

AZ-38
2002



ICSSSE 2002

15th International Conference / 15^{es} Journées Internationales

SOFTWARE & SYSTEMS ENGINEERING
and their APPLICATIONS



GÉNIE LOGICIEL & INGÉNIERIE DE SYSTÈMES
et leurs APPLICATIONS

December 3-5, 2002 — 3-5 décembre 2002

Centre pour la Maîtrise des Systèmes et du Logiciel
Conservatoire National des Arts et Métiers

Paris, France

Proceedings / Actes

Volume 3

Sessions 11-13, Journées du CMSL & Index

EAI: Concepts and Trends

Nicola Aloia¹, Cesare Concordia¹, and Grégory Weinbach²

¹CNUCE – CNR Via G. Moruzzi, 1
56100 Pisa (Italy

Phone: +39 0503152970, fax: +390503138091

²Objet Direct 38 Rue Copernic
75116 Paris
France

phone : +33 01 40 67 69 09 fax : +33 01 40 67 69 08

E-mail: cesare.concordia@cnuce.cnr.it

Abstract: The ever-increasing growth and penetration of Application Integration technologies and market is bringing about significant changes in how Enterprise Information Systems are organized and managed. Over the years information systems have evolved, today's enterprises (and in general organizations) want to use all information collected to add value to their products and services. Traditional information systems don't explicitly accomplish this task, they are used for facilitating the realization, communication and processing of the information, and, often, are constrained by the organizational structure, as well as the hardware and software architectures adopted. To respond to new needs, more flexible systems are needed. The computer systems cannot be monolithic; the different components should interact in a clever way to accomplish every function needed to reach the requested "equi-finality". Moreover, the increasing reliability of tools and techniques stemming from the "Internet revolution" is rendering ever more convenient to base the development of new applications on this technology, especially because of the great advantages it affords in terms of scalability and distribution. These two principles lead to the development of computer systems highly scalable and whose different components should be integrated allowing information exchanges between them. The main technique adopted to achieve this aim is the so-called Enterprise Application Integration (EAI). The present paper deals with the fundamental characteristics and current trends of Application Integration and related technologies and illustrates the problems that need to be tackled in development and management of an integrated and flexible enterprise information system.

Keywords: enterprise application integration (eai), information systems, web-based information systems

pattern is to create "meta-applications", whose capabilities are a composite of encapsulated services and integration logic. The services themselves may be implemented with independent components, wrapped applications, or even combinations of several applications and/or components.

- **Autonomous distributed:** the autonomous distributed integration pattern is characterized by the combination of application services to produce a composite behavior, as in the composite application pattern, but it goes a step further by exploiting frameworks for service discovery and dynamic method binding, enabling integration to occur at runtime (e.g. web services).

Currently data consistency is yielding to multi-step as the most widely used integration pattern, but probably there will always be problems for which data consistency is still the best approach. And despite the current hype around service-oriented architectures and web services, the composite application and autonomous distributed patterns are not replacements for the data consistency and multi-step approaches.

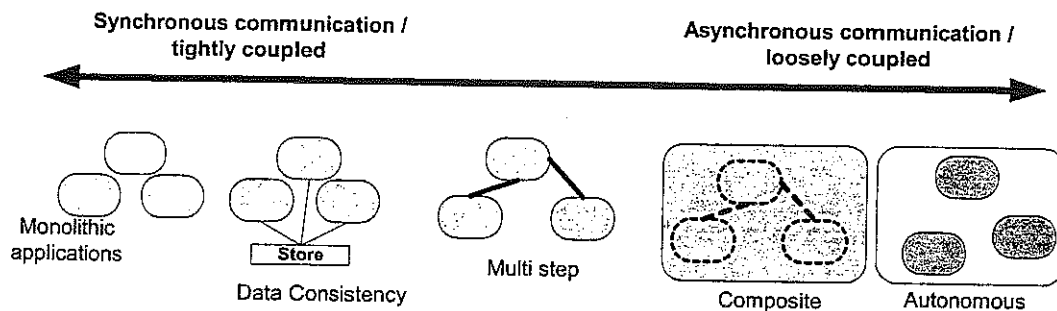


Figure 1 Integration patterns

INTEGRATING APPLICATIONS

There are two main views on integration: from inside the enterprise (A2A integration or Enterprise Application Integration (EAI)), and view from the outside (B2B integration (B2Bi)). For most organizations the main issue is to integrate consistently intra-organization and inter-organization aspects of their activities. In both views (Fig. 2): integration end-points are applications, services and data; end-points are accessed either directly or via middleware using resource-specific adapters. The core of both integration views are the services that determine how native interfaces combine resources, inputs and integration logic to produce the desired behavior for a given step: these are: routing, transformation, messaging and transaction services. How core services are implemented strongly depends on the integration patterns adopted. Most organizations have integration problems that fit more than one pattern, including some with hybrid requirements; there is an increasing need for integration solutions that offer a complete functional "stack".

INTEGRATION TECHNOLOGY: STATE OF THE ART

As previously stated, data consistency and multi-step integration patterns are currently dominating integration practice. The "messaging capabilities" are now a commodity; products like WebSphere MQ (formerly MQ Series) and Java Message Service (JMS) are today considered as de facto standards. Therefore innovation is pursued elsewhere, application integration solution providers are more and more offering tools whose aim is to merge B2B integration with A2A integration. As an example, products like WebSphere Process Manager (IBM) or TIB/InConcert (TIBCO) that allow users to define and manage business processes and workflow are completely integrated into IBM's and TIBCO's EAI suite offers. That means intra-enterprise and extra-enterprise integration strategy could be implemented using a unique approach, therefore avoiding duplication of adapter, transformation, management, and other elements. Another element that strongly influenced integration's evolution is the Internet. The opportunity to use this environment for implementing information systems offers considerable benefits, while at the same time posing new problems. Internet-based, hosted integration services are emerging as a cost-effective way for small and medium companies to participate in integration. Smaller players can participate in value chains with much larger partners, for instance iPlanet (Sun) or WebLogic Integration (BEA) offer many facilities to implement an Internet scale application integration approach, but also other solution providers furnish native software modules to include Internet advantages to their products.

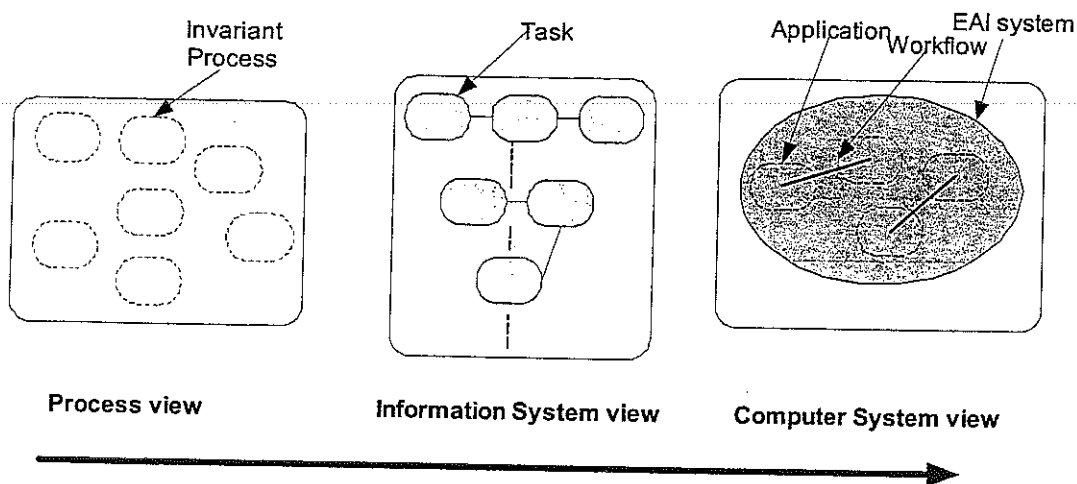


Figure 3 Strategies

Separating IS specifications and system implementation enables integration deliverables reusability. This principle allows reducing the effort of building and maintaining integration solutions in multiple runtime environments. The quality (completeness and non ambiguity) of the IS process modeling, combined with the use of integration tools (such as EAI) might allow automatic generation of some part of the computer system view (i.e. transformation, routing and workflow information), at the same time providing a high level of maintainability and global coherence.

There are some principles that drive each phase of view's definition, choosing an integration solution:

- To maintain environment independence: customers should be able to change completely or partly the information systems choices they made to enhance IS to organization evolving requirements. Open technical solutions allowing use of de facto standard messaging services, Internet protocols, multi platform programming languages (e.g. Java) and standard storage interface mechanisms (DBMSs and file systems) are preferred for achieving this goal. This principle ensures continued system viability in the face of change and enables maximum advantage to be realized from existing IT infrastructures.
- To manage functionality through abstraction: abstraction is used to allow users to deal with different applications and resources at the functional level. Dealing with resources through abstraction simplifies maintenance and makes it easy to specify systems that can be implemented in different environments with minimum changes.
- To model the system completely and deploy from the model (Model-Driven Integration): systems should be specified and maintained using diagrams, abstractions, and rules [7]. Building solution conformed to this principle simplifies system configuration, maintenance and management.
- To use a single set of tools and services to support EAI and B2Bi: in theory a common set of runtime services and design-time tools can be used to solve several integration problems. The goal is to build an infrastructure that supports both EAI and B2Bi needs. This principle can be implemented only if services are combined with the right set of abstractions and architectural layering.

Integration trends

Both integration technology and the integration market are evolving rapidly, therefore in this paragraph we try to do a snapshot of current trends.

Today integration is *pervasive*. As commercial offerings mature and organizations gain experience with start up projects, integration technology is being widely applied. All the patterns defined above are present in the current integration scene, reflecting the increasing scope and impact of the integration practices. Integration features are being embedded in application server platforms, messaging systems, portal servers, and other infrastructure technologies, as well as in commercial application products. However most of these offerings provide limited integration services, offering little value beyond their intended problem spaces. Organizations in specific industry sectors have particular integration needs that are not supported by most off-the-shelf integration solutions. Purpose-built solutions may address specific needs, but they offer limited support for broader enterprise integration requirements. In larger organizations, this can lead to a multi-vendor integration problem.

REFERENCES

- [1] Gosh, G.: Data Imperative: patterns in EAI behavior; eAI Journal, September 2001 pp22-28
- [2] <http://www.ibm.com/developerworks/patterns/>
- [3] Jean, G.: Urbanisation du business et des SI; Hermes 2000
- [4] Laudon, Kenneth C. and Jane P.: Management Information Systems; 4th ed., Prentice Hall, 1996
- [5] Linticum, David S.: Making EAI Scale; <http://www.intelligenteai.com/feature/010416/linticum.shtml>
- [6] Schoderbek, P.: Management Systems; Wiley 1977
- [7] The Model Driven Architecture (MDA) is the new strategy of the OMG:
http://www.omg.org/mda/executive_overview.htm
- [8] William Rue has established this taxonomy. Other taxonomy can be found in literature, see: Patterns for e-business: A Strategy for Reuse; IBM Press
- [9] Yee, A: Making EAI Work in the Real World: EAI Patterns;
http://eai.ebizq.net/enterprise_integration/yee_3.html