Abstract: Business process management systems play a central role in supporting the business operations of medium and large organizations. Because of this the security characteristics of these systems are becoming very important. The present paper describes the BWS architecture used to implement the open process aware information system DocuMentor. Using the proposed platform, the article identifies the security characteristics of such systems, shows the correlation between these characteristics and the security features implemented by the platform and presents examples of how the security of such systems can be enhanced using the extension mechanism.

Key-Words: BWL, Workflow, OBES, Open Application, Security.

1. Introduction

All modern businesses depend on complex business processes in order to conduct their daily activities. These processes involve internal or external information systems, documents and people. Traditional task based systems support the user in performing specific tasks, but they fail to integrate all the aspects involved in a typical business process. In order to manage holistically the business processes running inside an organization a new class of information systems named Process Aware Information Systems (PAIS) are used. According to [1], process aware information systems are information systems that manage and execute operational processes involving people, applications, and information sources on the basis of process models.

In order to fully reap the benefits from a PAIS, the system must be designed and implemented as an open system - OBES. In order to be considered an OBES a platform must include features that allow the following:

- creation of extension modules that add new UI and processing features;
- generic workflow management capabilities: extensible formal workflow language and associated runtime engine;
- bidirectional integration with other systems at the UI, service and data level using well defined integration points;
- automation of repetitive tasks using scripts or other automation features.

This paper presents a brief overview of the proposed platform, identifies the security characteristics specific to this type of systems, and presents the built-in security features, how these features influence the characteristics and how the platform extension mechanism can be leveraged to include custom security features.

2. BWS Architecture Overview

Modern economic information systems - EIS are large-scale distributed systems aimed at automating business processes within organizations. The features that characterize these systems are the size, purpose and importance to the organization. Large size is a defining feature of modern EIS. This is apparent both in terms of volume of data processed and the number of concurrent users and the size and complexity of component modules. The purpose of economic systems is to meet specific
business needs. Therefore, economic systems encode processes, rules and manipulate data entities within the organization. The ultimate objective of these systems is to increase profits by streamlining the organizations business processes. The **high importance** of EIS comes from the fact that business processes supported by these systems have become critical for the success of the organization in the context of increased computerization. Therefore systems must be robust enough to allow continuous operation and be able to handle the load when the volume of processed data increases.

All these characteristics make the development of economic systems very complex and risky task. The shortcomings of traditional EIS are:
- high cost and rate of failure of EIS development projects
- unnecessary complexity from the use of inappropriate tools and techniques;
- failure to implement a continuous updating strategy.

To eliminate these deficiencies of traditional EIS this paper proposes a new type of system called open business enterprise system - OBES. An OBES provides holistic business process management tools, has well defined interfaces required for two-way integration with existing systems and act as a development platform for extension components.

Because economic activity in organizations is not 100% automated, the implementation of business processes requires the management of a combination of manual activities and automated activities that use functionality provided by the existing systems. This situation creates two categories of problems in classical systems:
- enterprise system integration: this is necessary to avoid the human effort required to access and synchronize information from disconnected systems;
- holistic business processes management within the organization: involves the allocation and tracking of tasks performed by human actors or the systems to meet the objectives. This requires the presence of a mechanism for defining, implementing and monitoring processes. This mechanism must allow formal process definition, easy process update and automatic execution.

An OBES provides solutions to both problems.

An open business enterprise system,

\[
\text{OBES} = (\text{OSC, OSP, OSS})
\]

is an integrated set of subsystems that support operations, management and decision making, where:
- OSC – Open System Client is the interface that allows the final user to interact with the data and business processes managed by the system; it can be extended using modules that implement specific business operations or provide integration with legacy systems;
- OSP – Open System Processing is the service that implements system operations, manages process instances, exposes its functionality to the external systems using a well defined interface, and supports extension modules for integration with other systems or implementation of specialized processing tasks;
- OSS – Open System Storage stores the data manipulated by the business processes in a user defined structure.

Open business enterprise systems have the following characteristics:
- adaptability: allows the implementation of new presentation and processing features without requiring modification of the core system architecture or code;
- integrability: allows integration with other systems at the user interface level by creating plug-ins; the
service provides a mechanism for using the functionality implemented by other systems and expose its functionality through services; the extensible storage mechanism allows the integration of data;

- process management: offers formal mechanisms for describing business processes and implements a general mechanism for managing the execution of business processes, regardless of the business domain;

- automation: offers interfaces for constructing the automated data processing tools and allows automating repetitive tasks using scripting languages or other similar mechanisms.

An OBES architecture is a description of the structure of a system which comprises the components and the relationships between them. The proposed OBES architecture is called Business Workflow System - BWS. The objective of the BWS architecture is to describe the components necessary for a system designed to fully automate business processes inside a modern enterprise.

Figure 1 shows the main components of the BWS architecture and the dependency relationship between them. The three major components are the one specified in the OBES definition: OSS, OSP and OSC.

The BWS implementation of the OSS is called Flexible Data Storage - FDS. The FDS allows the user to define the entities necessary to support business processes. Figure 2 shows the main concepts used to define the structure of information inside FDS.

The main concept is the Data Collection Definition – DC. A data collection DC = (\{IF_{1-n}\}, \{DT_{1-m}\}, \{SC_{1-o}\}) describes the structure of the contained data items using the following structures:
• \{IF_{1-n}\}: a collection of n index field definitions; each index field definition specifies an attribute (name, data type, if required) that can be associated with each data item for retrieval purposes;

• \{DT_{1-m}\}: a collection of m document types (id, name) that defines the types of data items that can be added to the collection; each document type definition can contain an associated BWL program that is used to create process instances when a new data item is added to the system;

• \{SC_{1-o}\}: a collection of o status codes (id, name) that are used to specify the status of the data items contained inside the collection.

Data collections are organized for administrative purposes in collection groups. The process data is stored in data items or documents. Besides the actual data, a document contains metadata values as defined by the DC. All data objects contain an XML formatted store for extended properties associated with that particular object.

Authentication and authorization data is also stored inside the FDC. The authentication data consists of user descriptors and credentials. The authorization data consists of:

• Permission objects: triplets of the form (user id, target object id, operation type) that specify what are the allowed operations for a particular user;

• Rules objects: specify additional authorization rules based on the metadata defined by the DC.

The FDS supports data auditing. Each operation belongs to a session that records the user, location and time period. Operations performed against the data objects are recorded into the audit log. Each operation description is associated with a session and contains all a copy of all modified values. The audit log has two purposes. It offers the necessary information for security audits and allows the system to recreate any previous system state.

The OSP layer of the BWS architecture is responsible for collecting and processing of all the operations requested by the BWS client or by external systems. The components are:

• Data Access API: implements the interface to the FDS and offers services like data mapping, error handling and transaction support;

• WS Endpoint: is BWS service interface to the external world; it collects and de-code all operation requests from the outside and formats the operation results;

• Operation Implementation: is responsible for implementing each operation supported by the system; for each request it performs the authentication and authorization, establishes a transaction context, invokes the Data Access API to perform the actual processing and notifies the operation to the BWL runtime;

• BWL Runtime: manages the BWL workflow instances, notifies them when relevant event occur and offers them access to the FDS using the Data Access API;

• Plugin Manager: is responsible for loading and managing the platform extension components; the processing plugins that run inside the BWS service are responsible for customized data processing of the data items and access to the external systems for system integration purposes.

The Business Workflow Language – BWS implemented inside the service is described in detail in [3] and [4].

The OSC layer of BWS offers end users access to the systems. It is built around the Extensible Application Dispatcher mode. The proposed model facilitates the creation of a completely extensible client with full automation support. The BWS Client Model consists of:

• Data structures that describe the UI components and process data;

• Data modification event publishers;

• Operation event publishers.
The BWL language and BWS architecture were implemented in the DocuMentor platform. DocuMentor is an OBES that performs three major roles inside the enterprise:

- data management: allows the user to define the structure, retrieval methods and access rules for the data collections used to support the business processes;
- business process management: the platform offers a full BWL implementation that allows the user to manage the business processes inside the organization;
- development and integration platform: using the DocuMentor eXtension – DMX the client can extend the UI and processing components of the platform in order to create specialized modules or integrate other legacy systems inside the organization’s workflows.

Process data used by the application is organized into data collections and stored inside a distributed storage system. The storage system contains a global catalog that stores the collection metadata, security information and BWL program definitions. Each data collection store holds the entities processed by the system called documents, the associated versions and audit trail.

A DocuMentor implementation contains one or more processing nodes. Each processing node consists of a custom web server that hosts a web service endpoint, a BWL runtime instance and the data access components.

The client application implements the user interface of the platform. The main components of the client application are a scriptable client model, UI controls and service interface.

The DocuMentor platform can be extended using DMX components. A DMX component contains UI and processing assemblies. The UI assemblies add new features to the user interface by manipulating the scriptable client model and can communicate with external systems. The processing assemblies contain BWL statements that implement processing features and communication with external systems. These statements are used to integrate the new processing features inside the workflows. External systems can access the platform using the WS endpoint in order to access the data collections and workflows.

3. Information Security Characteristics for OBES

Although an OBES provides clear benefits in terms of flexibility and business efficiency, it also introduces specific challenges in security and information assurance. Because such systems are often used as the central hub of the enterprises IT infrastructure it’s critical that their information security characteristics are well understood and addressed in both design and implementation phases.

The most used model used to define security requirements and assist in covering all aspects of information security is the classic information security model called CIA Triad. The model defines three fundamental security characteristics: confidentiality, integrity and availability. The three characteristics defined by the model ([6], [7]) were adapted to the particularities of the OBES.

Confidentiality represents the assurance that information is not disclosed to unauthorized individuals, processes, or devices. For OBES this means that the documents stored in the data collections and the process definitions and instance data are protected by authentication and authorization mechanisms. These mechanisms must be present at all levels of the platform in order to protect from attacks that bypass the UI and use the WS endpoint or attacks that arise from the extension modules executed inside the processing nodes.
Integrity is the quality of an information system reflecting the logical correctness and reliability of the operating system, the logical completeness of the hardware and software implementing the protection mechanisms, and the consistency of the data structures and occurrence of the stored data. Integrity can be interpreted more narrowly to mean protection against unauthorized modification or destruction of information. The documents stored inside the data collections and the associated metadata stored inside the global catalog must be protected against both malicious and accidental alteration.

Availability represents timely, reliable access to data and information services for authorized users. This requires that the documents are accessible at all times to both final users and process instances and the system is resilient to attacks such as DoS, unexpected system failures and errors occurring in running processes. This can be achieved through isolation of the process instances, the use of multiple redundant processing nodes and distribution of data across multiple independent storage systems.

4. Methods Used to Implement Information Security at the Platform Level

The particularities of OPAS make implementing information security a challenging endeavor. The main challenges arise from the open nature of these systems and the fact that the system must support a variety of usage scenarios not known when the platform is designed and implemented. For example, the most common method used to implement confidentiality is making data incomprehensible by using symmetric or asymmetric encryption techniques. Unfortunately the encryption of all the data stored inside the system is impractical in most scenarios. In systems used for transaction processing the performance and flexibility cost encryption usually far outweigh the security benefits. Because of these restrictions precautions must be taken to provide a balance between the security features implemented at the platform level and the associated costs. In the design of the DocuMentor platform there are multiple methods implemented to ensure the security of the stored data without a major impact on performance and costs. The rules used to select the security features to be included inside the platform are:

- security features that are mandatory to implement one of the three information security characteristics must included in the platform;
- the features that have a major cost and can be implemented as extended are excluded from the platform;
- the features that would prevent the integration with other systems are not included at the platform level;
- The following paragraphs of this section present the security features included in the platform and the impact of these on the security characteristics.

Authentication of the users is performed using credentials based on user names and passwords. This method was chose because it offers a satisfactory level of security and allows easy integration with other systems. Additional authentication layers that use external authentication sources such as LDAP, digital certificates, biometrics or two way authentication can be implemented as extension modules.

Authorization is performed using permission objects. The permissions are triplets that associate an authenticated user with a target object and an operation. The presence of a permission object allows the user to execute the specified operation on the target object and its descendants. Authorization rules are enforced on all levels of the platform for final users, external systems that access the WS endpoint and operation executed by the DMX extension components.

For additional control a rule based filtering system was added for data collections. This allows administrators to use complex business rules to determine
the accessible objects for users at the data collection level. For example, an administrator could specify that user Bob has create, read and modify permissions for the Invoices data collection, but only for the invoices created by him and with a value less than 1000 USD.

All documents stored inside the system are versioned. Each modification of a document results in the creation of a new document version. The platform stores the current version of the document and RFC3284 ([2]) compliant diffs.

All operations performed by the system are recorded into an audit log. The audit log retains the information required to identify the user, its location and the modifications performed. The audit log, along with the document differencing system, has major information security implications:

- deters the authorized users from abusing or misusing their privileges;
- provides data that allows detection of security breaches and post factum analysis;
- prevents attacks because the users are more reluctant to perform unauthorized operations when they know that their actions are tracked;
- provides all the data necessary to revert unauthorized or accidental data modifications.

Parameters for all the operations performed by the system are validated according to the specified metadata and parsed to eliminate possibility of injection attacks. This raises the security level of the system by ensuring that the operation parameters are correct and do not pose a security threat. Additional business specific validations can be implemented using the BWL language.

Unlike other mainstream workflow languages, the BWL language is designed to prevent user from creating programs that structural control flow problems. A detailed analysis of the soundness property for BWL programs is presented in [5]. This increases the platform availability by eliminating the heavy resource usage required for executing malformed workflows.

In order to ensure scalability and availability of the platform, DocuMentor features a distributed processing and storage architecture. Data is spread over multiple collection data stores that can operate independently. Processing nodes are designed to be stateless, can access any data collection and execute any workflow instance. Each processing node can execute contains a replica of the global catalog and can operate even if the global catalog storage is temporary unavailable. Because the processing nodes are stateless they can be used in conjunction with a load balancer to distribute the processing load and make node failures transparent to the service user.

<table>
<thead>
<tr>
<th>Authentication subsystem</th>
<th>*</th>
<th>+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorization subsystem</td>
<td>*</td>
<td>+</td>
</tr>
<tr>
<td>Rule based filtering</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Document versioning</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Audit subsystem</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Validation and Parsing</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>BWL Soundness Property</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Distributed processing and storage</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Correlation between security features and security characteristics

Table 1 presents the correlation between the security features implemented in the BWS - DocuMentor platform and the identified security characteristics. Features required for meeting the minimum requirements of a
characteristic are marked with * and the ones that improve the level are marked with +.

5. Enhancing Security by Using DMX Extension Modules

Although the task of securing an open system imposes more challenges than traditional enterprise applications, it also offers the unique opportunity to implement security features beyond the ones included in the base platform by using the extension mechanism. The BWS – DocuMentor platform offers the possibility to create extension components that address implementation specific security issues.

Examples of such DMX components created are a generic digital signature and encryption module named DocSec and a domain specific loan modification authorization module named HampMod. The DocSec module consists of a client component a server component. The client component uses the DocuMentor client model to implement the following features:

- allows the user to select a certificate from the available CryptoAPI providers and use the certificate along with the private keys to sign an optionally encrypt the document;
- when the user tries to access a protected and / or signed document, the module retrieves the user certificate and use it to validate the signature and decrypt the document if necessary.

The server module of the DocSec module extends the BWL language with statements that validate the digital signatures using a list of trusted certificate authorities and allow extraction of data from certificate and signature. These statements are used to implement validation and processing rules based on the information contained in the signature.

The HampMod is an example of server side only extension component for DocuMentor that is used to authorize loan modifications for the Home Affordable Modification Program (a federal program of the United States). The module is used to authorize the creation and of loan modification forms stored as documents inside the platform. This authorization is done in addition to the standard authorization provided by the platform which is used for user authorization. The BWL statements added by the extension module implement business specific authorization rules. In order to validate the operations performed on the loan modification forms they gather borrower information from external banking systems and perform a NPV analysis in order to assess the borrower eligibility.

6. Implementation Methodology

Projects that aim to integrate enterprise applications are generally very complex and dynamic. The most known and used methodology for the implementation of integration projects is GERAM (Generalized Enterprise Reference Architecture and Methodology, [8]) developed by the IFAC / IFIP Task Force on Architectures for Enterprise Integration. This involves creating a global business model based on generic enterprise models (GEM), the generic modules (GM) and some specific ontological theories (OT). The integration solution is then developed based on this model. The traditional approach based on collecting all the requirements the proceeding with the system design, implementation and testing have failed in most situations. According to [9] and [10] approximately 70% of integration projects have failed. The main cause of failure was the lack of tools, methodologies and qualified personnel.

To reduce risks and increase the success rate of integration projects based on the BWS architecture and DocuMentor platform a new iterative development methodology called DM-CPI (DocuMentor - Continuous Process Improvement) was created.
DM-CPI methodology proposes a gradual approach to enterprise integration. Unlike traditional holistic approaches, DM-CPI does not propose to complete the initial modeling of the organization and its processes in a waterfall project. The proposed method is iterative. The integration project is implemented in successive iterations aimed at integration or improving of a particular process within the organization. Figure 6 shows the four stages that make up a DM-CPI iteration: identification of opportunities and requirements, process design and implementation, business process usage and evaluation.

The first stage identifies the opportunities and requirements and selects a process or a group of processes that will be the goal of the iteration. For each process the impact on the organization's performance and effort required for implementation is assessed. The processes with the greatest benefits / effort report are chosen for implementation. For the selected processes we determine the objectives to be achieved, the integration techniques and acceptance criteria.

The process design and implementation phase involves the development and testing of the processes selected in the identification phase using the tools provided by the DocuMentor platform.

In the business process usage phase processes created in the previous phase are put into production use. During this period we collect information relevant to the process execution by using the platform audit facilities and by interviewing people involved in the process.

In the measurement and evaluation phase data collected during the previous phase is analyzed. The purpose of this stage is to highlight the problems occurring in the implementation process and to identify improvement opportunities. The results obtained in this stage are used as input to the next iteration of the integration process.

The iterative nature of the proposed methodology increases the level for all three security characteristics: confidentiality, integrity and availability. This increase happens because each iteration introduces only small changes to the overall system and the security impact of the changes can be thoroughly analyzed before the new version enters into production use.

7. Conclusion

The present paper has presented the BWS architecture for OBES and the DocuMentor implementation and identified the information security characteristics required for such systems. Based on the characteristics the paper presents the methods used to implement the security features for the DocuMentor platform, the selection rules and the correlation between features and characteristics. The final section demonstrates how the extensibility features of an OBES can be used to enhance security beyond the features provided by the base platform.

References


