

INFORMATION SYSTEM INTEGRATION

For information systems, it is increasingly difficult to draw a line around an application system and say that you own and control it. For example, as value chains extend beyond enterprises, supplier and customer systems become part of each other's information architectures. Furthermore, in many application areas, data is distributed over a multitude of heterogeneous, often autonomous information systems, and an exchange of data among them is not easy. Figure 1 illustrates such a vertical fragmentation of organizational units. Each unit may be structured within three architectural layers, as described in the following.

The *business architecture* layer defines the organizational structure and the workflows for business rules and processes. It is a conceptual level expressed in terms meaningful to actual users of application systems.

The *application architecture* layer defines the actual implementation of the business concepts in terms of enterprise applications. At this layer, it is the central goal to provide the "glue" between the application domain described in the business architecture and the technical solutions described in the technology architecture. Research in information systems aims at filling the gap between business and

technology, which requires interdisciplinary cooperation between the application domain and information technology.

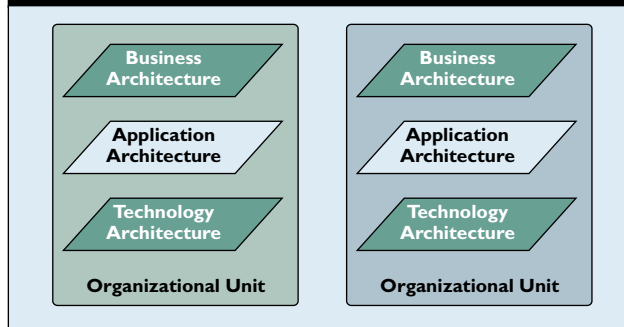
The *technology architecture* layer defines the information and communication infrastructure. At this layer, IT is challenged to achieve the business requirements.

It is important to realize that Figure 1 does not adequately reflect the reality. In practice, the business architectures of the individual organizational units cannot be treated in isolation: the business processes of cooperating units are highly interrelated and should be handled as such. Figure 2 illustrates this situation. Certain kinds of interactions among computer systems resemble interactions among people; thus, it is important to consider all levels when integrating those systems. A horizontal integration of the layers is required to support the business processes effectively, as indicated here.

Interorganizational processes. At this layer, business engineering [1] seeks to organize a commercial undertaking in a competitive way, whereby business processes cut horizontally through the traditional organization structure. Business process reengineering aims at continuously improving those processes.

To support the intraorganizational business processes within organizations effectively, the existing

Figure 1. Vertical fragmentation of organization units.



information systems must be integrated. This is already a nontrivial task, particularly if heterogeneous information systems exist (legacy systems). To allow, for instance, for electronic supply chains, interorganizational processes have to be supported, whereby the involved information systems are highly autonomous, making the integration process an even more challenging task.

Enterprise application integration. The goal is to integrate independent enterprise resource planning (ERP) systems at this layer. This is usually achieved by means of some kind of messaging services. Even the SAP R/3 approach, which aims at enterprise integration via one single database (no borders between enterprise units), acknowledges the fact that messaging services are required for integrating autonomous ERP systems, both within and across enterprises [4]. TSI Software's Mercator product (www.mercator.com), for instance, specializes in pre-built application adapters, data transformations, and messaging services among the ERP systems SAP R/3 and PeopleSoft.

The deployment of ERP systems often requires reengineering the business processes to align with the ERP system. However, it is usually unacceptable to require the business to change to the applications' functionality; instead the information architecture should align with the business organization. SI and componentization aim at supporting the business processes, while preserving the investments in (legacy) systems.

IT Disciplines involved in Information SI

For participatory development involving end users, applications should be built incrementally using techniques such as Rapid Application Development (RAD), Joint Application Design (JAD), and Prototyping. This special section, however, primarily concentrates on the interdisciplinary nature of SI among the different disciplines in IT, and discusses end-user involvement merely as a side issue. The articles in this special section are organized according to specific application domains. A variety of IT disciplines are studying problems and solutions for information SI:

Parallel and distributed systems. Research in operating systems, computer networks, and parallel programming systems concentrates, to a great extent, on managing the coexistence and coordination of multiple concurrent activities. In these areas, communication among system components and their synchronization are common problems to be solved.

With parallel programming, for instance, parallelism is used as a coordination mechanism and, accordingly, programming is divided into two separate activities: a sequential language that can be used to build single-threaded computations; whereas a coordination language is used to coordinate the activity of those computations. Thus parallel programming consists of putting together components and letting them cooperate.

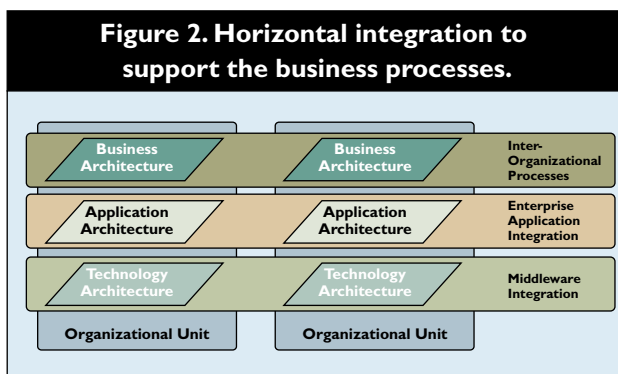
When it comes to SI, coordination is required for managing shared resources and dependencies among activities in and across systems; it is obvious we can learn from the trade-offs between different disciplines involved in SI. Malone and Crowston [2] already emphasized that the study of coordination is relevant for such dissimilar disciplines as computer science, linguistics, and operations research. Resource allocation, for instance, is widely studied in economics, organization theory, as well as in IT.

Database systems. Problems of coupling and integrating heterogeneous database and information systems have been addressed in the database area for some time. While research on parallel and distributed systems emphasizes the integration of computational components, research on database systems is more concerned with the integration of data. Federated database systems, for instance, approach the integration of heterogeneous databases by means of schema integration. However, when it comes to managing transactions over multiple local systems, for instance for executing transactional workflows, the problems to be solved are often very similar to those that arise in parallel and distributed systems. These similarities sometimes cause the reinvention of techniques that have already been elaborated in other disciplines.

Software engineering. When it comes to SI, we deal with complex systems of systems. Software engineer-

Applications need to understand the data provided by other applications; for instance, a common understanding is required of what a person's bank account is. Standardization of message formats and message content plays an important role in this context. Meanwhile, XML (www.w3.org) is emerging as the standard for defining the syntax of data structures to be transferred over the Internet. In order to provide interoperability across implementations, the concrete syntax and the semantics of standardized messages must be defined. Traditional EDI (Electronic Data Interchange) is often being reexamined to define the meaning of the transferred data, and XML is employed as the practical foundation used to structure this information.

Middleware integration. At this layer, the techniques for building componentized information systems with state-of-the-art infrastructures such as CORBA, database gateways, and transaction monitors, are employed. Middleware integration addresses the syntactical level (“plumbing” and “wiring”) while



Enterprise Application Integration also addresses a semantic level.

The borderline between Enterprise Application and middleware integration cannot always be pinpointed precisely. For instance, the Object Management Architecture of the OMG defines the Object Request Broker, which can be deployed for middleware integration, and also high-level services (such as busi-

ing is concerned with the systematic development of such complex systems. Work in this area deals with questions of adequate software architectures and design patterns for complex systems, composition of software components, the proper use and extension of middleware tools, and methodological approaches for the integration process.

With component-based development, it is expected that software systems may be created and maintained at lower costs and with increased stability through reuse of approved components in flexible software architectures [5]. When those components are information systems, a frequent requirement is that the systems to be integrated are to remain autonomous. Preexisting applications (legacy systems) must still be able to use their local data without modification. In this way (financial) investment can be preserved and a smooth migration toward modern systems can take place. The notion of “federation” is originally a political term: several states join together and constitute a federal system in which each state retains its autonomy up to a certain degree. This idea of federation can be transferred to the integration of preexisting information systems, which could have been developed independently (autonomously) within different departments of an enterprise. Here, considerable overlap with research on databases and parallel/distributed systems exists.

Artificial intelligence. Mediator and multiagent architectures are developed in this area to achieve the integration of heterogeneous information sources by means of (distributed) artificial intelligence techniques [6]. Usually, the integration is managed through the mediators and agents by means of logical rules employing artificial intelligence techniques. Computations have to be coordinated, and distributed access to data/knowledge bases and ontologies is essential for the deployment of mediator and multiagent architectures.

Multimedia systems. Often, the information to be integrated for multimedia systems consists of composite objects comprising different media components such as text, video, image, or audio (for example, for digital libraries). Techniques for multimedia systems, such as MIME or the XML-based synchronized multimedia language SMIL (www.w3.org/AudioVideo), are required in that context.

Multimedia systems are often distributed over computer networks. The Web, which utilizes the Internet, is an example of a huge, but not well-organized multimedia system. With respect to SI, content-descriptive metadata relating to the meaning of the actual multimedia objects plays an important role in distributed multimedia systems. For realizing multimedia systems, techniques for distributed systems, databases, software engineering, and artificial intelligence are essential. **G**

ness objects) that address Enterprise Application Integration.

The integration of heterogeneous systems is a research topic for different disciplines in IT, and includes the consideration of the requirements of the specific application domains involved. Therefore, the study of SI is highly interdisciplinary, as discussed in the sidebar “IT Disciplines Involved in Information SI.” Despite the differences among the various disciplines involved, the work on SI focuses to a great extent on three issues: autonomy, heterogeneity, and distribution, as discussed in the sidebar “Dimensions of Information SI.”

There is often no time and justification to replace legacy systems. New functionality must be integrated with other packages, existing applications, and data sources. Therefore, SI aims at building applications that are adaptable to business and technology changes

while retaining legacy applications and legacy technology as reasonably as possible. The speed of business and technology change does not allow time for total replacement, therefore, evolution and migration of legacy and new application systems is required. Migration and evolution aim at protecting existing investments and enabling rapid response to the changing user requirements. For managing the evolution of those complex systems, it is necessary to deal with change on the organizational level, group collaboration level, and system level in a coherent manner [3].

SI plays an important role in such application areas as health care, digital libraries, e-commerce, telecommunications, Web applications, and data warehousing, to name just a few. This special section presents the various problems and solutions for SI from different perspectives, with discussion of the specific requirements of selected application domains. The articles in this

Dimensions of Information SI

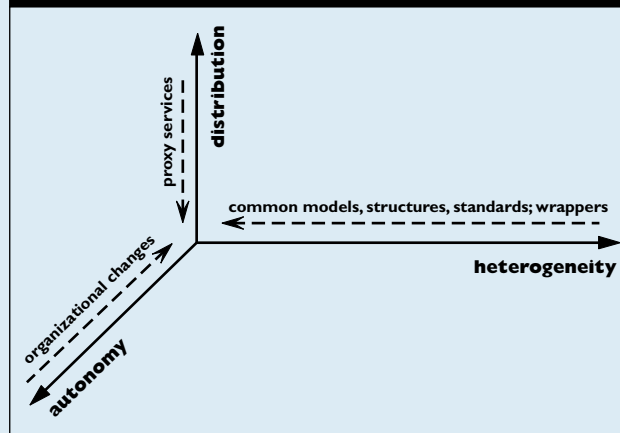
Despite the differences among the various disciplines involved, the work on information SI focuses to a great extent on three issues: autonomy, heterogeneity, and distribution as illustrated in Figure 3:

- Complex “systems of systems” are characterized by a controlled and sometimes limited integration of individual autonomous systems. Often, there are conflicts between requirements of integration and autonomy.
- Causes for heterogeneity are different database management and operating systems utilized, as well as the design autonomy among component systems.
- Much of the distribution is due to the existence of individual systems before overall systems are built (integration of legacy systems).

Usually, SI aims at approaching the origin in this system of coordinates in Figure 3. Typical solutions for the respective dimensions are:

Distribution. Proxy services are an established technique for “hiding” distribution. The idea of remote procedure calls (RPCs), for instance, is to replace the local callee’s and the remote caller’s ends of the procedure calls by stubs. The caller uses strictly local calling conventions giving him the illusion to call a local callee. In reality, it calls a (generated) stub that marshals (linearizes) the parameters and sends them to the remote end. At that end, another stub (sometimes called skeleton) receives the parameters,

Figure 3. Problem dimensions for SI: Autonomy, heterogeneity, and distribution. The dashed arrows indicate some general approaches to manage these issues.




unmarshals (de-linearizes) them, and calls the true local callee. The callee procedure itself, just as the caller, follows local calling conventions and is unaware of being called remotely. The marshaling and unmarshaling are responsible for converting data values from their local representations to some intermediate network format, and vice versa. The stubs can be regarded as proxies for the corresponding local procedures.

The Objects Management Group’s CORBA architecture, for instance, extends remote procedure calls to remote method calls in an object-oriented setting (www.omg.org).

special section can be categorized according to the three-layer architecture of information SI, namely the business, application, and technology layers.

Yang and Papazoglou address the business architecture layer in their article, discussing the integration at the level of the business architecture making reference to business-to-business e-commerce. To make e-commerce possible, it is necessary to let the information systems of dissimilar organizations cooperate. The article is a survey in which relevant problems are presented and their possible solutions are discussed in the context of a layered strawman reference architecture for inter-operation support in e-commerce. The Business Information Systems point of view to SI is presented, with achievement of business goals an important issue.

The article by Grimson et al. focuses on the application architecture, discussing the problem of integrating electronic patient records. For health care



SI aims at building applications that are adaptable to business and technology changes while retaining legacy applications and legacy technology as reasonably as possible.

information systems, integration is a decisive factor to successfully support the work within hospitals as well as for the cooperation among the various health care providers. The article describes several standards that have been defined and are being defined, and their actual use in practice. Since this article addresses the application layer, the integration among ERP packages

Heterogeneity. Due to the independent development and deployment of component systems, heterogeneity occurs at various levels and for various reasons. On a technical level, heterogeneity comes from different hardware platforms, operating systems, database management systems, and programming languages. On a conceptual level, heterogeneity comes from different programming and data models as well as different understanding and modeling of the same real-world concepts, for example, the use of the same name to denote different concepts (homonyms) and the use of different names for the same concept (synonyms).


Bridging heterogeneity is one of the most difficult tasks of SI. Typical techniques for overcoming heterogeneity are the use of common programming and data models, and similar structuring of information. Domain-specific standards are useful for defining the meaning of information to be shared among dissimilar organizations. Wrappers that provide unified interfaces are an established technique for integrating legacy systems.

The fact that autonomy of a source means not only having heterogeneity of access and representation (computer system, operating system, database system, interface conventions, and so forth), but also content heterogeneity (partial overlap, different organization, differences in term semantics) should be understood. For instance, the address of a person may be an attribute of person objects in one system and an entity with its own identity in another system. More work has been done on the former technical issues than

on the latter semantics issues, partially because semantic problems are often not noticed until the basic access problems are solved.

Autonomy. Autonomy of component systems is a critical issue for SI. Components may be autonomous in their design, meaning their developers chose the covered universe of discourse, programming models, naming concepts, and so forth. The systems may also be autonomous with respect to communication and execution, meaning that a component may independently decide how to handle interaction with the outside world.

The feasibility of reducing autonomy by technical means is highly limited. Usually, autonomy can only be reduced in connection with organizational changes. The implications of, for instance, enforcing a two-phase commit over several local databases by means of a transaction monitor may be unacceptable to the corresponding organizational departments due to the impact on the local system's execution performance.

As illustrated in Figure 3, SI aims at approaching the origin in this system of coordinates. However, it is not always possible—and even not always reasonable—attempting to eliminate autonomy, heterogeneity, or distribution entirely. For instance, distribution is a matter of fact when connecting systems of dissimilar organizations. Autonomy allows for flexible architectures whereby individual subsystems are able to adapt themselves to changing requirements. By allowing for heterogeneity, organizational departments may choose the optimal systems for achieving their individual business goals. 

and other applications is discussed in the context of Enterprise Application Integration. The Health Informatics point of view to SI is presented; to achieve semantic interoperability, health care-specific standards play an important role.

The articles by Rundensteiner et al. and Adam et al. address the technology architecture. Rundensteiner et al. discuss the technology architecture, making reference to data warehouses offering a rich view of the field. To build data warehouses that aim at supporting decision support systems, it is a basic requirement to integrate the data from operational

information systems; thus the emphasis is on coping with the dynamics in the operational information sources.

The article by Adam et al. focuses on integration of digital libraries and presents a survey of the field, addressing CORBA, mediators, and agent architectures. For digital libraries, the integration of information from different sources is a central problem to be solved. The integration of multimedia objects is an important issue in that context. The similarities of the two technology articles are evident when comparing their figures that illustrates the respective system architectures.

It should be noted that within this special section not all possible facets of information SI can be covered, but I hope you will enjoy reading the articles in this section and derive some conceptions beneficial to your own work. **C**

WILHELM HASSELBRING

(hasselbring@acm.org) is an assistant professor at the Infolab in the Department of Information Management and Computer Science, Tilburg University, The Netherlands.

REFERENCES

1. Jacobson, I., Ericsson, M., and Jacobson, A. *The Object Advantage: Business Process Reengineering with Object Technology*. Addison-Wesley, Wokingham, England, 1994.
2. Malone, T. and Crowston, K. The interdisciplinary study of coordination. *ACM Computing Surveys* 26, 1 (1994), 87–119.
3. Michelis, G.D., et al. A three-faceted view of information systems. *Commun. ACM* 41, 12 (Dec. 1998), 64–70.
4. Munz, R. Usage scenarios of DBMS. Keynote Address at VLDB'99, Sept. 1999; www.dcs.napier.ac.uk/~vldb99/IndustrialSpeakerSlides/.
5. Szyperski, C. *Component Software*. Addison-Wesley, Harlow, England, 1998.
6. Wiederhold, G., Ed. *Intelligent Integration of Information*. Kluwer Academic Publishers, Boston, 1996.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
