INTRODUCTION

Enterprise Application Integration (EAI) is defined as the process of coordinating and synchronizing multiple heterogeneous software applications (Serain, 2001). Enterprise Systems (ES) are proposed as integrated Information Systems (IS), covering most of the enterprise business processes (Sandoe, 2001). IS frameworks and Enterprise Architectures (EA) for modeling business processes can be used to model different aspects of EAI. Considering several of these frameworks, which have been extensively compared in (Losavio et al., 2005) and in particular the integration approaches proposed by the Brown’s Conceptual Model of Integration (BCMI) (Brown...
et al., 1994) and the Integration Views of Sandoe (Sandoe, 2001), we have defined a business integration framework, the EAI Framework (EAIF), expressed as a standard UML (Unified Modeling Language) model (OMG, 2003).

EAIF can be considered an integration-oriented EA framework, since it models processes, services and mechanisms for ES. The term framework means in general a set of assumptions, concepts, values, and practices that constitutes a way of viewing reality (O’Rourke et al., 2003).

The main contributions of EAIF are to unify the EAI terminology using the UML standard and to provide support to detect inconsistencies in the definition of integration processes, the services adapted to these processes and the mechanisms implementing these services.

The main goals of this work are on one hand to present the EAIF textual specification and instantiation guidelines and on the other hand, to experiment the modeling of a CRM system instantiating the EAIF specification. The case study is a J2EE-based Customer Relationship Management (CRM) System, called J-énésis, for customer retention and loyalty. It follows an upward/forward process within a training organization and provides forward services (Acuña &Rodriguez, 2003).

This paper is structured as follows, besides this introduction and the conclusions: Section 2 presents the EAIF overview and specification. Section 3 describes the case study and its instantiation.

**EAIF**

EAIF is a business integration framework specified in UML which provides standard and unified definitions of the ES elements related with people, processes, services and mechanisms for EAI (Losavio et al., 2002; Losavio et al., 2003). It is an extension of the BCMI (Brown et al., 1994) for integrated CASE technology. Two levels of this model, process and services, were extended with the three integration views proposed in (Sandoe, 2001): the backward, forward and upward views. Backward and forward integration refer to the integration of the internal and external business aspects, respectively. Upward integration considers those aspects related to business management. However, the mechanisms level was found independent from the Sandoe’s views (Losavio et al., 2002). BCMI was also extended with the people level, concerning the human factors (Losavio et al., 2005).

UML was selected as the specification language for EAIF because it is now widely used as an industrial standard modeling language. It supports aggregation and specialization and it provides extension mechanisms. In this sense, EAIF is flexible to technological evolution since it can be easily extended with new information technology mechanisms.

In this work, the EAIF UML model has been complemented with a textual specification of the main classes and guidelines to facilitate the instantiation process of the framework. The textual specification is simply an informal structured text or template, complementing the UML class specification. The idea was to avoid the complexity involved in the UML class diagrams, leaving in the diagrams only the class boxes with its name and relations, improving the legibility and facility of use of the framework. It is clear that a formal specification language could be recommended in this case, but it is outside the scope of this work, which focuses mostly a simple instantiation experience, according to precise guidelines.

This section presents an overview and the main parts of the textual specification of the framework.

**EAIF Overview**

Figure 1 presents the UML (OMG, 2003) class diagram of EAIF. In order to reduce complexity, only the classes, relationship and role names are shown. Notice that the navigability of the associations is bidirectional and that only a limited range of very well known processes, services and mechanisms is shown in the class diagram. Some of the classes could be further specialized, such as the «Sales and Marketing Management Process» class which can be specialized into the classes Sales Management Process and Marketing Management Process. They are not shown to abridge the presentation.

**EAIF Specification**

In what follows, a textual specification of the classes and relationships, complementing the UML diagram, is presented. Figures 2, 3, 4 and 5 show diagrams with the main classes of figure 1, where the cardinality is omitted and only the specification of the class attributes is provided for each class, indicated by the arrow. The attribute name is in italic and their description is in no italic.

In what follows only the classes that will be used to instantiate the J-énésis the case study (see Section 3) are specified, in order to abridge the presentation.

*Forward Processes* class extends *Processes.*

*Upward Processes* class extends *Processes.*

*Sales and Marketing Management Process* class extends *Upward Processes.*

The «solution» attribute of the Patterns class is a reference to a known pattern catalogue or library, containing components, connectors, behavior, quality properties and any other information related to the pattern. The structure of this attribute depends on the structure of the catalogue (Buschmann et al., 96; Gamma et al., 95; Schmidt et al., 2001). The components of the Information Technology class are complex elements that can be further specified if necessary and they should be customized for each instance.

Architectural class extends Patterns
Design class extends Patterns

Table 1 shows the specification of the relationships between People, Processes, Services and Mechanisms classes of figure 1. Notice also that in the associations, multiplicity i and role i are related to association end i.
Figure 2. Specification of the Processes level

Figure 3. Specification of the Services level.
Figure 4. Specification of the Mechanisms level.

Figure 5. Specification of the People level.

Table 1. Specification of the relationships between People, Processes, Services and Mechanisms classes.
CASE STUDY DESCRIPTION AND INSTANTIATION

In this section the CRM system case study is presented. The Customer Relationship Management (CRM) approach is a new widely used business strategy and still evolving. It is centered on the customer and its relation with the organization (Turban et al., 2005). CRM addresses all the customer touch points, such as face-to-face, Internet, or phone (Sandoe, 2001). It integrates sales, marketing and service strategies. It helps to establish collaborative relationships with customers on a long-term basis, using information technology such as databases, data warehouses, and data mining.

In general, CRM solutions consider three phases (Tiwana, 2001):

Acquisition: the organization acquires new customers. The organization competes to provide better product/service according to the customer needs.

Enhancement: it increases sales per customer. It gives good supplies at low cost. It has built-in pricing flexibility.

Retention and Loyalty: it requires customer knowledge to build service adaptability and use incentives to retain customers, such as the creation of new products. The ability to retain customers is a major determining factor.

J-énesis is a CRM system covering the services for the customer retention and loyalty phase. It offers, through Internet, registration services to attend software courses, providing products and services that consider customer needs (profiles) (Acuña & Rodriguez, 2003). Java 2 Platform Enterprise Edition (J2EE) (Szyperski, 2002) is used as the integration mechanism.

Processes

The J-énesis application supports the following processes:

Marketing: to provide data on software courses, customers’ profiles and customer retention and loyalty strategies for marketing decision making.

Customer service: to provide customer satisfaction using the information about products, services and customers’ profile.

J-énesis provides facilities to gain customer knowledge to build service adaptability, Deliver new products that meet current customers’ needs and create and transmit incentives to retain customers. However, it does not support a call center service.

The users of J-énesis are: marketing analyst, customer service analyst and customer.

Services

The following services are identified:

Collect, recall and update customer profile: personal information, preferences, capabilities, and markets and business environment data.

Collect, recall and update software courses: information on software courses and services according to the customer’s profile.

Collect, recall and update customer retention strategies: to deal with encouragement strategies to maintain customer loyalty.

Historic: all transactions are stored in databases for further analysis purposes.

Subscribe/unsubscribe software courses: this service allows customer to subscribe/unsubscribe software courses according to the customer’s requirements.

Note that in general, the collect, recall and update operations are required to be reliable, efficient and attractive; historic transactions must be reliable and efficient, and finally efficiency is also required for the subscribe/unsubscribe operation.

Architecture

J-énesis is structured according to a three-layer (Shaw & Garlan, 1996) architecture. The layers are user interface, data and business logic. Some of the design patterns used are Persistent Data Manager and Observer (Rogers, 1997).

Information Technology

J-énesis uses the following technology:

The J2EE platform is essentially a distributed computing platform; Java-centric environment from SUN, for designing, developing, building and deploying component-oriented ES. A J2EE application is a collection of software components that are engineered to be distributed across multiple computing tiers over the Internet or Intranet (Szyperski, 2002).

The java web server is Tomcat 4.0.x (Apache, 2002).
The following J2EE 1.3 APIs were used in the development system:

- **JDBC 2.0 (Java Data Base Connectivity)** for database connectivity between the Java and MySQL.

- **Java Servlets 2.3** extends the functionality of a Web server, generating dynamic content (*HTML and XML*) with Java Server Pages (*JSP*) 1.2

- **JavaMail 1.2** and interacting with web clients using Hypertext Transfer Protocol (*HTTP*).

The Simple Object Access Protocol (SOAP) is the protocol for communication between applications. SOAP is a platform independent, language, based on XML.

The *Java™* language was used to develop the application (SUN, 1997).

**EAIF instantiation with the J-énesis case study**

In this section, the EAIF instantiation with J-énesis is presented. Notice that the textual specification is directly used instead of UML instance diagrams, in order to simplify and abridge the presentation.

The EAIF classes are instantiated with J-énesis according to the guidelines presented in Table 2.

<table>
<thead>
<tr>
<th>Step</th>
<th>Classes instantiation</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Forward/Customer Service Process</td>
<td>Web services related to: Collect/retrieve data on customer profile Documentation on software courses Subscribe/unsubscribe software courses</td>
</tr>
<tr>
<td>2. Identification of the roles involved in the processes (<em>People class</em>)</td>
<td>People {role = “Marketing Analyst”}</td>
<td>Responsible for establishing policies and services for customers retention and loyalty, which will be applied to software courses sale.</td>
</tr>
<tr>
<td></td>
<td>People {role = “Customer”}</td>
<td>Use services concerning software courses</td>
</tr>
<tr>
<td></td>
<td>People {role = “Customer Service Analyst”}</td>
<td>Responsible for applying policies, and service about software courses to customers established by marketing analysts.</td>
</tr>
<tr>
<td>3. Identification of the services: Backward, Forward and Upward (<em>Services class</em>)</td>
<td>Forward/Customer Relationship Management Systems</td>
<td>J-énesis is a CRM System for the retention and loyalty phase</td>
</tr>
<tr>
<td>4. Identification of the architecture of the services used in the processes (<em>Mechanisms class</em>)</td>
<td>Architecture {style = “layers”}</td>
<td>Three layers: User Interface, data and business logic</td>
</tr>
<tr>
<td></td>
<td>Pattern/Design/ Persistent Data Manager</td>
<td>It controls access to a collection of independent, fine-grained, persistent objects required by the J-énesis services</td>
</tr>
<tr>
<td></td>
<td>Pattern/Design/ Observer</td>
<td>It provides an automatic event notification mechanism, to inform customers about software courses update</td>
</tr>
<tr>
<td>5. Identification of information technologies implementing the services (<em>Mechanisms class</em>)</td>
<td>Information Technology Hardware/PCs Software/Java Software/HTML Software/XML API’s Software/Java Servlets 2.3 Software/JSP 1.2 Software/ JavaMail 1.2 Database/ MySQL Communication/ JDBC 2.0 Communication/ HTTP Communication/ SOAP</td>
<td>Business rules (organization strategies and policies)</td>
</tr>
</tbody>
</table>

---

*Table 2. EAIF instantiation process with the J-énesis case study.*
Notice that for this application, the subclass Marketing Management Process of class Sales and Marketing Management Process is instantiated as an upward process.

Figure 6 shows Marketing_upward_process class specification.

Figures 7 (part a – part b) show of J-énesis_forward_process class specification. Figures 8, 9 and 10 show the instantiation of the Services, Mechanisms and People classes, respectively.

Note that services functionality 2 and 3 have similar quality properties even if they are used by different people; in Collect, recall and update software courses and Collect, recall and update customer retention strategies only recall is performed by Customers; the others are performed by

```
Marketing_upward_process class extends Marketing Management Process
process_model_name = "Marketing model"
process_goal = "this process manages the information related with software courses, customer profile and customer retention and loyalty strategies"
process_strategy = "provide response and effective assistance to clients according to retention strategies and maintain information related with customers and courses"

Process model elements01 class instantiates Process model elements

structural_unit_of_process_model: Array of structural_unit_of_process_model01 = "step"
table_of_process_model_elements: Array of structural_unit_of_process_model01

Marketing up ward_process class extends Marketing Management Process
process_model_name = "Marketing model"
process_goal = "this process manages the information related with software courses, customer profile and customer retention and loyalty strategies"
process_strategy = "provide response and effective assistance to clients according to retention strategies and maintain information related with customers and courses"

Process model elements01 class instantiates Process model elements

structural_unit_of_process_model: Array of structural_unit_of_process_model01 = "step"
table_of_process_model_elements: Array of structural_unit_of_process_model01

Marketing up ward_process class extends Marketing Management Process
process_model_name = "Marketing model"
process_goal = "this process manages the information related with software courses, customer profile and customer retention and loyalty strategies"
process_strategy = "provide response and effective assistance to clients according to retention strategies and maintain information related with customers and courses"

Process model elements01 class instantiates Process model elements

structural_unit_of_process_model: Array of structural_unit_of_process_model01 = "step"
table_of_process_model_elements: Array of structural_unit_of_process_model01
```
J-énesis_forward_process class extends Customer Service

process_model_name = "based in CRM"

process_goal = "this process administers the information related with software courses, customer profile and customer retention and loyalty strategies. Also, it allows the access to remote resources to customer"

process_strategy = "increase customer retention and loyalty using incentives and allow access to relevant information on client and courses through Internet"

Process_model_elements02 class instantiates Process_model_elements


diagram:

```plaintext
Table of Process Model Elements:

<table>
<thead>
<tr>
<th>Structural Unit Name</th>
<th>Activity</th>
<th>Input</th>
<th>Product</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collect and update software courses data and retention and loyalty strategies data</td>
<td>Collect software courses data</td>
<td>Software courses data</td>
<td>MySQL management system, web interface</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Recall customers data</td>
<td>Customer identifier, Software courses data</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Update retention and loyalty strategies data</td>
<td>Customer identifier, Retention and loyalty strategies data</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Update retention and loyalty strategies data</td>
<td>Customer identifier, Retention and loyalty strategies data</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Structural Unit Participant:

J-énesis_process_people02, J-énesis_process_people03
```

Figure 7. EAIF instantiation J-énesis_forward_process class with J-énesis (part a).

Customer Service Analysts. Portability is transparent, for the use of the Java language and usability is required by both kinds of people since the have to change data and consult (maintainability) directly on-line; recoverability is desired for fault tolerance and efficiency is provided by the network context.
Figure 7. EAIF instantiation J-énesis_forward_process class with J-énesis (part b).
J-énesis_service01 class **instantiates** Customer Relationship Management System functionalities: Array \([1..n]\) of functionality **where**

functionality \([1]\) = \(\{\text{"Collects, recall and update of customer information"}, \text{"reliability/availability"}, \text{"reliability/recoverability"}, \text{"efficiency"}, \text{"usability"}, \text{"portability"}, \text{"maintainability"}\}\)

functionality \([2]\) = \(\{\text{"Collects, recall and update of software courses"}, \text{"reliability/availability"}, \text{"reliability/recoverability"}, \text{"efficiency"}, \text{"usability"}, \text{"portability"}, \text{"maintainability"}\}\)

functionality \([3]\) = \(\{\text{"Collects, recall and updates customer retention strategies"}, \text{"reliability/availability"}, \text{"reliability/recoverability"}, \text{"usability"}, \text{"efficiency"}, \text{"portability"}, \text{"maintainability"}\}\)

functionality \([4]\) = \(\{\text{"Historic data registration"}, \text{"reliability/availability"}, \text{"reliability/recoverability"}\}\)

functionality \([5]\) = \(\{\text{"Subscribe/unsubscribe software courses"}, \text{"reliability/availability"}, \text{"reliability/recoverability"}, \text{"usability"}, \text{"efficiency"}, \text{"portability"}, \text{"maintainability"}\}\)

**Figure 8.** EAIF instantiation services class with J-énesis

J-énesis_mechanism class **instantiates** Mechanisms
integration_method = \(\{\text{"message passing"}, \text{"GUI"}\}\)
communication_model = \(\{\text{"synchronous"}\}\)
quality = \(\{\text{"J2EE compliant"}, \text{"portability"}, \text{"interoperability"}, \text{"maintainability"}\}\)
architecture: J-énesis_architecture
IT: J-énesis_information_technology

J-énesis_architecture class **instantiates** Architecture
description = \(\"J-énesis components and connectors\"
architecture_style = \"layers\"
problem_context = \(\text{"Customer Relationship Management Systems"}\)
quality = \(\text{"maintainability"}\)
patterns = \{Pattern01, Pattern02\}

Pattern01 class **instantiates** Persistent Data Manager
pattern_name = Persistent Data Manager
problem_description = \(\text{(Rogers, 1997)}\)
solution = \(\text{(Rogers, 1997)}\)
quality = \(\{\text{"reliability"}, \text{"maintainability"}\}\)

Pattern02 class **instantiates** Observer
pattern_name = Observer
problem_description = \(\text{(Rogers, 1997)}\)
solution = \(\text{(Rogers, 1997)}\)
quality = \(\{\text{"reliability"}, \text{"maintainability"}\}\)

J-énesis_information_technology class **instantiates** Information Technologies
hardware = \{\text{windows PC’s}\}
software = \{\text{Java, HTML, Java Web Server Tomcat4, Java Servlets 2.3, Java Server Pages (JSP) 1.2, Java Mail 1.2}\}
communications = \{\text{HTTP, TCP/IP, JDBC 2.0}\}
database = \{\text{MySQL}\}\)

**Figure 9.** EAIF instantiation mechanisms class with J-énesis.
J-énesis_people01 class **instantiates** People
role = “Marketing analysts”
goals = “They establish policies, products and service to customers”

J-énesis_people02 class **instantiates** People
role = “Customer Service Analysts”
goals = “They apply policies, products and service to customers, established by Marketing analysts”

J-énesis_people03 class **instantiates** People
role = “Customer”
goals = “Customer define the business requirements and expectations. Customer subscribe/unsubscribe software courses using internet”

**Figure 10.** EAIF instantiation people class with J-énesis.

J-énesis_process_people_role02 association **instantiates** process_people_role
association end 1 = J-énesis_people02 – Customer Service Analysts
association end 2 = J-énesis_forward_process – CRM processes
association name = “processes_people_role”
type = “association”
role 1 = “drive”
role 2 = “are_driven”
multiplicity 1 = 1
multiplicity 2 = 1

J-énesis_process_people_role01 association **instantiates** process_people_role
association end 1 = J-énesis_people01 -- Marketing Analysts
association end 2 = Marketing_upward_process –Marketing decision making process
association name = “processes_people_role”
type = “association”
role 1 = “drive”
role 2 = “are_driven”
multiplicity 1 = 1
multiplicity 2 = 1

J-énesis_process_people_role03 association **instantiates** process_people_role
association end 1 = J-énesis_people03 --Customer
association end 2 = J-énesis_forward_process – CRM processes
association name = “processes_people_role”
type = “association”
role 1 = “drive”
role 2 = “are_driven”
multiplicity 1 = 1
multiplicity 2 = 1

**Figure 11.** EAIF instantiation of the relationship between Processes, Services, Mechanisms and People classes with J-énesis.

Figure 11 shows the instantiation of the relationship between Processes, Services, Mechanisms and People classes. Notice that «mechanisms» are transparent to «Marketing Analysts», «Customer Service Analysts» and «Customer», which means that the association mechanisms_people_role is not applicable in this case. This is due to the fact that only the production stage and not the development stage of the system, is considered.

It is important to notice that the EAIF instantiation allows to specify the complete processes, services, mechanisms and people performed by a training organization, even if some of the processes are not supported by explicit J-énesis services; for example, the process «Marketing_upward_process». However, since this process is required by the business activity in the organization, the EAIF process instance with the J-énesis process could recommends the implementation of additional support services as a business strategy. For example, an ERP System and a Decision Support System could be integrated to the CRM System.
During the instantiation process we have observed that the information related to the case study must be available and domain experts’ support is highly recommended. The goal is to avoid the modeling of unreal situations.

The EAIF specification and instantiation guidelines allow the modeling of all the concepts for this case study. The analysis obtained after the instantiation process (Table 2) is presented in what follows:

Although during the instantiation was not necessary to add new elements to EAIF, we consider it extensible because it is specified in UML. New classes could be added to EAIF, using the aggregation/composition and generalization/specialization mechanisms, if needed.

All the elements of the applications could be represented by EAIF. Nevertheless we cannot conclude its completeness because further study is needed.

EAIF was easy to use, because besides the UML model enriched with a textual specification, guidelines were provided for the instantiation process.

CONCLUSIONS

The main contributions of EAIF are to offer a standard, organized and unified view of the main aspects involved in an integration project and to provide support to detect inconsistencies monitoring the changes in the definition of integration processes, the services adapted to these processes and the mechanisms implementing these services. EAIF is specified as a standard UML model to provide this standard organized and unified view, and it is complemented with a textual specification and guidelines to facilitate the instantiation process. UML was selected as the specification language because it is an accepted industrial standard modeling language and it provides extension mechanisms to support the technology evolution. This is reflected in changes at the EAIF mechanisms level, where new
information technology mechanisms can be easily added to the model. It is clear that formal specification languages could be recommended instead of UML and the textual specification, but it is outside the scope of this work, which focuses mostly a simple instantiation exercise according to precise guidelines, that can be easily followed and reused by practitioners. Notice that the instantiation process provided can be considered an iterative process, if the iteration is performed for each process at process level with its services and iterating again for each services, considering the mechanisms involved. The process is also incremental if intermediate products are considered in the iterations.

The instantiation of EAIF with a real operational forward integration service system as case study, a J2EE-based CRM system for customer retention and loyalty, respectively, was presented to experiment the applicability of the framework. It is obvious that the EAIF maturity will improve as more case studies will be considered. Moreover, we feel that the instantiation guidelines could be further refined in order to minimize the need for a domain expert. At present, EAIF has no automatic tools support, but its standard model can be used to define the conceptual base for these tools.

The EAIF application to a case study pointed out some aspects of the CRM system studied, such as processes that are not supported by services. This aspect can help process engineers and designers to review their automation strategies. In this sense, EAIF is a useful tool to detect and document business processes.

Future work will focus on the extension of the specification, using the UML extension mechanisms. Stereotypes can be defined and OCL (Object Constraints Language) can be used at the mechanisms level, to formalize the quality requirements specification for the architectural solutions, where standard quality models can be also used at this stage. The extended specification of EAIF can be used to describe the transformations between the EAIF levels and the CIM, PIM and PSM models in a Model Driven Architecture (MDA) (OMG, 2001) context.

REFERENCES


