

Comparison of EAI* Frameworks

Francisca Losavio, Centro ISYS, LaTecS, Facultad de Ciencias, Universidad Central de Venezuela, Caracas

Dinarle Ortega, Departamento de Computación, Universidad de Carabobo, Valencia, Venezuela

María Pérez, Departamento de Procesos y Sistemas, Universidad Simón Bolívar, Caracas, Venezuela

Abstract

EAI (Enterprise Application Integration) is defined as the set of plans, methods and tools aimed at modernizing, consolidating, integrating and coordinating information systems within an enterprise. An EAI Framework (EAIF) is presented to provide sound and unified definitions of the modeling elements involved in the EAI approach. The goal of this work is to compare EAIF with other widely accepted frameworks that can be used to model EAI. A set of reusable features are defined to identify the main weakness and strength of these frameworks. As a result, EAIF has been enriched with the “human aspects” feature.

1 INTRODUCTION

It is a normal practice of commercial organizations to apply Information Technology (IT) for supporting their whole business process, trying to solve complex problems that are common in increasingly competitive worldwide business environments. Information must be “shared” within the organization to guarantee a better planning, control and evaluation of the work processes, inside and outside the organization. However, many existing Information Systems (IS) supporting these processes, are considered “automation islands”, since they cannot communicate easily with systems inside the organization and even less outside it, with external systems of clients and suppliers. In order to provide a complete, efficient and reliable support, the IS must be integrated. IS integration means to unify independent IS, with the purpose of providing shared information and give a valid support to the whole organizational process. The desired result is an Enterprise System (ES), covering most of the enterprise business processes [Sandoe01]. ES crosses the boundaries of traditional business functions (such as marketing and finance) in order to reengineer and improve vital business processes all across the enterprise [O’Brien03],

* Enterprise Application Integration

such as Enterprise Resource Planning (ERP) [Sandoe01] and e-business solutions [Whitten04]. The term *EAI* refers to the *plans, methods and tools aimed at modernizing, consolidating, integrating and coordinating IS within an enterprise, where standards play an important role* [McKeen02].

One of the most crucial points to achieve EAI, is to combine all the organization assets or expertise, including IT (such as data base technology, distributed and real-time computing, communication technology) and organizational (such as Business Process Reengineering and Workflow Redesign), to support the complexity of all the organizational processes. Notice that the integration of IS requires also more traditional technology like middleware for databases, user interface standards and middleware for real-time and distributed applications, based on adapters and brokers [Themistocleous01].

Since EAI is a new approach to integration, it lacks formalization, organization and unification of the related concepts [Laudon04], [O'Brien04], [Sandoe01], [Whitten04]. The integration problem has not been entirely resolved and it is still a very expensive process in terms of human and technological efforts. By far, the largest intangible benefit to enterprise integration is the recent improved availability of the information that is shared by the organization using integrated IS and network communication facilities and the establishment of enterprise-wide standards for information resources [Sandoe01]. In consequence, the importance of integrating applications is at present a main concern of the software community, in particular the modeling of EAI for obtaining a more organized and unified view of the main aspects involved. The term *framework* is considered here as a set of *assumptions, concepts, values, and practices that constitutes a way of viewing reality* [O'Rourke03]. Important works have been proposed up to now and can be considered for EAI modeling, in particular, the works of [O'Rourke03], [Zachman03], [Whitten04], [Stojanovic01], [Stojanovic02], [Cummins02]. The main goal of this work is to compare an EAI Framework, which will be called EAIF, defined to unify the elements related with processes, services and mechanisms for EAI [Losavio02], [Losavio03], with these related works. Criteria are formulated to identify strength and weakness of these frameworks. EAIF was constructed extending the known Brown's Conceptual Model of Integration (BCMI) [Brown94]. As a result of the comparison, EAIF has been enriched with the "human aspects" feature.

This paper is structured as follows, besides this introduction and the conclusion: Section 2 describes EAIF. Section 3 presents a survey on common IS modeling frameworks that can be also used to model EAI: the Zachman, Whitten, Cummins and Zoran's frameworks. Section 4 presents an analysis of Zachman, Whitten, Cummins and Zoran's frameworks vs. EAIF. Section 5 contains the comparison of the frameworks studied, on the base of the comparison criteria formulated.

In what follows, the main characteristics of EAIF are presented.



2 EAI

EAI arises as an evolution of the CASE technology approach [Losavio02], [Losavio03]. Historically, in the 90's decade, the software engineering community proposed several approaches for integrated CASE environments [Brown94], [Pérez99]. EAI is presented as an extension of BCMI [Brown94] with the EAI views [Sandoe01]. Before discussing this framework in details, BCMI and the EAI views are briefly presented.

Brown's Conceptual Model of Integration (BCMI)

Brown [Brown94] proposes a conceptual model, or three-tier framework corresponding to different abstraction levels, for describing the integration of the tools constituting a CASE environment (see Figure 1). The *central level* contains the services (functionality of the CASE environment) offered to the final users. The *third level* contains the mechanisms used to implement the services. The *first and more abstract level* corresponds to the organizational process providing guidelines (goals in terms of steps and tasks, constraints) for selecting the services offered by the CASE system. The relation between the services level and the mechanisms level is an *implementation* relation. A service can be implemented by several mechanisms. The relation between process and service levels is an *adaptation* relation [Losavio02].

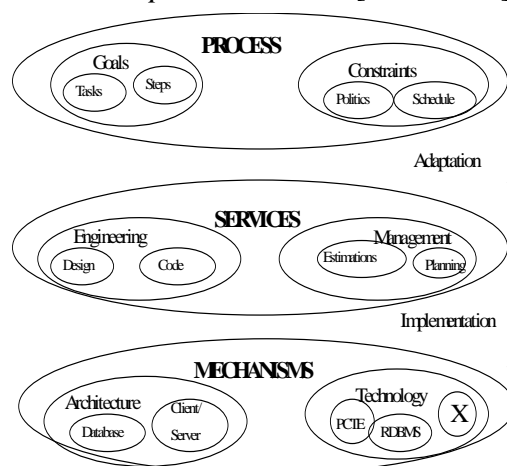
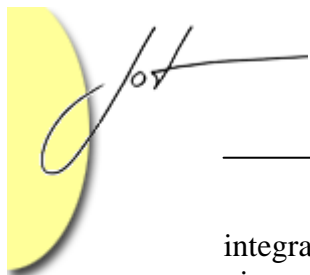


Fig. 1: Brown's Conceptual Model of Integration [Brown94]

The EAI views approach and its relation with BCMI is discussed in what follows.

The EAI Views Approach

In order to synthesize the different integration trends, three integration views are proposed in [Sandoe01], to allow software integration in an incremental way: *backward*, *forward* and *upward* views. The Backward Integration (BI) refers to the integration of the internal organizational processes. The Forward Integration (FI) refers to the integration of those organizational processes related to entities which are external to the organization, such as clients, partners and suppliers. The Upward Integration (UI) means the



integration of those organizational processes related to decision making. These three views will drive the extension of BCMI, in the sense that each model tier will be refined using the three EAI integration views.

Extension of BCMI with the EAI views

EAIF is proposed extending BCMI according to a top-down refinement of the Process, Services and Mechanisms levels of the Brown’s framework with the BI, FI and UI views.

1. The Process Level. In order to refine this level, the most commonly used organizational processes are described for each integration view. Notice that each business process has its own goals and constraints, and it depends directly from the specific functionality that is required by the integrated system (see table 1). *Business process* are the work , procedures, and rules required to complete the business tasks, independently of any information technology used to automate or support them [Whitten04]. Table 1 shows the processes (goals) and examples of sub-processes for each process [Mendoza96] in the three views BI, FI, UI, extending the Process Level of BCMI [Losavio02].

Level	Integration views		
	Backward	Forward	Upward
Process	<ul style="list-style-type: none"> • Sales and marketing: order processing, order tracking • Manufacturing/production: inventory management, purchasing, shipping, plant and equipment maintenance • Financial and accounting: account payable, account receivable, cash management, forecasting, product-cost accounting • Human resources: personal administration, personal planning, benefit accounting, training • Development and research: design and testing virtual prototyping of products, ergonomic analysis, simulation of assembly 	<ul style="list-style-type: none"> • Telemarketing • Web sales • Retail sales • Customer service 	<ul style="list-style-type: none"> • Sales: sales management, sales planning • Manufacturing/production: production planning, material requirement planning, inventory management • Financial and accounting: product-cost accounting, cost-center accounting, asset accounting, general ledger, financial report • Human resources: applicant-tracking, contract cost analysis

Table 1: Extension of the Process Level of BCMI with internal, external and management organizational processes

2. The Services level. In what follows, the extension of BCMI is presented at Services Level, with the three integration views. Table 2 summarizes: Services are presented in terms of the typical IS supporting the processes (see Table 1), according to the three integration views BI, FI, UI [Losavio02].

Level	Integration views		
	Backward	Forward	Upward
Services	<ul style="list-style-type: none"> • CASE tools • Legacy Systems • TPS (Transaction Processing Systems) • ERP (Enterprise Resource Planning) 	<ul style="list-style-type: none"> • CRM (Customer Relationship Management) systems • CEM (Customer Experience Management) systems • B2C (Business to 	<ul style="list-style-type: none"> • EIS (Executive Information Systems) • ESS (Executive Support Systems) • MIS (Management Information



		Consumer) systems <ul style="list-style-type: none"> • B2B (Business to Business) systems 	Systems) <ul style="list-style-type: none"> • DSS (Decision Support Systems) • KMS (Knowledge Management Systems)
--	--	---	--

Table 2: Extension of the service level of BCMI with the services supported by the internal, external and management organizational processes

3. The Mechanisms Level. According to Brown, two types of components are considered at this level: *architecture* and *technology*. Hence, a brief survey on software architecture and information technology used for integration will be given before presenting the extension of the mechanisms level.

- a) The Architecture (components and connectors with a behavior [Shaw96]) is the main mechanism to articulate the services supporting the organizational processes, according to BCMI [Losavio02]. *Architecture* is considered here as the structure of the components of a system, their interrelationships and guidelines governing their design and evolution over time. The architecture becomes the basis of systematic development and evolution of software systems [Ramdane-Cherif02], in particular, this applies to EAI applications. The architectural styles [Shaw96] considered for EAI are event-based and implicit invocation, layers, repository. The architectural patterns and design patterns [Buschmann96], [Schmidt01] that can be used for structuring EAI applications are: Broker, Microkernel, Reflection, Wrapper Façade, Component Configurator, Interceptor, Extension Interface, MVC, PAC [Losavio02], [Losavio03].
- b) The Technology Information (IT) is defined as a set of resources available for managing changes and to give support to people in the development of the activities related to an organization [Laudon04], [O'Brien04], [Sandoe01]. The following resources are considered: Hardware, Software, Database Technology, Communication Technology, Human Resources [Losavio02].

The BI, FI and UI views are not relevant at the extension of BCMI at Mechanisms Level. These views share the same mechanisms. Table 3 summarizes these results [Losavio02].

Level	Components	EAI
Mechanisms	Architecture	<ul style="list-style-type: none"> • Styles: layers, repository, event-based, implicit invocation • Patterns: distributed, adaptable, interactive and re-configurable
	Technology	<ul style="list-style-type: none"> • Hardware • Software: middleware, GUI and multimedia facility • Database technology • Communication technology • Human

Table 3: Mechanism Level of the BCMI for EAI

Figure 2 shows EAIF, which is constituted by BCMI (see Figure 1) extended with the EAI views, expressed in UML [Losavio03]. Notice that the Mechanisms Level, as the lowest abstraction level, holds the general styles and mechanisms which will be used to implement all the services corresponding to the three views of EAI. EAIF uses the ISO 9126-1 standard to specify the architectural quality related to ES [Losavio03].

The following section presents a brief description of the frameworks proposed by Zachman [Zachman03], Whitten et al. [Whitten04], Zoran et al. [Stojanovic02] and, Cummins [Cummins02] in chronological order.

3 REFERENCE FRAMEWORKS

Several frameworks are used to develop IS, in particular ES. The frameworks proposed by Zachman and Whitten are general for the IS development. Zoran instead proposes a specific framework for Component Based Development. Cummins proposes a management approach to integration and an Enterprise Integration Architecture.

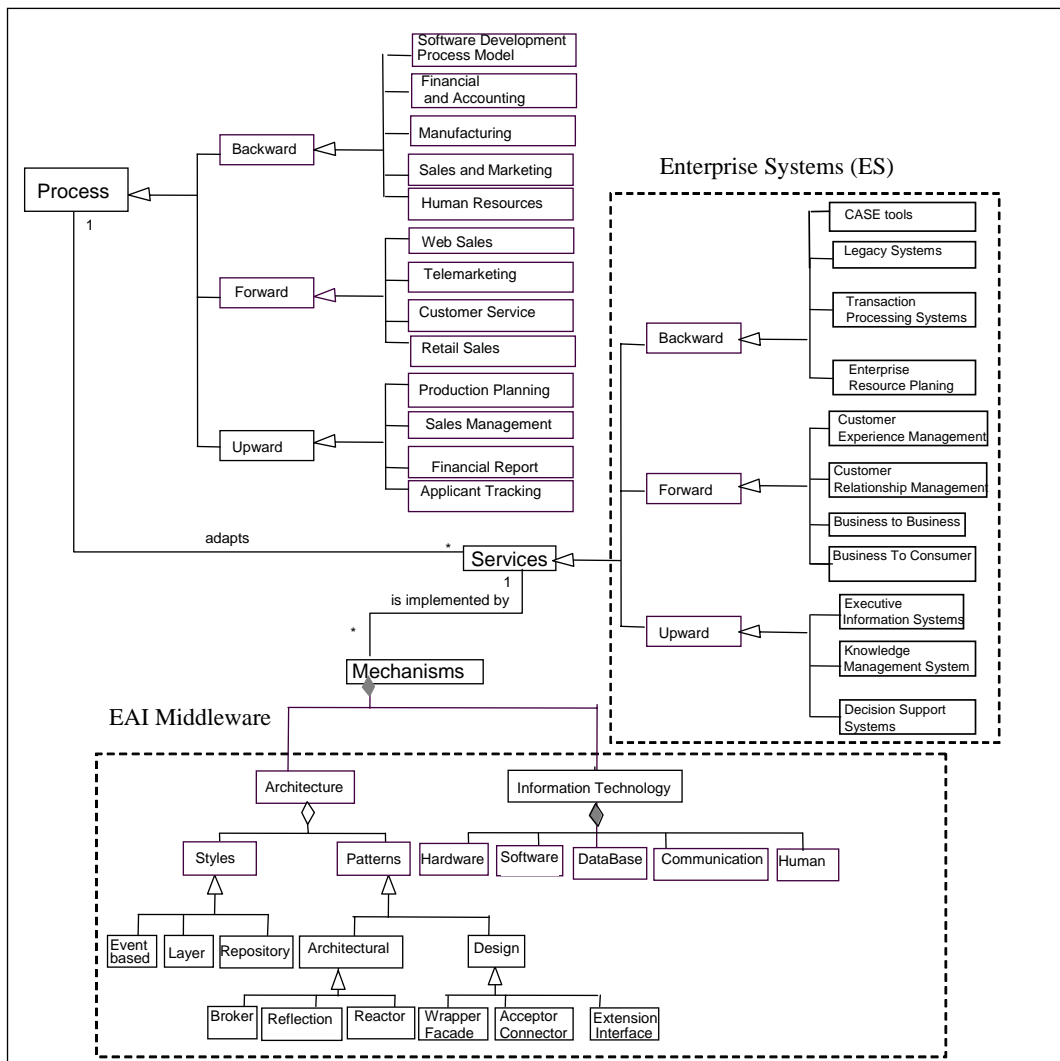
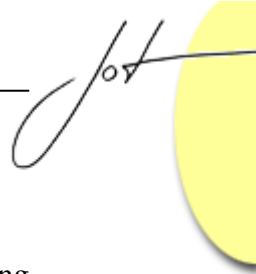


Fig. 2: UML representation of EAIF



Zachman's Enterprise Architecture Framework

This framework was proposed in 1980. It offers a classification scheme for organizing descriptive representations (models, pictures, diagrams or textual document) of an enterprise that produces a product or provides services (an ERP system, a payroll application, for example the Boeing 777 or a dry cleaning service). It has two dimensions, *perspectives* and *aspects* (see Figure 3). The main goal of this framework is to construct and understand the concepts involved in the enterprise architecture using a classification scheme for artifacts that describes the various types of designs. This framework presents a holistic view of the enterprise [O'Rourke03]. In particular:

The *perspectives* allow the visualization of a problem domain through the view of all the people involved: the "Planner", the "Owner", the "Designer", the "Builder", the "subcontractor" and the "Functioning Enterprise". These persons and their different interests must be understood in order to manage all their requirements. The descriptive representations of the final product or service of an enterprise are designed to express the concepts/constraints relevant to the different perspectives over the engineering and manufacturing processes [Zachman03]. The main perspectives are: "Scope (Row 1)", "Owner (Row 2)", "Designer (Row 3)", "Builder (Row 4)", "Out-of-Context (Row 5)" and "Functioning Enterprise (Row 6)".

The *aspects* or *abstractions* [Zachman03] refer to the questions required to understand each problem domain. They represent the independent variables, constituting a comprehensive description of the subject or object, including [O'Rourke03]: *What* (things, column 1), *How* (processes, column 2), *Where* (connectivity, column 3), *Who* (people, Column 4), *When* (timing, Column 5), *Why* (motivation, Column 6). All these aspects must be evaluated from the perspective of each person involved.

The Zachman's framework allows the modeling of different enterprise aspects from the perspectives of all the people involved in the organization; it is not intended explicitly for software systems integration.

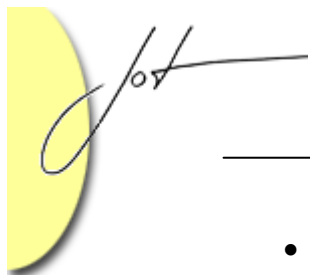
The following is a general framework proposed by Whitten for the development of different kinds of IS.

Whitten's framework for IS development

The first version of this framework was proposed in 1998; in this paper the last version is discussed (see figure 4) [Whitten04].

In what follows, a brief description of all the components, are presented.

- a) Players: Perspectives of the persons or "Players" involved in the development of IS (left side rows). They are: Project Manager, Systems Analyst, System Owner, System User, System Designer and System Builder.
- b) Business and Technology drivers: Perspectives of the Business drivers and the Technology drivers (superior and inferior columns, respectively). The superior columns of the framework represent the three main goals of IS:



- Improve Business Knowledge: data are processed to obtain information which is translated into knowledge.
- Improve Business Processes and Services: processes express the desired functionality of IS. The processes are realized by persons or computing systems.
- Improve Business Communications and Collaboration between the people involved. IS must efficiently provide user's and system's interfaces.

They conform the Business Drivers and refer to the management of business tendencies impacting IS, such as economy globalization, e-business, security, privacy, partners collaboration, knowledge management and business process reengineering. The following three columns are located in the inferior part of the framework:









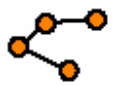





















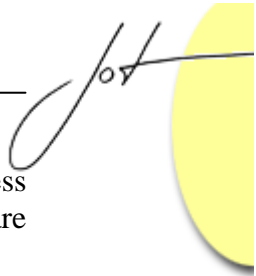
	Data What	Process How	Network Where	Role Who	Timing When	Motivation Why
Scope (contextual) Planner	List of things important to the business 	List of processes the business performs 	List of locations in which the business operates 	List of business responsibilities 	List of events significant to the business 	List of business goals/strategy 
Business model (conceptual) Owner	e.g., Semantic Model 	e.g., Business Process Model 	e.g., Logistics Networks 	e.g., Workflow Model 	e.g., Master Schedule 	e.g., Business Plan 
System model (logical) Designer	e.g., Logical Data Model 	e.g., Application Architecture 	e.g., Distributed System Architecture 	e.g., Human Interface Architecture 	e.g., Processing Structure 	e.g., Business Rule Model 
Technology model (physical) Builder	e.g., Physical Data Model 	e.g., System Design 	e.g., Configuration Design 	e.g., Presentation Architecture 	e.g., Control Structure 	e.g., Rule Design 
Detailed representations (out-of-context) Subcontractor	e.g., Data Definition 	e.g., Program 	e.g., Network Architecture 	e.g., Security Architecture 	e.g., Timing Definition 	e.g., Rule Specification 
Functioning enterprise	e.g., Data	e.g., Process	e.g., Network	e.g., Organization	e.g., Schedule	e.g., Strategy

Fig. 3: Zachman's Enterprise Architecture Framework [Zachman03]

- Database Technologies: capture, storage and use of business knowledge.
- Software Technologies: automation and support to business processes and services.
- Interface Technologies: support to business communication and collaboration.

They constitute the Technology drivers. They refer to IT advances impacting IS, such as networks, internet, object technology, mobile technology, collaborative technology.



Columns in the upper part use columns in the lower part. For example, Improve Business Knowledge uses Database Technologies and Improve Business Processes uses Software Technologies to automate the selected processes.

- c) Process activities: Perspective called “Process” of the activities related with the system’s development process (right side rows). These activities constitute the right hand side of the framework and are related with the software development process. They are:
 - General activities: project management, process management, feasibility analysis, fact-finding, system initiation, system analysis, system design and system implementation.
 - Software development: scope definition, problem analysis, requirements analysis, logical design, decision analysis, physical design, construction and testing, installation& delivery [Whitten04].
- c) Product: Perspective called “Product” related to the Architecture Framework for IS (center cells): Information Scope & Vision, Functional Scope & Vision, Communications Scope & Vision, Business Data Requirements, Business Process Requirements, Business Interface Requirements, Database Design, Business Process Design, Software Design, Interface Design, Database e Interfaces Solutions, Commercial Software Packages and Custom-Built Applications.

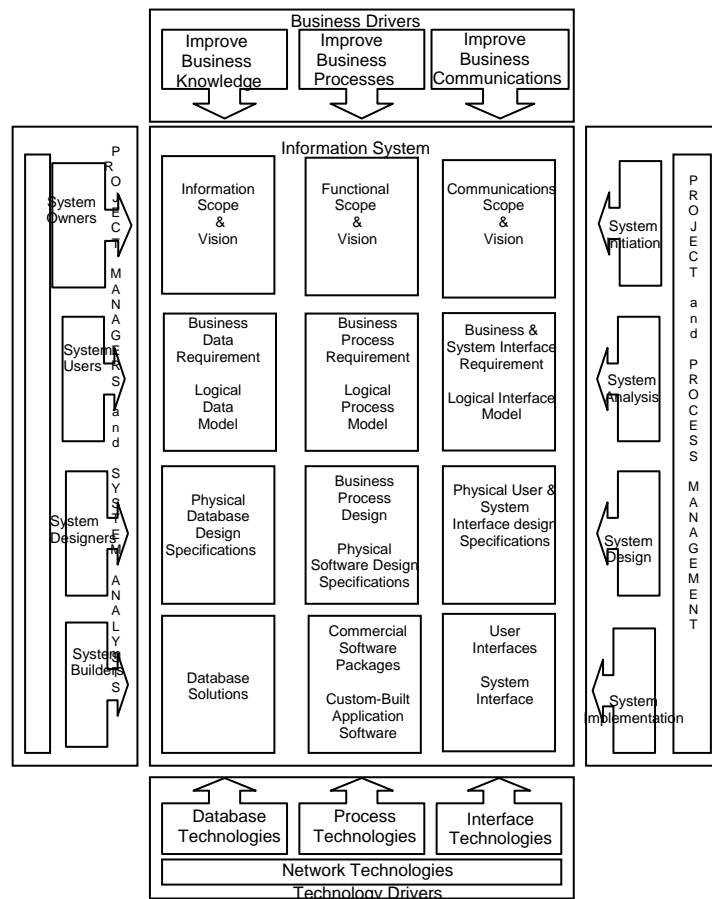


Fig. 4: Whitten’s framework [Whitten04]

Combining the concepts of Component-Based Development and ISO RM-ODP viewpoints.

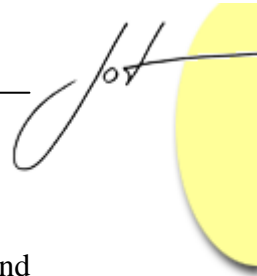
This section presents a summary of the framework proposed by [Stojanovic02]. This approach combines the paradigm of plug-and-play systems development, known as Component-Based Development (CBD) [Szyperski98] and the International Standardization Organization (ISO) Reference Model of Open Distributed Processing (RM-ODP) [ISO98]. In this way, a system specification based on components and a framework for ES development, are proposed. In the following sections an approach combining CBD and RM-ODP is considered.

Zoran's framework integrate the CBD and the RM-ODP concepts. It provides a specification involving not only structural and behavioral aspects of ES, but also the human and organizational aspects of the enterprise. The component-oriented viewpoints allow an integrated view of the system through its constituent parts. Viewpoints specifications evolve coordinately through time, having as result a system specification used as a base for the ES future (evolutionary) development. [Stojanovic01]. In order to handle changes, the separation of concerns paradigm has been introduced (recall, for example the first object-oriented languages architectural constructions, like MVC in Smalltalk). CBD and RM-ODP help to handle complex problems, facilitating the separation of concerns [Stojanovic02]. In consequence, a model-driven framework is proposed in Figure 5 for systems development integrating CBD and RM-ODP.

It consists of three consistent and related architectural models, where each model covers a particular system's aspect and together they provide a complete, business-driven, technology independent, system specification that can be easily translated to the particular implementation. The framework is based on different types of related components, such as the business system and the distribution environment. It allows traceability from business processes through the software final artifacts. The component notion is the unifying element through the different concerns of the system [Stojanovic02]. The models are:

1. "Enterprise Architecture Model" (EAM) specifies the behavior of the system in the business context.
2. "System Architecture Model" (SAM) defines the system's structure in terms of the configurations of the services offered by the system and their interactions. It provides the business services related with the business requirements. The "Computational viewpoint" is used to specify the functionality of an ODP system, with transparent distribution facility [Stojanovic01].
3. "Distribution Architecture Model" (DAM) specifies a distributed infrastructure in terms of its distribution services, location and communication in the n-tiers architecture.

The following is a framework proposed by Cummins for enterprise integration.



Cummins Framework

The strategic goals of most business enterprises are to reduce costs, improve quality, and respond quickly to business problems and opportunities. Enterprise Integration contributes to each of these objectives. Enterprise Integration is first and foremost a senior management to create a harmonious business environment supported by information systems. Cummins depicts enterprise integration from a management viewpoint (see table 4) [Cummins02].

Management strategies to enterprise integration must address the four different domains depicted in figure 6. Each of these domains represents different challenges.

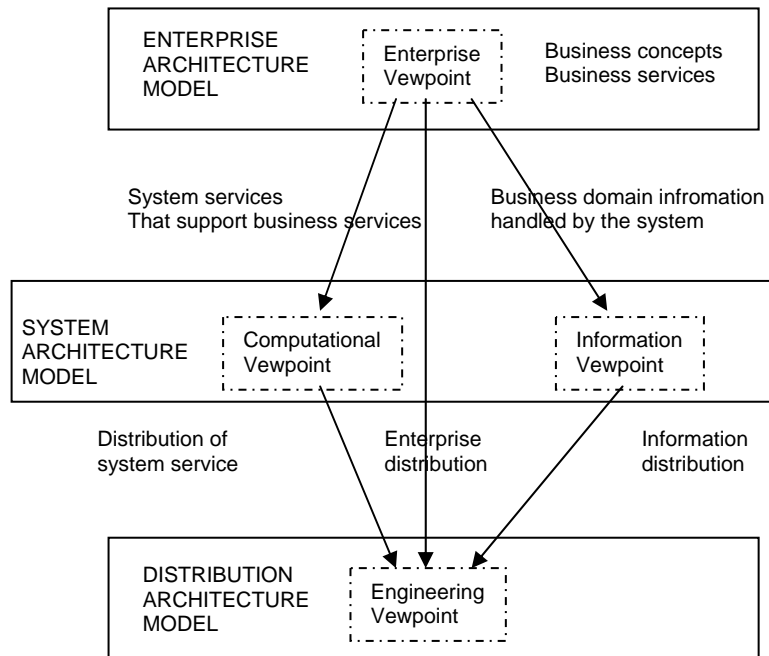


Fig. 5: Relation between the proposed models and the RM-ODP Viewpoints [Stojanovic02]

Management viewpoint	Brief description
Virtual Enterprise	It integrates customers and business partners just as electronic commerce
Corporate Domain	A corporation can incorporate a number of Business System Domains, just as the incorporation may incorporate a number of organizational units
Business System Domains	It is the set of business processes and applications that share components and maintain consistency among the state of those components, including the capability to roll back a computational transaction that cannot complete successfully and to recover to a consistent state in the event of a system failure
Business Processes	They may determine the order in which activities occur, the participation of users, and the actions to be taken to resolve exception
Business Applications	An application will be defined in this context as a unit of software that provides a set of closely related business function. Multiple applications may share a database, common services, and a Web server
Applications Component	Applications should evolve to a component structure, even though some components could be interfaces for large, complex, monolithic solutions

Table 4: Enterprise integration: management viewpoint

1. **Users:** An enterprise may have thousands of employees, suppliers and clients. The people of the enterprise must be organized, motivated, and led to achieve enterprise integration.
2. **Business processes:** The infrastructure and Internet technology will enable the cross-functional automation of business processes. Some of these business processes represent services that are available to other parts of the organization.
3. **Applications:** In the EAI implementations, many legacy applications will continue to serve the enterprise for years to come. In addition, many enterprise needs may have to deal with COTS applications. Applications must be developed to plug into the enterprise infrastructure and they must be complying with all infrastructure standards and applicable industry standards. Note that EAI middleware is called EAI and ES are called EAI implementations by Cummins.
4. **Infrastructure:** It is the complex of computers, networks, software, and associated services that support the operation and interconnection of many systems. It provides connectivity, an operating environment, and shared resources to reduce the burden and promote synergy between systems. Once the appropriate infrastructure exists, the cost, risk, and duration of efforts to implement new systems will be reduced significantly. However, when the infrastructure is not yet established, the cost of implementing initial applications will seem excessive, and the benefits of enterprise integration may be minimal (see figure 7).

Users
Business Processes
Applications
Infrastructure

Fig. 6: Enterprise Integration Management domains [Cummins02]

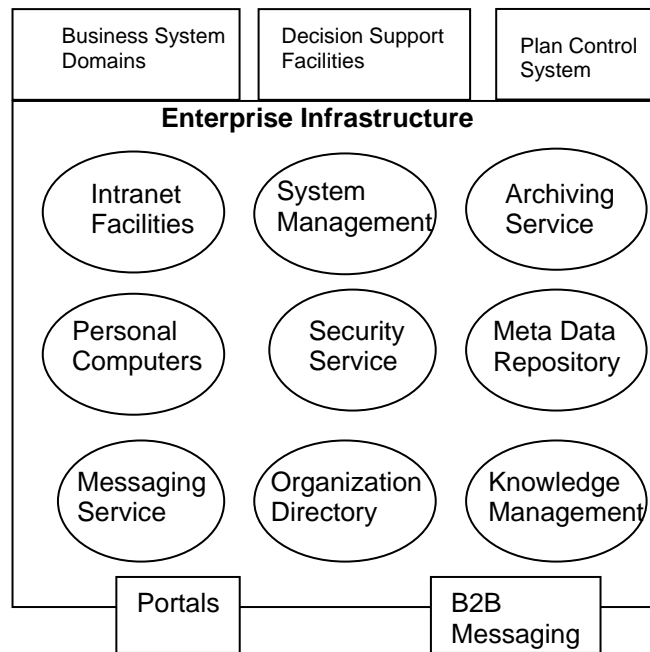


Fig. 7: The enterprise integration infrastructure [Cummins02]

Cummins defines the Enterprise Integration Architecture (EIA). It is based on current technology, and industry trends. Using industry standards minimizes the architecture's dependence on a particular vendor and enhances its flexibility. The goal of EIA is to manage business process with workflow management facilities so that the processes are visible and manageable [Cummins02].

The following general characteristics of EIA are Distributed computing, Component-based applications, Event-driven processes, Loose coupling of business functions, Decision support information, Workflow management, Internet access, Personalization of interfaces. These characteristics provide a perspective on key changes in the nature of systems, the relationships between systems, and the relationships between systems and users.

The following section presents the analysis of each of the framework discussed with respect to EAIF (see Section 3).

4 ANALYSIS OF ZACHMAN, WHITTEN, CUMMINS AND ZORAN'S FRAMEWORKS VS. EAIF

This section has the purpose of establishing sound comparison criteria to enhance strengths and weakness of known frameworks with respect to EAIF, in order to propose an improved version of the latter.

Zachman's framework vs. EAIF

Zachman's framework offers the following strong points

1. It can be used to analyze any business object or enterprise portion. It provides a holistic vision through the Generic and Enterprise specific frameworks.
2. It is not required to construct all the models defined by the framework according the problem domain.

Zachman's framework main aspects	EAIF
What	Processes
How	Services and Mechanisms
Where	Mechanisms
Who	Not present
When	Not present
Why	Processes

Table 5: Comparison between Zachman's aspects and the EAIF levels

Aspects *Who* and *When* are not considered in EAIF, according to Table 5. The absence of the *Who* aspect implies a weakness in EAIF, where people or human aspects are not considered. However, EAIF is a specific framework for EAI and Zachman's general framework presents a holistic view of the enterprise. However, Zachman's framework can be used in ES development.

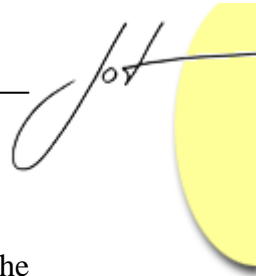
Whitten's framework vs. EAIF

Whitten's framework is specific for the development of different types of IS. Stakeholders and activities for software development play a central role.

Whitten's framework perspectives	EAIF
Business drivers columns	Processes
Technology drivers columns	Mechanisms (technology)
Left hand side rows (people involved)	Not present
Right hand side rows (development process)	Not present
Interior cells (activities related with the development of IS)	Services, Mechanisms (architecture)

Table 6: Comparison between Whitten's framework and EAIF

Table 6 shows that EAIF is not concerned with the software development process and the people involved, being at a higher abstraction level. With respect to the business and technology drivers, EAIF details the kind of processes and services, specifying also the kind of technology required to implement the services. Finally, with respect to the interior cells, EAIF is more concerned with the middleware architecture, detailing architectural and design patterns.



Zoran's framework vs. EAIF

Zoran's framework presents the following strength: the different aspects of the integration levels are identified according to the ISO-RM-ODP standard, allowing the construction of the system on the basis of its component elements. In particular, the distribution concept is considered in details. EAIF considers distribution at architectural (mechanisms) level, in the middleware patterns specification. EAIF is specified in UML, benefiting from a standard graphic notation language with an enriched semantics with respect to the description of Zoran's framework. However, Zoran states that UML and its extension mechanisms must be used to describe the RM-ODP viewpoints [Stojanovic01].

Services in EAIF are implemented combining different technologies and architectures. Zoran instead only considers the architectural view, since it is technology independent. These aspects should be discussed further if both frameworks were formulated as PIMs (Platform Independent Models), according to a MDA (Model Driven Architecture) approach.

The *Enterprise Architecture Model* corresponds to the EAIF *Process* level; the *System Architecture Model (Information Viewpoint)* corresponds to the EAIF *Services* level. Finally, the *System Architecture Model (Computacional Viewpoint)* and *Distribution Architecture Model* levels correspond to the Mechanisms level of EAIF.

Cummins's framework vs. EAIF

Cummins's framework concentrates on the EAI implementation aspects, detailing business management and infrastructure aspects. Cummins focuses on the notion of Enterprise Integration Architecture (EIA), to establish a set of characteristics that the enterprise must possess to perform software integration. This notion has not been considered in the other frameworks. The Infrastructure concept is similar to the Services and Technology levels of EAIF. For example, Intranet Facilities is part of the Technology level and System management is part of the Services level, in EAIF. However, the combination of architecture and technology aspects allows more precise guidelines for implementation, in EAIF. Moreover, the specification of the design patterns can be easily reused in the development of ES, since the whole framework is specified in UML.

The following section presents a summary of the comparisons.

5 FRAMEWORKS COMPARISON

This section presents table 7, where the rows are the criteria used for comparison. "Yes" or "No" indicate the presence or absence of the criteria. Otherwise, the element involved is explicitly mentioned. They are defined in what follows.

Comparison criteria

1. **Human aspects:** the stakeholders involved are considered.
2. **Standards:** the usage of standards is considered.

- a) **Specification:** the framework is specified as a standard or it uses a standard.
- b) **Use of:** the use of standards in the development of ES is suggested.
3. **Organizational/strategic aspects:** issues related with the business process of the enterprise are considered.
4. **Integration:**
 - a) **Specific:** the purpose of the framework is explicitly the modeling of integration.
 - b) **Model:** the theoretical base on which the framework is founded, such as: CASE approach, CBD-ODP, etc.
 - c) **Types:** different kinds of integration are modeled by the framework, for example: Backward, Forward and Upward integration.
 - d) **Levels:** hierarchical levels are defined, for example: Process, Services and Mechanisms.
5. **Type of services:** a classification of the services offered by the framework.
6. **Architecture:** the architecture is explicitly specified in a level the framework. For example, EAIF considers the architecture in the mechanisms level.
7. **Technology:** the technology is explicitly specified in a level of the framework. For example, EAIF considers the technology in the mechanisms level.

Summary of the Comparison

Criteria		Zachman	Whitten	Cummins	Zoran	EAIF
Human aspects		<ul style="list-style-type: none"> • Planner • Owner • Designer • Builder • Subcontractor 	<ul style="list-style-type: none"> • Project Manager • System Owner • System user • System analyst • System designer • System Builder 	Users level	Enterprise view	No
Standards	Specification	No	No	No	ISO-RM ODP	UML
	Use of	No	No	Technology standards (CORBA)	No	ISO 9126-1 to specify ES architectural quality
Organizational/strategic/Aspects		Yes	Business drivers	Process level	Enterprise view	Process level
Integration	Specific	No	No	Yes	Yes	Yes
	Model	No	No	No	<ul style="list-style-type: none"> • CBD • ISO RM-ODP 	Integrated CASE

	Types	No	No	<ul style="list-style-type: none"> • Internal business functions • Business functions performed outside the corporation 	No	<ul style="list-style-type: none"> • Backward • Forward • Upward
	Levels	No	No	<ul style="list-style-type: none"> • Users • Business Process • Application • Infrastructure 	<ul style="list-style-type: none"> • EAM • SAM • DAM 	<ul style="list-style-type: none"> • Process • Services • Mechanisms
Type of services		No	Functional specification /transformation	<ul style="list-style-type: none"> • Distributed, component-based • Legacy systems • COTS • Decision Support • Plant 	No	<ul style="list-style-type: none"> • Backward • Forward • Upward
Architecture		Aspects and Perspectives	Products Perspective	Application level and several infrastructure elements	Levels: SAM (computational viewpoint) and DAM (engineering viewpoint)	Mechanisms
Technology		<ul style="list-style-type: none"> • Column 3: Network, "where" • Row 4: Builder 	Technology drivers	<ul style="list-style-type: none"> • EIA elements • Infrastructure elements 	No	Mechanism

Table 7: Comparison of the five frameworks

To conclude this discussion, notice that Whitten’s framework is a specialization of Zachman’s. Both allow the development of different kinds of IS. With respect to the frameworks of Cummins, Zoran and EAIF, they are more concerned with the modeling of the integration of applications. In particular, Zoran is centered on distribution; Cummins is more devoted on the management and technological aspects of the integration and finally EAIF is focused on the architectural aspects, considering different integration viewpoints, but it takes no account of the human aspects of the integration process.

EAIF enrichment

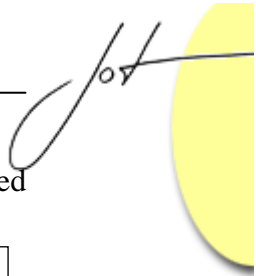
This section presents an improved version of EAIF in figure 8, as a result of the comparison presented in Table 7. Even if the IT definition in EAIF involves human aspects, this issue is not explicitly treated. In this sense it is the EAIF major deficiency. Since the organizational and strategic aspects of the business and the people involved was not considered in general. Moreover, the software development process itself was not included either. People related with the enterprise must be explicitly involved to achieve integration, at each of the established levels of integration: Process, Services and Mechanisms. Hence, a superior level has been added, including the People class,

considering all the persons involved in these levels. In each level a precise functionality is maintained. Changes in one level will not affect the inferior levels. The “drives” relation between the People and Process, People and Services and People and Mechanisms classes means that each person can drive at least one process in the Process level, one service in the Services level and one mechanism in the Mechanism level. In particular, the class “Software Development Process Model” has been added as a specialization of the Backward Process class, meaning that the enterprise can have its own software development process model. In consequence, the people involved are the “stakeholders” in the software process. Notice also that the added CASE tools class, specializes the Backward Service class, meaning that the software tools supporting the software development process model specified in the upper level are services offered by an ES.

6 CONCLUSION

The study of the Zachman's, Whitten's, Cummins' and Zoran's frameworks have facilitated the identification of strong and weak points of EAIF. Among the strong points, there is the introduction of the different integration views “backward, forward, upward”, for each integration levels. Another advantage is the standard UML specification used. This issue will facilitate the formulation of EAIF as a PIM (Platform Independent Model) of the MDA (Model Driven Architecture) approach [D'Souza01]. In order to achieve this goal, significant work needs still to be done to define each concept of EAIF using UML extensions mechanisms, such as stereotypes and constraints, in order to obtain a generic EAI standard PIM. The technological level should be revised in order to formulate the PSM (Platform Specific Model), since this model is independent form technological aspects. In consequence, only architectural aspects should be treated as mechanisms. The technology involved should be considered only for PSM. Moreover, the translation of this EAIF PIM to obtain an EAIF PSM needs also to be done, by developing models of platform specific component languages and defining rules for mapping from platform independent components to specific middleware standards such as CORBA and SOAP, for example. These mappings or transformation rules must be shown to be correct with respect to certain semantics preserving properties. These issues are ongoing research.

Among the weak points of EAIF, the lack of the “human aspects” concern was noticed. Most of the persons working in the enterprise are in general directly involved with the EAI process. This issue agrees with the tendency in software development processes, where the participation of the stakeholders is fundamental for achieving the desired project goals. In particular, the holistic view of the organization is crucial to implement an EAI solution. This aspect, found as a result of this study, has been easily added to EAIF, showing also its flexibility to extensions, adding an upper “People” level to the model, involving people in each inferior level. Moreover, another weak point was the absence of the software development process in the model, which has also been included as an enterprise backward process, with the corresponding tools supporting the process at services level. In this way the main ES activities are modeled by EAIF. Moreover, these extensions have confirmed the flexibility of EAIF. Finally, the criteria



provided for the comparison can be easily reused to compare other frameworks related with the ES development.

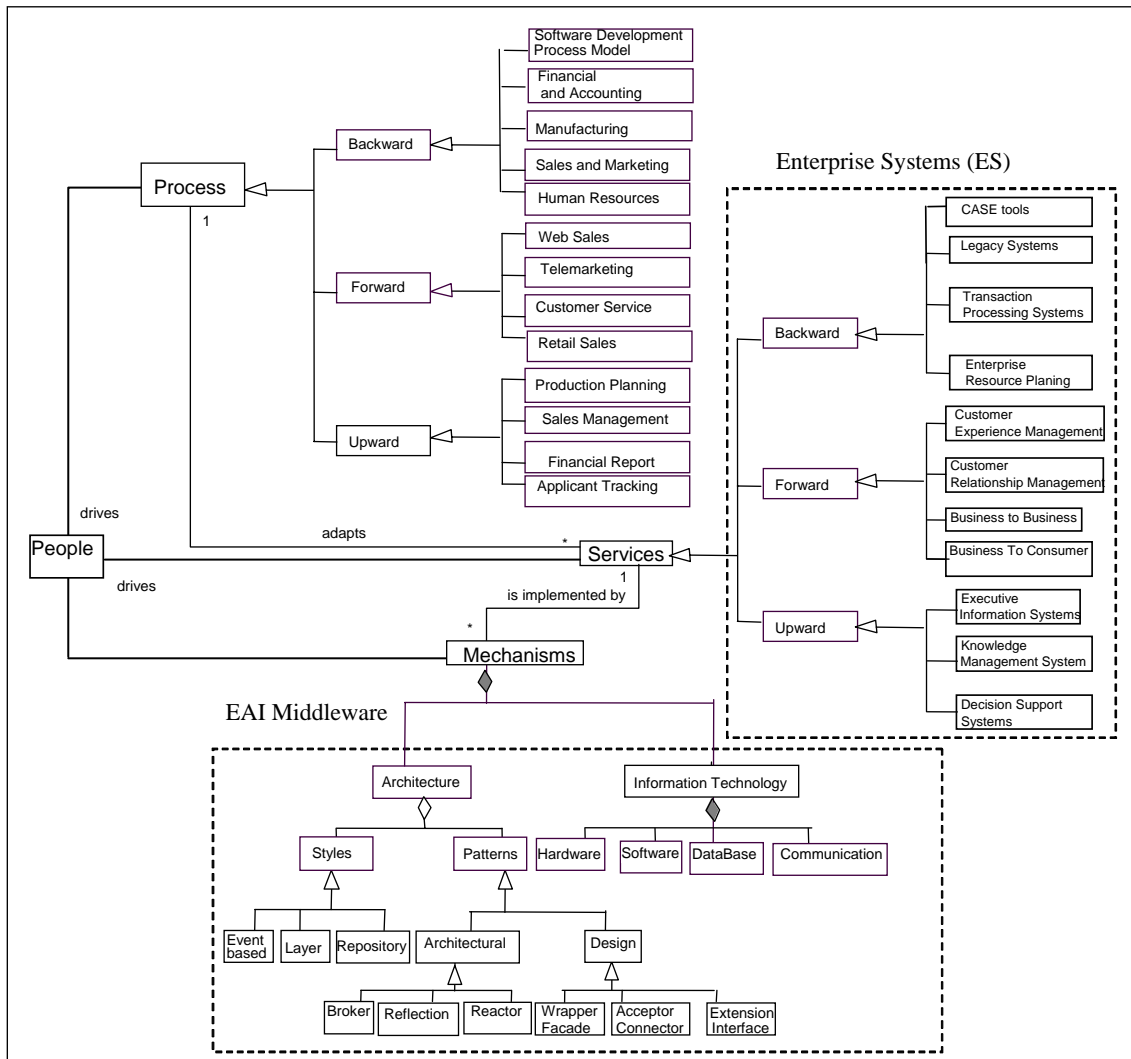
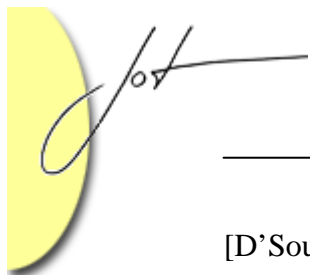


Fig. 8: Enriched EAI Framework

REFERENCES

- [Brown94] Brown A., Carnery D., Morris E., Smith D., Zarrella P.: *Principles of Case Tool Integration*, Software Engineering Institute, Oxford University Press, 1994.
- [Buschmann96] Buschmann F., Meunier R., Rohnert H., Peter S., Michael S.: *A System of Patterns*, John Wiley & Sons Ltd., 1996.
- [Cummins02] Cummins F.: *Enterprise Integration*, Wiley Computer Publishing, 2002.



- [D'Souza01] D'Souza, D.: "Model-Driven Architecture and Integration: Opportunities and Challenges", available at <http://www.catalysis.org/publications/papers/2001-mda-reqs-desmond-6.pdf>, 2001.
- [ISO98] International Standard Organization, Information Technology, "Basic Reference Model of Open Distributed Processing", ISO/IEC 10746-1, 1998.
- [Laudon04] Laudon K., Laudon J.: *Management Information Systems*, Eighth Edition, Prentice Hall, 2004.
- [Losavio02] Losavio F., Ortega D., Pérez M. "Modeling EAI", Proceedings of the XXII International Conference of the Chilean Computer Science Society (SCCC 2002), IEEE Computer Society Press, pp. 195-203, Copiapo, Atacama, Chile, 2002.
- [Losavio03] Losavio F., Ortega D., Pérez M. "Towards a Standard EAI Quality Terminology", Proceedings XXIII International Conference of the Chilean Computer Science Society (SCCC 2003), IEEE Computer Society Press, pp. 119-129, Chillán, BÍO-BÍO, Chile, 2003.
- [McKeen02] McKeen J., Smith H., "New Developments in Practice II: Enterprise Application Integration", Communications of the Association for Information Systems, Vol. 8, pp. 451-466, 2002.
- [Mendoza96] Mendoza M.: *Sistemas y Procedimientos. Nuevas Tendencias en la Contaduría*, Universidad Católica Andrés Bello, Caracas, Venezuela, 1996.
- [O'Brien03] O'Brien, J.: *Introduction to Information Systems. Essentials for the e-Business Enterprise*, Eleventh Edition, McGraw-Hill, pp. 217-218, 2003.
- [O'Brien04] O'Brien, J.: *Management Information Systems*, Sixth Edition, McGraw-Hill, 2004.
- [O'Rourke03] O'Rourke C., Fishman N., Selkow W.: *Enterprise Architecture Using the Zachman Framework*, Thomson Course Technology, 2003.
- [Pérez99] Pérez M.: "Arquitectura para Ambientes Case Integrados", Tesis Doctoral en Ciencias de la Computación de la Universidad Central de Venezuela, Caracas, Venezuela, Febrero, 1999.
- [Ramdane-Cherif02] Ramdane-Cherif A, Lévy N., Losavio F. "Dynamic Reconfigurable Software Architecture: analysis and evaluation", Proceedings of WICSA 2002, Montreal, Canada, August, 2002.
- [Sandoe01] Sandoe K., Corbitt G., Boykin R.: *Enterprise Integration*, California State University, Chico. John Wiley & Sons, Inc., 2001.



- [Schmidt01] Schmidt D., Stal M., Rhonert H., Buschmann F.: *Pattern-Oriented Software Architecture*, Vol. 2, John Wiley & Sons, Ltd., 2001.
- [Shaw96] Shaw M., Garlan D.: *Software Architecture - Perspectives on an Emerging Discipline*, Prentice Hall, 1996.
- [StojanovicZ01] Zoran Stojanovic, Ajantha Dahanayake¹, Henk Sol: "Integration of Component-Based Development Concepts and RM-ODP Viewpoints", 1st. International Workshop on Open Distribute Processing Enterprise, Computation, Knowledge, Engineering and Realisation, pp. 98-109, WOODPECKER 2001.
- [StojanovicZ02] Zoran Stojanovic, Ajantha Dahanayake: "Components and Viewpoints as Integrated Separations of Concerns in System Designing", Workshop on Aspect-Oriented Design (in conjunction with the 1st International Conference on Aspect-Oriented Software Development), Enschede, April 22-26, 2002.
- [Szyperski98] Szyperski C.: *Component Software: Beyond Object-Oriented Programming*, ACM Press, Addison Wesley, 1998.
- [Themistocleous01] Themistocleous M., Irani Z.: "Evaluating Application Integration: an Exploratory Case Study", Seventh Americas Conference on Information Systems, 2001.
- [Whitten04] Whitten J., Bentley L., Dittman K.: *Systems Analysis and Design Methods*, Sixth Edition, McGraw-Hill, 2004.
- [Zachman03] Zachman J.: this document is a response to the OMG BRWG RFI Version # 1b Copyright, 2003. Excerpted from The Zachman Framework for Enterprise Architecture: A Primer for Enterprise Engineering and Manufacturing. Available at <http://www.zachmaninternational.com>, 2003.

About the authors



Francisca LOSAVIO received the Doctor degree in Computer Science in 1991 and a 3ème. Cycle Doctor Degree in Computer Science in 1985, both from the Paris-Sud University, Paris XI, Orsay, France. She also obtained a MSc. degree in Computer Science in 1983 from the Simón Bolívar University, Venezuela. Titular Professor at the School of Computer Science, Faculty of Science, Venezuela Central University, Caracas, she works at the ISYS (Ingeniería de Software Y Sistemas) Research Center, where she coordinates the Software Technology Laboratory (LaTecS). She has participated in national and in European Community research projects. Her main research axes are software architecture, software quality and software development process. E-mail: flosav@cantv.net



Dinarle Ortega is a lecturer member of the Computer Science Department of the Faculty of Science and Technology, of the Carabobo University, Valencia, Venezuela. She is a PHD student in the Computer Science postgraduate program of the Venezuela Central University. Her research area is Software Engineering, in particular Software Integration and Software Architecture. E-mail: dinarleortega@yahoo.es



María Angélica Pérez de Ovalles is a member of the Association of Information Systems. Titular Professor at Simón Bolívar University. Ph.D. Computer Science (1999), Central de Venezuela University; MSc. Information Systems (1993), Católica Andrés Bello University; Electrical Engineering (1975), Metropolitana University. Publications on: ISACC'95, Mexico; AIS'96, USA; AIS'97, USA; AIS'98, USA; AMCIS '99, USA; CLEI '99, Uruguay; JOOP 12(6); AMCIS '00, USA; Journal Information & Software Technology, 2000; JOOP, 2000; SCI '00, USA; Revista de la Facultad de Ingeniería de la UCV 15(2); AMCIS '01, USA; JIISIC '01, Argentina. Research interest: Process improvement, Software Engineering, Methodologies, CASE Tools, Information Technology. Expertise areas: Information Systems, Methodologies and Software Engineering. E-mail: movalles@usb.ve