and features of event-based architecture and service-oriented architecture model, we present the architecture of ERP systems we need. With a combination of event-based architecture of an architecture model, and Service-Oriented Architecture, we aim to produce an integrated and flexible enterprise resource planning system. In our studies we encountered articles on the same theme, they also have an end goal of our study,

thus referring to them and examine their shortcomings in this section can be useful. Agent-oriented architecture provides flexibility in ERP systems with the results of research that can help us achieve the goals in our article. Software architecture is an important part of an ERP software package that can help ensure the flexibility of the system and the process of reverse engineering. These parts of the environment in which the possibility of developing specific or ERP software package meet the specific needs of the users. on the other hand, one of the key distinctions compared to other systems integrated with ERP systems into their business processes based on best practices is of course, if necessary, is based on the specific needs of the buyer must become possible. One of the most important results-oriented architectural design factor was that one of the main limitations of current development environments lack flexibility and intelligence in supporting the creation and implementation of the reform process. These limitations, based on an agent-oriented architecture can be overcome[1,4].

The development process of ERP systems and administrative barriers at the macro level is another study that could help the process of this article. The objectives of this study was to examine the development process of ERP systems in organizations and institutions, and obstacles that lead to failure in implementing and running the system are noted. The method of the study is an analysis of territorial cohesion and spatial nature of the research field as its data is collected using questionnaires and interviews. This study tries to answer questions such as: How can ERP improve a company's business performance? What factors are often the causes of failure in ERP projects? The system-level software companies produce software packages related to how much progress has been made in order to provide a suitable architecture in which results can be used[2]. To identify and rank factors that influence the level of readiness for implementation of Enterprise Resource Planning software companies can integrate information that organizations need to remain competitive in all sectors of a common base, through the integration of processes within the ERP distribution software modules by the functional areas of the organization to resolve. Despite the advantages that ERP implementation achieved, a high degree based on the failure of these projects are implemented. Companies implementing ERP judge three quarters of these projects to fail. Ninety percent of the time and money spent on ERP projects on time and within budget take more than seventy percent of them while they do not achieve the anticipated benefits. Reasons for the high rate of failure are known to be alack of knowledge and indicators of success of the ERP companies and also implementation arrangements for the projects[3].

Implementation of Enterprise Resource Planning ERP designed one of the most important points that need to be considered. Most ministries and agencies and large corporations implement a proper integrated information system in their operational environment and frequently have large costs and result as reported in the study phase where the recognition system or implementation is inefficient. Problems in organizational processes are characteristics by many reasons employers and contractors are also put place, causing this failure to be evaluated. One of the solutions for the problems and the reasons behind the use of software tools that can be integrated into existing ERP products is based on the production processes standardized and related to problems in the implementation process in the system corresponding to large organizations. Their aim is to scan documents related to companies and experts in development and implementation of ERP and ERP product selection and implementation methods thereby. The results of this study indicate that the ERP products are too costly used in changing management culture in the organization while the network also has remained safe and sustainable. Then it was made possible using existing software ERP Enterprise resource planning in a good way so the benefits of increased productivity and efficiency of the organization are made about three years after the capital return[8,11].

Since the objective of this paper is to design an architecture combining Service-Oriented Architecture model and event-based architectural style and the architecture of ERP systems according to research conducted in the field of information systems and management applications in the field of software architecture, we can say that ERP design using hybrid architectures can provide better results.

5. The Overall Structure of the Proposed Architecture

In this paper, the proposed system architecture is divided into two distinct parts with different functions. The need to create two separate parts of the system comes from the extensible nature of these systems and the fact that these systems are part of the general system duties and

responsibilities are shared between all system components where the entire system perform administrative tasks. Such structure of the system could be more stable and variable components of the management systems the term refers to the part of the core system aimed. The core of the system reusability makes special-purpose subsystems as modules attached to the system in the form of an integrated system operating all necessary coordination between the inner core and the outside core kernel module management information[12,13]. In addition, other parts of the system that are the core of the system variables and the presence or absence of these sections can be used for various purposes in the work. These sectors are accomplished under the management and control system of the core elements that are included in this section. The term module refers, this article, to core management task in the system responsible which has its own components and the internal structure of which is considered while the core components and other structures can function. The core work within dependent modules to fulfill their duties under the core of what we consider. It detects the boundaries between the inner core and external components must be done in such a way that any dependence between core and external modules do cause desirable level of modularity and reusability of the systems well as the ability to achieve[11,14,15]. In Figure 1, based on the core system architecture and interior components and subsystems can be seen.

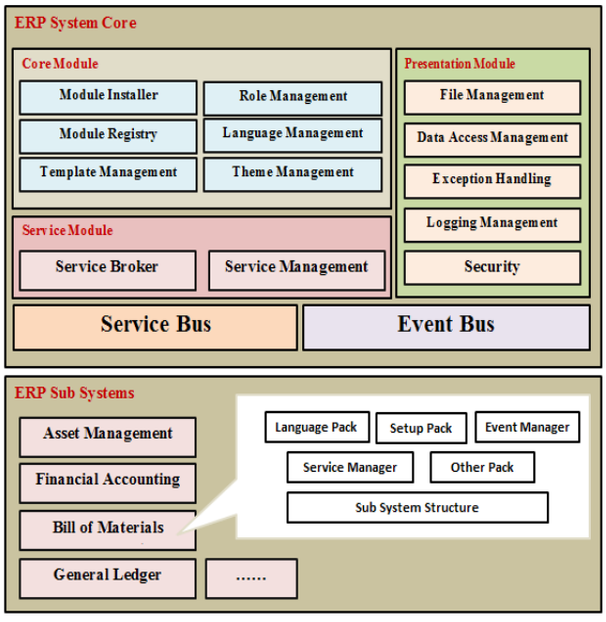

Figure 1. Structure of the Proposed Architectural Details.

As seen in Figure 1 you can see the structure of the proposed architecture is divided into two parts, the overall ERP System Core and ERP Sub Systems. ERP System Core has a central role in the system and reusable Web-based system also needs to bean infinite number so a variable was added to the system and ERP Sub-Systems in the form of independent modules and their goals to advance. ERP System Core sector has several internal components that manage, and service delivery modules and subsystems are required. Type and number of components required can change over time accordingly, but what is important to the ERP System Core architecture, is for our proposal to be based on the objectives defined in the ERP system and include elements of the EventBus and Service bus as a provider of management and implementation of the entire system. The EventBus connection between the inner core and core modules that interact with a sub system manages and controls the communication between modules and sub system hives, while Service Bus provides the internal and external services. The unit subsystems are ERP Sub Systems. In fact, a special-purpose subsystem is needed, it can be added as a module in our proposed architecture, so the system internal structure is completely independent four proposed architecture in the architecture of our proposed frame work in order to be able to play. The core can interact with an interface adapter when it is needed to be able to interact with the EventBus and Service Bus. In other words, our proposed architecture is the design of a subsystem with no limits on their internal structure, but if it does not play in an integrated system within the core, it must have the ability to communicate and interact with the core of the system, and to do this will require a relationship. When a subsystem is added the module system can provide all the facilities and services so the core modules will with respect to the privacy of those which use the service to other modules[14,16].

5.1 The Core of the Proposed Architecture

The core of our proposed solution should have separate sections for a series of independent modules and subsystems and their management, because some of these features function properly as control modules for central as well as directors of the necessary modules. Why do we need the not maintain information about the state of the core modules in the system, when we define it as the key to achieving what we need in the proposed approach used in our model? The proposed architecture of the module

and the configuration modules responsible for information and Module Installer is the Module Registry shown as the core. Moreover, in this architecture, we have all the required service modules in Figure 1 as part of the possibility to access resources in the form of a separate department with in the core modules and subsystems which have been isolated. The whole system can be integrated with away to access resources[11,15]. Therefore Presentation Module as a tool was used in order to access a variety of resources and has been defined within the kernel, so that all departments and independent modules of the system can be accessed, and powerful interface could access and manage their own resources to their own benefit. Therefore we have also defined a set so Presentation Module, as it can be seen in [11,15], in order for the system with a unified approach to access resources. A variety of resources have been defined within the kernel, so that all departments and independent modules of the system can be accessed, and powerful interface could have access to and manage their own resources to benefit. We also included a set of modules that provide services to the runtime modules defined within the Presentation Modules and subsystems, so that they can handle management of the service and their security codes and what they need in their operations in the proposed architecture using services it provides. The system no longer acts independently to resolve these challenges but takes advantage of a variety of mechanisms as it may be the core system that serves with high performance and provides them with what they need. Each section contains the core services that are functionally related[17]. Core Module Registry has a number of components including the data storage modules and Role Management where modules are responsible for modules configuration tasks. Modules add ability to manage multi-language and module system Language Management Users and user access control, and module Presentation add personalization functionality to the system as also included in this section. Template Management[16,17] which has several components and parts is to manage and control access to various resources including module service. This section contains modules that can be called Service Module Noted which other parts of the kernel, including all parts and different parts of the work within the proposed architecture can access a broker[16,18]. Moreover, noted that all parts and different parts of the frame work proposed architecture can operate have access to and compliance with standards related to your service in and out of the system[16,18]. Besides such important sectors as modules that can be provided by the core of the proposed architecture to serve Internal and external kernel modules a turn time and high integrity management system errors. The Security module system ensures data security and privacy, note that all these modules are based on Figure 1 the insensitive section entitled Presentation Module. The dynamics and development capabilities of the objectives of the proposed architecture and various parts of the system modules which can be detected and embedded in to the core of the system and therefore there are a number of types and modules in the proposed architecture as the basis of the core modules we offer to be fixed, yet it is important to know that they can easily be added or removed[16,17].

5.2 The Proposed Architecture for Event Processing Solutions and Services

5.2.1 Events Processed

According to Figure 1, parts listed inside and outside the operating system kernel achieve their goals and services interdependently so the relationship between them must be established in the proposed architecture as such. How to communicate between different parts of the architecture should be defined, so that we can maintain the system's modularity and mobility and expandability altogether. Therefore, according to the event-based architecture inside the nucleus of an EventBus, the system was used to manage and control the communication between the modules. The EventBus-based communication mechanism can send message stopper form various messages received and recognized based on standard structure and process play back systems[19]. All modules and subsystems independent of the core and the inner core of the EventBus have access and are able to communicate within its request in order to send a message to the structure. Figure 2 demonstrates the overall structure and mode of operation of the system shows the EventBus.

According to Figure 2, EventBus exists as a core system which has two separate sections of the Event Process or and the Input Buffer is designed to store requests as logged to the Event Bus when there is a plurality of received messages. Order entry and processing system is able to reach its destination. Event Processor Event Processing Engine system is the same as the input queue of incoming messages to the EventBus. Input Buffer is located in the system as shown in Figure 2. Event Processor Core Module Registry is used to identify modules in the system

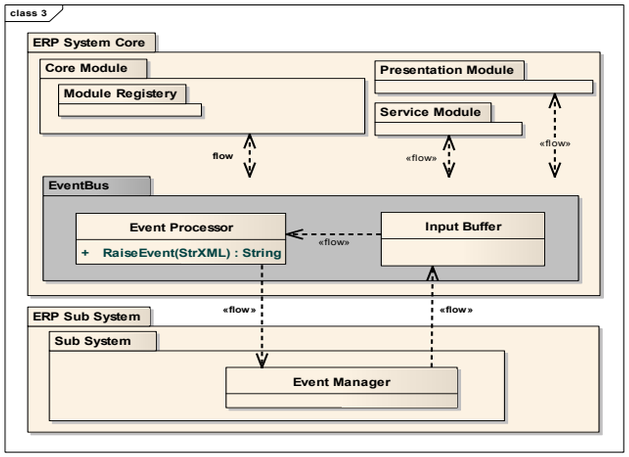

Figure 2. EventBus and how it Functions in the Proposed Architecture.

to take advantage, so when an independent subsystem in the kernel sends a request message to the Input Buffer as it is then processed by the Event Processing Engine or the Event Processor. The source and target modules extracted from the incoming message are then used by the module Registry to identify and verify the identity of the court room to broadcast the message to the destination, or the destination of the message. The Event Processing Engine can be input to the EventBus messages without processing system broadcast to all modules and each of its modules independently. To identify and receive messages and take away the message and causes performance depends on the design of policies. The Event Processor should be considered, depending on the operating environment of the system.

5.2.2 Processing Service

Given the form Figure 1 of the proposed architecture, we have emphasized on Architectural Support for external services. The following ERP systems achieve their goals related to services and external services and infrastructure architecture must also be provided to communicate with the outside world. Hence, according to service-oriented architectural model, we used a Service Bus Inside the core of the management and controlled subsystem communication with external services available used. The Service Bus message-based communication mechanism of action cannot receive messages according to a standard structure and are known using Service Management process are an identification services for a service call using Service Broker module to act. All subsystems independent of the core and the inner core of the Service Bus can access and interact with it to call service[19,20]. Figure 3 shows the Overall structure of Service Bus and mode of operation of the system.

According to Figure 3, Service Bus in the core system has two distinct parts namely the Service Process or and Input Buffer, Service Processor Processing Engine Service Bus service request system that requests input into the input queue and Input Buffer in System. As shown in Figure 3 you can see that in the Service Module Service Processor from Service Management module is used to identify services available and accessible system uses. When a sub system in the form of a request for a service call sends a message to the core, this message into the Input Buffer is then processed by the Service Processor. Network Processing Engine or Service Processor from Service Management has requested the information requested service and the identification and authentication service action requested by the Service Broker provides a service call.

5.3 Interaction between Modules in the Proposed Architecture

After defining the structure and EventBus Service Bus and how they turn towards applications for the subsystems and modules to system modules structure is Subsystems and how they should act and how to communicate with other modules and services to and what steps to take steps towards establishing a healthy relationship to the success-

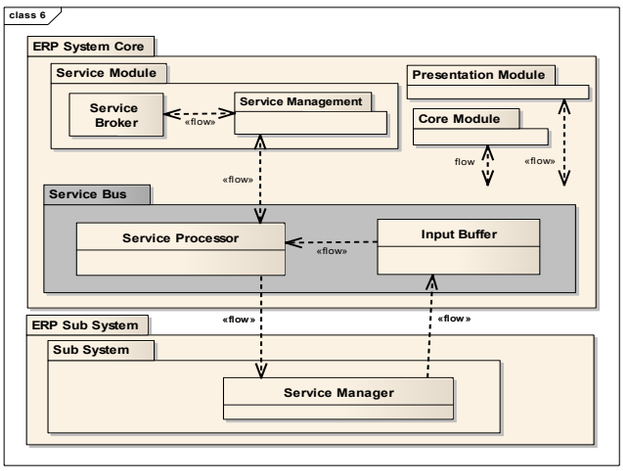

Figure 3. Service bus and how it functions in the proposed architecture.

ful. Considering that the existing module software systems based on different architectures and different structures of organization designed and therefore changes in their internal structure and build a standard structure for all modules operating costs as well as some restrictions on the system will be. In this paper, our proposed architecture is independent of the internal structure of modules and their internal structure changes do not So that the interior of your module to any act that would. The proposed architecture is important, what's important and external structure of subsystems and how they interact with other modules. In this regard, the best way to achieve this goal is to define a standard interface for the system. So that the task interface and the ability to communicate between the internal and external structure of the module is Send and receive messages about responsibility and take charge of their own module[19–21].The main task of the interface messages and requests can be identified within the structure of the system and sent to the EventBus Service Bus and the core, and if a message from the EventBus Service Bus modules were sent or received, it can detect and will be processed.

5.3.1 *Defining a Standard Interface for Modules and Subsystems*

Interface module or the Event Manager and Service Manager subsystems are within the standard structure for all modules are the results which can be designed by the interface module as soon as the system core. When adding modules to the system, it dynamically generates the interface module and the module to allocate new entrants, and then, if necessary, communicate with the outside of the module itself and also use core system services, in order to produce a proprietary interface. In general, our proposed architecture is a module that can operate in the interface according to what your core standards provides[22]. But what is important here is the internal structure of the module interface and how it functions in different situations. To achieve an efficient and flexible structure, designing module interfaces should be the first task for the interface modules and the system should be liable to be given. Tasks module are shaped according to the expectations of the system modules and each module can be changed in terms of expectation and therefore the dynamic development of the important factors should be considered in the design module interface. But the main task of the module is the interface module is to send and receive messages modules making up the core of the system able to communicate with other modules and external services. If such detailed scenario establishes connections between modules and if the module generates messages encoding and decoding from other tasks the interface module responsible for considering these points. Figure 4 is a proposed model for the internal structure of the module interface. The expandability and flexibility of qualitative characteristics that have been observed this structure are, however, are not constant with respect to different operating environments as they require that our proposed architecture needs be developed[19,22].

We proposed architecture of the module interface Event Manager and Service Manager are the same within the internal structure of the system, and they almost have to be a model. In Figure 4 we provide the most basic structure for the interface module.

Figure 4 shows an independent unit within a module interface module, as the structure of the system is active. Interface module includes both input and output buffers. The purpose of the Design Event Manager and Service Manager is for the buffer power and performance to meet in the Module Processor interface when traffic is high in the system. To preserve the module it needs to be able to process and respond to all messages and requests. The output buffer contains the messages that the core module or modules and subsystems are no longer active. Input buffer contains the messages received from the system kernel or other modules as the response is returned from a service. Module Processor accomplished the task of

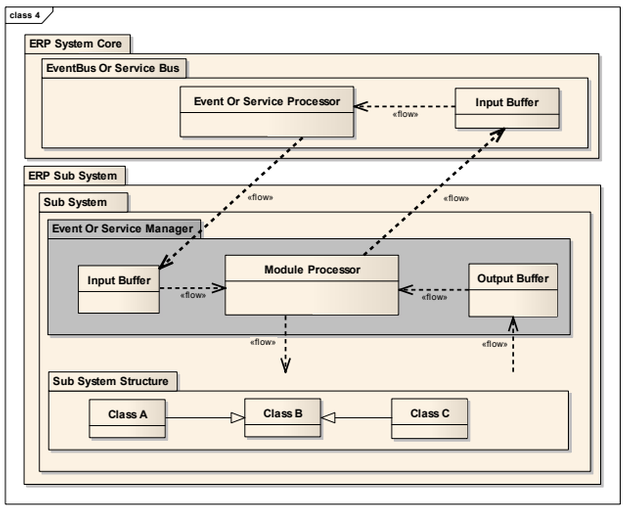

Figure 4. Internal Structure and External Interface Module.

broadcasting messages in the input and output buffers. Module Processor activates as soon as a message reaches, and the input and output buffers reacts after encryption or decryption of messages are sent to their destination.

5.3.2 Structure of Sent and Received Messages

In this paper we combine the event-based architecture and service-oriented architecture for distributed messaging systems used. In this article we concluded that our method for communication between modules and various parts of the system is based on message exchange. Here there is a question that must be answered and a message should be sent back containing information between subsystems and services. How could the structure be designed and integrated in the entire system? And how all sub-systems known are to be processed? What is certain is that the designer can in any manner that will enhance the system's performance, define the Message Structure. Concerning the structure of the message, a point that must be observed is that the message must be the original data which may be best protected and completed as the destination must be legible. Message structure must be such that the origin and destination of the message is quite clear; therefore, in order to have a well-defined structure of message we have used XML to design. The purpose of designing the structure of an XML-based message which can be defined for specific parts of a message tag in order to maximize the readability of the message, and the message processing speed of the various modules should ideally increase. The message structure is developed so the achievement of the goals of the system is easy to implement[19–22].

Figure 5 is an example of the messages you see and the returning message which can be recognized as a standard message in the system and all modules as long as sections of the structure of the form Figure 5 to convey information or to apply to their intended use.

As shown in Figure 5 Event in sensitive tag names contain basic information related to the message header,

```
<Event EventType="Sent | Reply" EventID="ID" SentEventID=""
SenderName="senderName" SenderID="senderID" ReceiverName="Core | a Module | *"
ReceiverID="ModuleID | *" DateTime="">
        <InputFields>
                <Field Name="fieldName" Value="value">
                ...
        </InputFields>
        <OutputFields>
                <Field Name="fieldName" Value="value">
                ...
        </OutputFields>
</Event>
```

Figure 5. The Structure of Messages Exchanged between the Modules.

and the message offers Information Message Input Fields as Output Fields located within the tags are transferred. Profile of each of the tags used in the structure shown in Figure 5 and its application in Table 1 is given.

5.4 Case Study: Information Recovery from a Database

Given that most ERP systems, databases, integrity and security of data, their main source of data storage is therefore said to be one of the most commonly used modules in the proposed architecture data Access Management as a module specifically to store and retrieve data in database

Table 1. Tags are represented in the Structure of the Message in the Figure 5

Description	Tag
Specifies whether the message is sent in response to a message is issued.	Event Type
Code specifies the event or message is unique and can be controlled by the system kernel.	Event ID
If the message is the return type of this attribute specifies that the message is issued in response to either.	Sent Event ID
Specifies the name of the module or unit is sending the message. System kernel or a module that can be	Sender Name
Module or unit of code specifies the message sender can be found in the module information registry module within the core activity is achieved.	Sender ID
This attribute specifies the name of the recipient. That could be one of the core units or a system or Several independent modules	Receiver Name
Code specifies the recipient and can be found in the module information registry module within the core activity is achieved.	Receiver ID
Defines the date and time the message through.	Date Time
This attribute contains information that is sent to the destination port as input or contains information from the source to the destination application Information on this option can be set identifier/value is.	Input Fields
This attribute contains information that identifies the message destination, the message processing a response message back to the sender that it Information on this option can be set identifier/value is	Output Fields

management systems. Now consider the online library management system in the form of a sub-system added to the ERP system. If some of your access sub-system data want to be moved or restored based on their request, they have to be transformed in the form of a string containing a command T-SQL interface. Function module interface communicate between the modules where the core of the system is responsible. There is a single module interface called the Application Module Processor sub-system in a standard message format based on the XML setting which adds meta-data including the source and destination of the message to be applied to the EventBus as the core sample system sends messages to be exchanged between the interface and the EventBus in the Figure of online library management system (6). After entering the EventBus and waiting in line at the entrance to the Event Process or available on the EventBus, Event Processor processes meta-data in the message to spread it to the favored destination. Event Processor Module Registry is to identify and access the modules, the system uses, after identifying the target module or modules, where the Data Access Management Message is sent, as it is simultaneously sent to the desired purposes[20–22].

Figure 6 views messages produced by the online library management module interface as a set of meta-data. The information about the event provides the message source and destination. In formation messages are transmitted in the form of tags, Input field in T-SQL are the basic information which consist of a command as data retrieval requests book codes greater than 20.

The request message via the EventBus Data Access Management module is delivered. Data Access Management module has requested the message processing and analysis. Also a detailed examination of the application and the desired access rights to the resource is allowed, as application implementation and results provide the same message format in Figure 7 sets the request

```
<Event  EventType="Sent" EventID="176" SentEventID=""
        SenderName="Library Managment Module" SenderID="432"
        ReceiverName="Data Access Module"
        ReceiverID="56" DateTime="23:13:35">
        <InputFields>
                <Field  Name="SelectCommand"
                        Value="Select * from Book Where BookID > 20">
        </InputFields>
        <OutputFields>
                <!-- Return Information -->
        </OutputFields>
</Event>
```

Figure 6. The Message Generated By the Interface Management Module Library.

```
<Event   EventType="Reply" EventID="210" SentEventID="176"
        SenderName="Data Access Module" SenderID="56"
        ReceiverName="Library Managment Module"
        ReceiverID="432" DateTime="23:13:40">
        <InputFields>
                <Field  Name="SelectCommand"
                        Value="Select * from Book Where BookID > 20">
        </InputFields>
        <OutputFields>
                <Object>
                        <Field Name="BookID" Value="22">
                        <Field Name="Title" Value="C++">
                        <Field Name="Pages" Value="441">
                        <Field Name="Author" Value="D&D">
                </Object>
                <Object>
                        <Field Name="BookID" Value="31">
                        <Field Name="Title" Value="VB">
                        <Field Name="Pages" Value="372">
                        <Field Name="Author" Value="John Smith">
                </Object>
        </OutputFields>
</Event>
```

Figure 7. Returned Message from the Data Access Module in Response to Online Library.

as a response to an online library management module. Of course, the answer must be sent to the EventBus-core system and then be sent to the online library management system. Typical message structure returned from the Data Access Management module is shown in the Figure 7.

As shown in Figure 7, Data Access Management module responds to a request online library management system module providing a list of books. The code in the format tag 20 is greater than Output Fields as it returns this module. You can also receive messages from the meta-data as well as the type of the returned message indicates that the message was sent in response to each request.

6. Evaluation of the Proposed Architecture

According to the proposed architecture and event-based architecture combining Service-Oriented Architecture, Tips and features can be noted that the proposed architecture in terms of software quality attributes, apart from other architectures. Event-based architecture intends to isolate different parts of computation. The proposed architecture provides a high level of accountability reformation in modules to send data to each other and improve indirect control. Event-based architecture aims to detect other modules as independent of other modules. Under the proposed architecture, parallel modules can work together only when proper amounts of data are exchanged. Our main goal is to utilize event-based architecture. The proposed architecture is a feature that can serve data to a client or customer while each service provider acts simultaneously and at the same time provides the client with control over the process, if the

a synchronous operation is only given to the customer. The possibility of using event-based architecture can be elevated via increasing the scalability of the system. Apart from the event-based architecture and service-oriented architecture, the proposed architecture improves the characteristics of any other models of software architecture that could not be achieved. Among the benefits of the proposed architecture using service-oriented architecture model we can find the flexibility to convert to a service-oriented process model to a rapid creation of new processes and compositions of software services in order to compete in any changing business environment, taking into consideration provision and use of the best options among the array of available services and a more active involvement of IT in organizations.

Using the proposed architecture, we can solve the problem of integrating complex systems and meet the professional needs. It would also be possible to divide any project into smaller components which can be performed independently for each sub-project, as we can also easily achieve and improve monitoring and control while calculating independently. Using the proposed architecture for ERP systems, systems simply require users to be satisfied. Problem solving and improving the integrity of data transfer between systems would help establishing information systems rather than replacing the whole system. Systems as such will be more popular among the stakeholders because they reduce costs and time to be reconfigured.

7. Conclusion and Further Work

Through a presentation of the internal architecture and structure of the so-called modules substantiated the fact that our suggested architecture not only introduces a new framework based on a combination of event-based architecture and the service-oriented architectural module which develops ERP systems but also owns numerous software qualities namely reusability, expandability, and trustworthiness. Our model makes it possible for the modular systems to be developed by various groups working independently. Our research has created a framework in the study of ERP systems which allows us develop multiple modules simultaneously. It is possible to develop a new architecture and framework in software development where various modules can work independently in the same framework and interact, But would only be possible if the development of the above-mentioned model is done according to the principles and needs in ERP systems in the developing industry of software and qualitative principles of software. And in future it would be possible to develop our model through the use of technologies independent of platform as a part of a modular architecture, in order to elevate levels of availability and security of information in ERP systems and improve availability of information services.

8. References

1. Al-Mashari M, Zairi M. Supply-chain re-engineering using enterprise-resource planning (ERP) systems: an analysis of a SAP R/3 implementation case. Int J Phys Distrib Logist Manag. 2000; 30(3/4):296–313.
2. Al-Mudimigh A, Zairi M, Al-Mashari M. ERP software implementation: an integrative framework. Eur J Inform Syst. 2001; 10(4):216–26.
3. Boykin RF. Enterprise resource-planning software: a solution to the return material authorization problem. Comput Ind. 2001; 45:99–109.
4. Irmert F, Daum M, Meyer-Wegener K. A New Approach to Modular Database Systems. SETMDM '08 Proceedings of the 2008 EDBT workshop on Software engineering for tailor-made data management; New York, USA. ACM; 2008.
5. Hohpe G, Woolf B. Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. The Addison-Wesley Signature Series. Addison-Wesley; 2004.
6. Rozanski N, Woods E. Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives. Addison-Wesley Professional; 2005.
7. The Architecture Working Group of the Software Engineering Committee. Recommended Practice for Architectural Description of Software Intensive Systems. IEEE Standards Department; 2000. Report No.: IEEE P1471–2000.
8. Bernroider E, Koch S. ERP selection process in mid-size and large organizations. Bus Process Manag J. 2001; 7(3):251–7.
9. Gharehchopogh FS, Amini E, Zebardast B. A Three-Layer Architecture based Approach for Data Access Layer in the Information Systems Production. IJARCET. 2013; 2(2):761–6.
10. Gharehchopogh FS, Amini E, Zebardast B. Aspect-Oriented Software Development based Solution for Intervention Concerns Problems: Case Study. Int J Comput Appl. 2013; 63(4).
11. Bao Y, Sun X, Trivedi KS. A Workload-Based Analysis of Software Aging, and Rejuvenation. IEEE Trans Reliab. 2005; 54:54–7.

12. Bourret R, Coates AB, Harvey B, Holman GK, Kay M et al. Advanced XML Applications from the Experts at The XML Guild. Thomson Learning Inc; 2007.
13. Masdari M, Amini E. A New Approach to Organized Data Access in Software Product Line. International Journal of Applied Information Systems. 2013; 5(9):23–28.
14. Clements P, Bachmann F, Bass L, Darlan D, Ivers J, Little R et al. Documenting Software Architectures, Views and Beyond. The SEI Series in Software Engineering. Addison-Wesley; 2003.
15. Frankel DS. Model Driven Architecture: Applying MDA to Enterprise Computing. Wiley Publishing; 2003.
16. Zachman JA. A Framework for Information Systems Architecture. IBM Systems Journal. 1987; 26(3):276–92.
17. Garlan D, Shaw M, Okasaki C, Scott C, Swonger R. Experience with a course on architectures for software systems. Proceedings of the Sixth SEI Conference on Software Engineering Education; Springer-Verlag; 1992 Oct.
18. Bass L, Clements P, Kazman R. Software Architecture in Practice. 2nd ed. Addison Wesley; 2003 Apr 11.
19. Chen D, Dharmaraja S, Chen D, Li L, Trivedi KS, Some RR, Nikora AP. Reliability and availability analysis for the JPL Remote Exploration and Experimentation System. Proceedings of DSN; 2002.
20. McGovern J, Ambler SW, Stevens ME, Linn J, Sharan V, Jo EK. A Practical Guide to Enterprise Architecture. Prentice Hall PTR; 2003.
21. Erl T. Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall PTR; 2005.
22. Gharehchopogh FS, Zebardast B, Amini E. Analysis and Design by Agent based MaSE Methodology: A Case Study. Int J Comput Appl. 2013; 63(4):10–5.