UDK 005.22
Preliminary communication

METHODOLOGY FRAMEWORK FOR PROCESS INTEGRATION AND SERVICE MANAGEMENT

[1]Darko Galinec, [2]Slavko Vidović
[1]*Republic of Croatia, Ministry of Defence,*
Information and Communications Technology Department, Bauerova 31/2, Zagreb
darko.galinec@morh.hr
[2]*University of Zagreb, Faculty of Organization and Informatics, Varaždin, Croatia*
slavko.vidovic@foi.hr

Abstract: *History of information systems development was driven by business system's functions automation and mergers and acquisitions - business subjects integration into a whole. Modern business requires business processes integration through their dynamics and thus enterprise application integration (EAI) as well. In this connection it is necessary to find ways and means of application integration and interaction in a consistent and reliable way. The real-time enterprise (RTE) monitors, captures and analyses root causes and overt events that are critical to its success the instant those events occur [6]. EAI is determined by business needs and business requirements. It must be based on business process repository and models, business integration methodology (BIM) and information flow as well. Decisions concerning technology must be in function of successful application integration. In this paper EAI methodological framework and technological concepts for its achievements are introduced.*

Keywords: *business process, EAI, methodological framework, RTE.*

1. INTRODUCTION

To integrate two business functions or two business systems it is necessary to connect their business processes with application support and data exchange. Processes appertaining to one application system create data which will be used by another application system.

First and the key reason for the integration of business systems' applications are user business needs for business processes and information flow, and changes in business processes occuring during business transactions.

The next integration reason is related to the differences in technologies by means of which applications are made. Integration should be carried out to connect technologically different applications. Because of process complexity which includes breakdown of the existing business processes and applications, business processes changes on the basis of business needs and user requirements, modeling of such processes and new applications and their connection it is necessary to shape methodological framework. The use of this framework should result in the successful completion of EAI projects.

1.1. EAI SHORT HISTORY

Applications have been developed on the basis of business system function architecture (for example automation of production, procurement, sales etc. functions, i.e. applications for their support. EAI related requirements also appear with business combinations - business subjects integration into a whole (fusion, consolidation, acquisition). In such cases information systems applications, of the subjects which are going to be merged, are integrated, From early sixties to the late seventies of the last century, business systems applications were simple in view of shape and functionality [5]. Business system data integration was not considered at all, the aim was to support manual procedures by PC. During eighties the need to integrate applications within business systems was recognized. There were attempts to reshape existing applications to make them suitable for integration. Since during nineties ERP applications prevailed, the existing applications and data were adjusted to ERP system. The aforesaid could be done only through EAI introduction as a logical sequence of events. Later on the advantages of integration of multiple business processes through existing applications have been conceived. Other factors which have contributed to the EAI development include growth of applications intended for supply chain management (SCM) support and business to business integration (B2B), applications for modern business processes support and web application integration. Modern business requires process centric approach, i.e. end-to-end (E2E) management and control of business system. The process includes various applications intended for the support of different business processes and functions.

1.2. REAL-TIME ENTERPRISE

Use of information is a key to the RTE to identify new opportunities, avoid mishaps and minimize delays in core business processes. The RTE will then exploit that information to progressively remove delays in the management and execution of its critical business processes. This is based on contention that there is always prior warning before every major favorable or unfavorable business surprise. Critical business processes are those of high value and importance for the business system, the improvement of which will significantly affect the business result [6].

Process models enable business analysts and system architects to work together to establish event-driven and service-oriented architectural styles in composite business applications. Business systems that use business process models will have more success in designing and implementing efficient and effective business processes. To accomplish this, business analysts should be well-versed in creating business models that explicitly identify events, and they should understand the concept of services and event handlers. System architects should be capable of interpreting business process models and establishing application architectures with service-oriented and event-driven styles that trace back to the requirements specified in those models [11, p. 1].

2. BUSINESS NEEDS AND USER REQUIREMENTS CONCERNING EAI

The users are key factors for EAI accomplishment. Their requirements, based on business needs and business processes which should be integrated, determine the purpose of EAI and are used to model it. Process approach as a result has models and business processes repository i.e. knowledge base on the entire organization [12, p. 2]. Transactions between business processes should be performed according to the security rules, based

solely on the real business needs. Experts developing EAI system should support transactions technologically, but they must not define them or decide on them.

2.1. OWNERSHIP OVER PROCESSES, RESPONSIBILITIES AND PROCESS STEPS

There is defined ownership over processes in a business system. One owner can manage one or more roles which include respective responsibilities. Business process includes several process steps. Within roles one or more process steps can be made, presented on workflow chart. Process application support should be provided for, from business needs and user requirements to the completion of the initiated process. Process steps which have been exercised within roles are not E2E, but workflow charts (relationship between E2E and workflow chart may be one-to-many, but one-to-one as well).

On the hierarchical level of business system, which is above the level on which business process owners are, there is the main owner of the process. He is the owner of all existing processes and workflow charts. In this connection, he can initiate the changes if some processes proceed in the inadequate way.

If the owner of the role completion process has at his disposal technologically identical applications there is no need for EAI since the applications are of the same sort. If there are different technologies (applications), EAI is necessary.

2.2. BUSINESS PROCESSES CHANGES BASED ON BUSINESS NEEDS

Business needs and user requirements should be modeled considering the impact on EAI. By introducing software requirements into the stream of analysis, applications can be reconfigured automatically as a result of matching business requirements to actual performance (Fig 1) [3, p. 4].

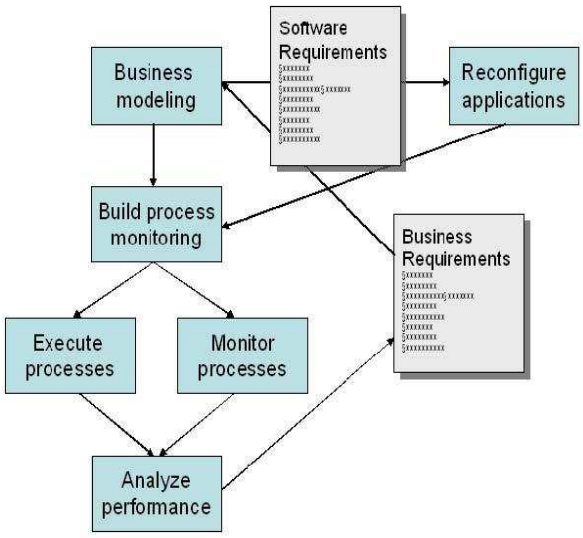

Figure 1. Applications reconfiguration based on business requirements

During process completion, data are collected and process execution is analyzed in view of which the requirements for business process changes are articulated. Business needs and user requirements change business processes models, new software requirements and needs concerning application changes turn up as well.

Because of business needs in modern application systems response time between requirement for business process change and corresponding application change must be extremely short, measurable in short time units (days for ex.). Main goal is to enable undisturbed business proceeding, respecting needs for process and application changes. Delays provoked either by process model changes (based on business needs) or by necessary technological interventions must be within defined period of time.

Modern business requires automatic process management i.e. applications to support it. To achieve the aforesaid it is necessary to configure applications according to the rules of contemporary architectures: service oriented (SOA) and events driven (EDA). On the technological level there are tools used to develop system for business process management (BPM); process modeling and workflow charts, business processes repository, simulation and detection of business bottle necks. On the basis of the results obtained the processes are reshaped and optimized. In this way optimized processes change the existing model and can be transformed into executive code of the process (which will manage business and application on the basis of business rules) or can be used for further modeling and development of new applications. Performance of the process is analyzed and tracked, reshaping will go on till the satisfying solution is reached.

To conclude, successful contemporary business systems require adequate interaction of all business processes and technology support. This can be obtained by means of BPM system, which in an integral way, uses applications conformed to business needs.

Since the applications are frequently technologically different, prior to BPM realization it is necessary to accomplish EAI.

3. EAI METHODOLOGICAL FRAMEWORK

By means of proceedings and application, EAI methodology must ensure consistency and reliability of an integration system. The example can be found in United States Department of Defense C4ISR document which has been used in the preparation of the document «NATO C3 System Architecture Framework (NAF)» for the architecture of command and control system with information system support. This document quotes that Defense Information Infrastructure Common Operating Environment encompasses architecture, standards, software reuse, shareable data, interoperability, and automated integration in a cohesive framework for systems development [1, p. 137].

According to Zachman framework, contextual and business model are independent of the computer platform (1st level), system model (logic) is also independent of the computer platform (2nd level). Technology model (physical) depends on computer platform (3rd level) and includes descriptions of models, architectures and descriptions used by technicians, engineers and contractors who design and create actual product. Because of the aforesaid, models, architectures, descriptions containing business system borders and the ways of its interactions with outside environment, and models with descriptions used by the individuals-the owners of business processes-in the focus of the authors' research.

Nowadays there are several EAI methodologies. One of them [2] mentions the following methodological steps:

1. Estimate of EAI necessity resulting in the respective Estimate Report
2. Strategic EAI planning and implementation, resulting in Strategic Report

3. Development of application and technical architecture, resulting in Detailed Implementation Plan
4. EAI implementation, monitored through Testing Reports and Quality Assurance Survey.

In this paper the authors present a new EAI methodology which is driven by business processes and is carried out in phases. EAI should be carried out in five methodological phases:

1. Business processes modeling and planning of EAI process needs according to business level (planning)
2. Analysis of communication and semantic requirements with transactions-requirements towards EAI
3. Providing for interoperability through three-level EAI model (design)
4. EAI development by means of adequate technology (construction)
5. EAI implementation and acceptance.

3.1. BUSINESS PROCESSES MODELING AND PLANNING OF EAI PROCES S NEEDS

The real business process has been monitored, application systems of a business system (if there are any) and its processes, and interaction processes with environment as well.

The way in which applications intercommunicate and real needs for their integration are defined.

Example: In the business part of the customer relationship management (CRM) application system there are applications which, on the basis of received input values, in the interaction chain, cooperate with applications coming from service management (SM) and resource management (RM), and in this way connection with supplier relationship management (SM) applications is established (Fig 2).

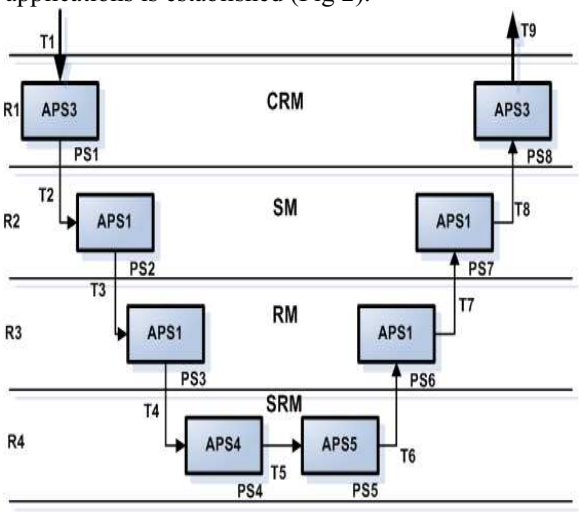

Figure 2. Defining the scope for integration of business system's applications

On the basis of user request the E2E process is initiated (transaction T1). On the CRM level within 1st role (R1) 1st process step with the support of the 3rd application system takes place (APS3).

On the SM level, within the next role (R2), PS2 is performed with the support of APS1 (application development technology differ from the one previously mentioned).

RM level includes R3 and makes PS3 (T3) possible with the support of APS1 (technology being the same as on the aforegoing level).

On the basis of occurences of the previous level, the performance of SRM level is initiated. Within this role T4 and T5 occur, PS4 and PS5 are performed with the support of APS4 and APS5 (different application development technologies when compared with the previous level and at on same level as well)

Transaction T6 (R3 repeated) refers to the communication between SRM and RM levels (PS6) and support of technologically different application, when compared with the previous one-APS1

Next role (R2 repeated) contains communication on the SM level, on the basis of T7, performance of PS7 (technology remains unchanged-APS1).

Eventually, on the CRM level, R1 is performed (PS8, T8) with the support of APS3 (technology differs from the previous one). The user is given response (APS3, T9) and the process E2E is completed.

EAI is necessary with the interactions of applications developed by different technologies.

3.2. ANALYSIS OF COMMUNICATION AND SEMANTIC REQUIREMENTS WITH TRANSACTIONS-REQUIREMENTS TOWARDS EAI

Based on the previous step, the kind of communication between applications will be determined with regard to the sort of data which interact.

Example:

There are certain applications whose integration will not be automated or will be delayed because of rare incidence and/or importance of data exchanged of business system (APS2). In such cases interaction and data exchange will be performed manually.

Only applications developed by different technologies are integrated.

APS3 interacts with the environment and data processing is carried out under the request/reply principle (Fig 3).

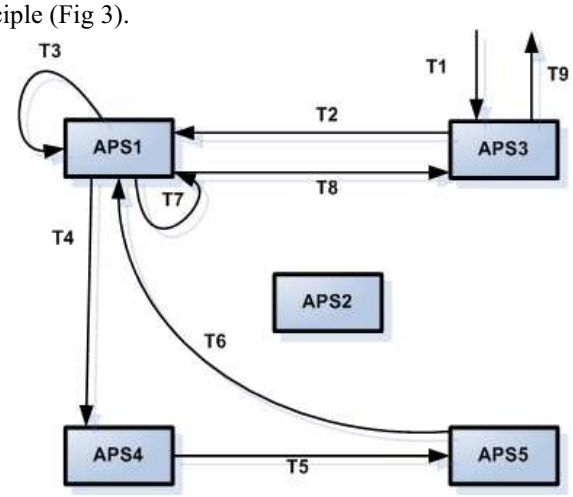

Figure 3. System and its environment – communication and semantic requirements

RTE architecture must accelerate and combine the efforts of many business units and their respective application systems running on different computers, sometimes in different enterprises. The characteristics of events, message-oriented middleware (MOM) and publish-and-subscribe communication patterns are well-suited for the "fast and furious," highly integrative nature of RTE strategies [8, p. 2].

The key difference between conventional business processes and a new generation of improved business processes is that the new processes are event-driven [9, p. 1].

3.3. PROVIDING INTEROPERABILITY THROUGH THREE-LEVEL EAI MODEL

Business system architecture is based on the interoperability standards. Interoperability is capacity of the information and communication systems and business systems to support data flow and provide exchange of information and knowledge [13, p. 9].

Interoperability framework includes norms, standards and recommendations which describe achieved or desired agreement of the interested parties with regard to the interconnection way. Interoperability framework is changeable document which must follow technological, normative and business changes. Interoperability levels are process, semantic and technical ones as shown in Table 1 [13, p. 9].

Table 1. Interoperability levels

Type	**Description**
Process	Relative to business goals, business process modeling and cooperation achievement between different units whose structure and work mode are not necessarily congruous. To fulfill system user needs and to reinstall available and simple user services it is necessary to establish process interoperability
Semantic	Relative to data meaning. Thanks to this level exchanged data have the same meaning at the starting point and destination, pieces of information originating from various information resources are linked in a meaningful way.
Technical	Relative to the norms and standards used for the interconnection of computer systems and services. It includes open interfaces, network and security services, middleware, integration presentation and data exchange.

So, it is necessary to define real EAI need and to examine it in view of process, semantic and technical interoperability.

Fig 4 shows transactions among technologically different applications- those for which EAI is necessary: APS1-APS3 (T8), APS1-APS4 (T4), APS3-APS1 (T2), APS4-APS5, APS4-APS5 (T5), APS5-APS1 (T6).

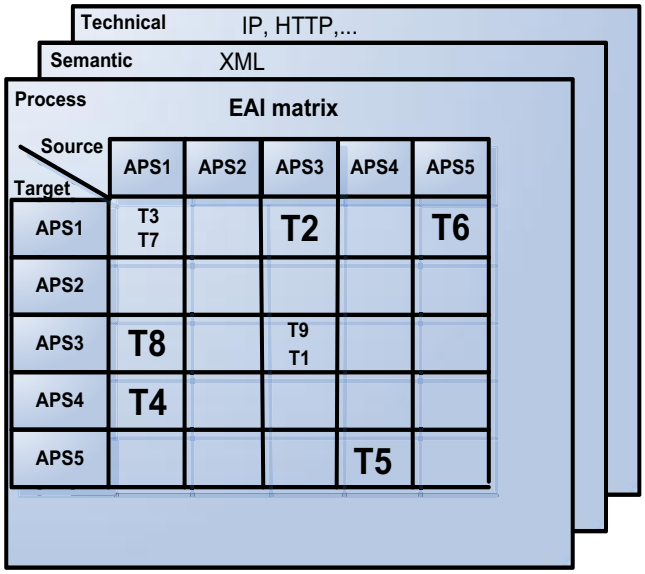

Figure 4. EAI matrix and interoperability

EDA and SOA are compatible but distinct concepts, each with its own advantages and limitations. Enterprises need both. EDA and SOA have many similarities. Both support distributed applications that go beyond conventional architectures, both use a modular design based on reusable business components, and both may be enabled through Web services. Architects and developers must understand the local business requirements and process models to determine whether SOA, EDA or some combination of them is right for each aspect of each new business process [10, p. 1].

Unlike SOA, EDA is the design vision for long-running asynchronous processes (SOA is best applied to real-time request/reply exchanges). In EDA, a process node posts an event (in SOA, a process node makes a targeted processing request). In EDA, posting an event reflects results of some past processing (in SOA, making a processing request directs future processing). In EDA, the poster of the event is disconnected from the processors of the event, if any (in SOA, the requestor of service knows the service and depends on its existence and availability) [7, p. 3].

The nature of data retrieval transactions fits well with the model of SOA (make request for information, wait for the reply, disconnect on receipt of the reply). The nature of update transactions fits well with EDA (request the update, ensure delivery of the request, release the resources without waiting for the time-consuming process of applying the updates). The nature of composite transactions fits well with SOA (represents a complex real-time process as a single transaction, hide the complexity of composition and integration behind the wrapper interface). The nature of multi-step processing fits well with EDA (monitor status, trigger processing based on changes of status, evolve a process through its component steps as status changes from initiation to completion) [7, p. 3].

Conventional systems use pull-based, request/response patterns for most program-to-program and program-to-database communication. Each recipient program continuously loops, polling to see if a new event has arrived. If it polls infrequently, the business process runs slowly because the data is waiting for too long. If it polls often, the network and system overhead is unacceptable because lots of needless requests may be made before an event appears.

Event-driven systems send each update individually as soon as it is recognized by the sending application. Therefore, they use push communication in which the timing of data delivery is determined by the sender. The sender is the first to know that an event has occurred and can send it immediately so the data is not sitting idle until each recipient polls. Push communication is more scalable, because nothing is sent until the event is available. Push-driven systems accelerate a business process by minimizing the handoff time between activities.

4. ORGANIZATIONAL AND TECHNOLOGICAL CONCEPT

Enterprises that want to operate in real time must expand their use of event-oriented design, message-oriented middleware and publish-and-subscribe communication [8, p. 1].

A successful RTE will use the principles of event handling at the business and the technical level. Event-driven business processes are usually implemented best by event-driven software design and middleware technologies. Enterprises should expand the number of places where they:

- Handle events individually rather than in batches
- Use push-based rather than pull-based communication patterns
- Deliver information updates to multiple destinations simultaneously
- Design with an explicit focus on the idea of events
- Manage the process of event notification in a systematic fashion [8, p. 2].

Generic architecture of the information system (Fig 5) takes into account real business needs and participants in the business process of the complex system. Instead of integration layer, the concept of integration bus supported by network (ESB) is introduced. Applications, as a part of technology, are available for those users who are permitted to use them. Such information-centric and network-centric approach implies that applications given to the users in order to enable the data, information and knowledge flow depend on network as the only application “container”.

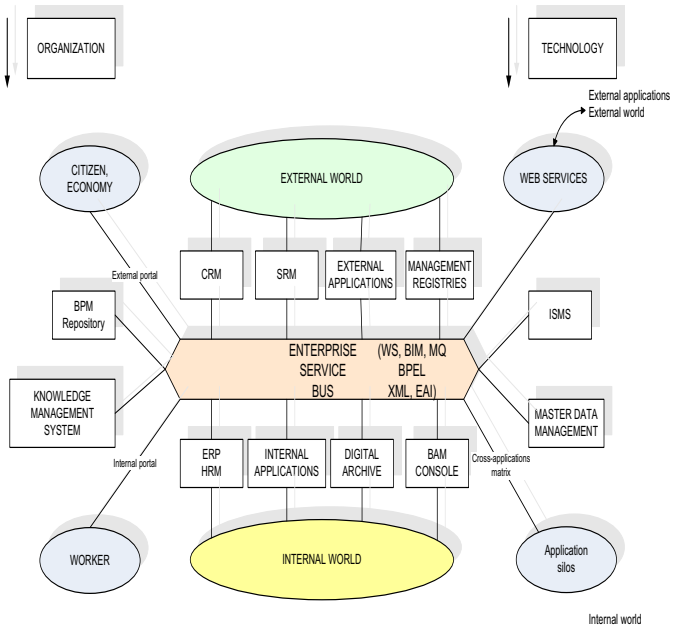

Figure 5. Generic architecture of information system

4.1. EAI DEVELOPMENT BY MEANS OF ADEQUATE TECHNOLOGY (CONSTRUCTION)

Business systems can use various commercially available EAI middleware products. They select their products based on composition of their enterprise, their plans and buying practices. The basic architecture of business systems' mission and business application software is evolving rapidly as developers apply modern design concepts such as SOA, EDA and intermediary-based application integration. The new generation applications and commercially developed business applications is more flexible, powerful and easier to maintain than traditional application software because of its modularity and layered structure.

The change in application architecture is made possible by advances in middleware technology and related standards such as XML and Web services. Significant innovation is appearing in virtually every middleware. Platform middleware products, such as application servers and application platform suites, were once thought to be approaching a steady state based on .NET and Java 2 Platform, Enterprise Edition (J2EE).

However, new platform technology using micro-kernels and message-based design is emerging to create new styles of platform. Integration middleware is also evolving as vendors and user companies try new approaches to application integration. Integration process uses middleware intermediaries that apply transformation, business process management and rule-based routing to application-to-application communication.

Integration middleware products such as Enterprise service buses (ESBs), integration suites and programmatic integration servers leverage XML message formats and Web services communication standards (for example SOAP and WSDL) to reduce the complexity of developing integration links among applications. However, traditional middleware products are a poor match for the service-oriented and event-driven applications being designed nowadays. Business systems, enterprise architects and project

leaders must be aware of the innovation that is occurring in middleware to be able to develop an orderly plan for leveraging the new technologies as suitable for their environments. The open-source movement is already having an effect on the J2EE application platform market, and is expected to affect decisions for other platform middleware and ESBs.

4.2. IMPLEMENTATION AND ACCEPTANCE

The occurrence of an event (a notable thing that happens inside or outside the business) can trigger the invocation of one or many services. Those services may perform simple functions, or entire business processes. This interaction between events and services is commonly referred to as event-driven SOA (EDSOA). A service may generate an event. The event may signify a problem or impending problem, an opportunity, a threshold, or a deviation. Upon generation, the event is immediately disseminated to all interested participants, which evaluate the event and take action. The event-driven action may include the invocation of a service, the triggering of a business process or further information publication. In this interaction, the service is purely one of many event sources in a broader event-driven architecture. A broader event-driven architecture stretches beyond event-driven SOA, to include real-time information flow and analysis, and complex event processing (CEP).

E2E processing must be adequate, technically feasible, operatively acceptable and usable through integrated application system.

Modern business systems are designed to perform a variety of functions; for most applications MOM is prevalent. Communication middleware helps programs talk to other programs. It is software that supports a protocol for transmitting messages or data between two points as well as a system programming interface (SPI) to invoke the communication service. MOM also provides for the safe (for example using strong security and reliable, guaranteed once and only once) delivery of messages. Protocols and SPIs used in communication middleware can be proprietary (for example IBM WebSphere MQ or Microsoft MSMQ) or based on industry standards such as ASN.1, Distributed Computing Environment (DCE) remote procedure call (RPC), CORBA/IIOP, Java Message Service (JMS) or Web services (based on Simple Object Access Protocol [SOAP]). Today, business systems are focusing on enterprise architectures based on SOA, Service-oriented development of applications (SODA), service-oriented business application (SOBA) and Web services.

EAI project is considered successful only if it can be proved that the new and the inherited system can work together prior to their application and usage by end users.

5. CONCLUSION

Business needs and goals which must be reached by a business system are the prime movers of the changes in application systems and information technology (IT). Application development is based on the function approach and architecture and it represents the beginning of EAI. High efficiency concerning performance of particular processes is not sufficient if they are not adequately integrated in E2E process. Today business is managed through E2E process approach and the process itself is supported by different applications originating from the function architecture of a business system. EAI is an arranged infrastructure intended for applications linkage (resources, technologies and interfaces) and transactions completion among application systems.

Interoperability (process, semantic and technical) is a processing relation between two business subjects rendering business transactions possible.

The highest business need level is RTE: an enterprise which in real time monitors environment and reacts on the incentive i.e. perform business transactions.

The purpose of EAI is to provide interoperability which will render possible inside and outside business system transactions so that eventually business system would function as RTE.

Since EAI infrastructure connects applications used to support business processes, EAI is of great importance and influence in business process management. Applications used to support business processes connected by sequence of business logic (by means of BPM system and on the basis of business rules) render possible E2E process, since they are used for consistent integration of processes constituent to E2E process approach.

Only on the basis of effective performance of E2E process, performance of individual process steps (shown on the diagram within roles) can be considered successful. By integration of applications intended for the support of particular process steps, E2E process can be supported as well. When united process steps give expected results, business system process functioning is satisfying.

The architecture of a modern application system (which supports real needs) will be based on business components as software components supporting particular business entities.

Such an integral architecture consists of two mutually complementing and interacting architectures – SOA and EDA.

Technological concept which supports them is an ESB. It includes numerous integration and communication technologies, among other things MOM middleware and Web services.

Successful application integration is the one which will provide for the quick flow of all transactions important for business and realization of imposed goals, serving all business processes which may need them.

IT, if used in an adequate way, is in the integration function and delivers new value to the business system.

To develop EAI system, new methodological framework is shaped based on the business system processes models, and is implemented in phases. Requirements concerning business transactions should be identified and analyzed, business processes should be modeled as well. These will result in models and repository of business processes as a reflex of a real business and the origin of business processes automation. Based on the methodological framework developed, integration of application systems is performed by InfoDom - Zagreb and then successfully implemented through several business projects.

REFERENCES

[1] C4ISR Architecture Working Group (AWG) (1997): C4ISR (Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance) Architecture Framework Version 2.0, Department of Defense, USA

[2] Cognizant Technology Solutions (2005): EAI Methodology, http://www.cognizant.com/services/advancedsolutions/eai/eai_method.htm, New Jersey, USA

[3] Ericsson M. (2004): Business modeling practices: Using the IBMRational Unified Process, IBM WebSphere BusinessIntegration Modeler, and IBM Rational Rose/XDE, IBM Corporation, USA

[4] Frankel D., Harmon P., Mukerji J., Odell J., Owen M., Rivitt P., Rosen M., Soley R. M. (2003), The Zachman Framework and the OMG's Model Driven Architecture, Business Process Trends, USA

[5] Gormly L. (2002): EAI Knowledge Base: EAI Methodology, http://eai.ittoolbox.com/documents/document.asp?i=1246, Information Technology Toolbox, Inc., Scottsdale, USA

[6] McGee K., (2004): Gartner Updates Its Definition of Real-Time Enterprise, DF-22-2973, Gartner, Inc.

[7] Natis Y. V., (2003): Service-Oriented Architecture Scenario, AV-19-6751, Gartner, Inc

[8] Schulte R. W. (2002): A Real-Time Enterprise is Event-Driven, T-18-2037, Gartner, Inc.

[9] Schulte R. W., (2003): The Business Value of Event-Driven Processes, SPA-20-2610, Gartner, Inc.

[10] Schulte R. W., Natis Y. V., (2003): Event-Driven Architecture Complements SOA, DF-20-1154, Gartner, Inc.

[11] Thompson J. (2004): Process Modeling Improves Business Application Architecture, G00123625, Gartner, Inc., USA

[12] Vidović S., Galinec D. (2003): The KMS Architecture Design for Service Providing, Proceedings of the 14th International Conference on Information and Intelligent Systems - IIS 2003, Faculty of Organization and Informatics, Varaždin

[13] Working Group on IT Architecture within the Coordinating Information Committee (2003): White Paper on Enterprise Architecture, http://www.oio.dk/arkitektur/eng, Ministry of Science, Technology and Innovation, Copenhagen, Denmark

Received: 12 March 2007

Accepted: 28 October 2007