

Macro Patterns: Translating E-Business Strategy into Information Systems Architecture

by Borys Stokalski

Macro Patterns: Translating E-Business Strategy into Information Systems Architecture

by Borys Stokalski

I.S. ARCHITECTURE MANAGEMENT: E-BUSINESS SURVIVAL SKILL

In the era of “Internet speed,” one can feel deep disregard for formal frameworks in software development. This disregard stems from a lack of evidence that complicated mental models such as defined software processes, formal project management methodologies, and so on actually add value to our practice. Even John Zachman, father of the Enterprise

Architecture Framework, agrees that the sole notion of making and managing formal models of business is (and in a way always has been) countercultural. On the other hand, the same “Internet speed” and the very nature of intensively competitive markets may make us see that managing the corporate IT architecture effectively is a business survival skill. It affects in a number of profoundly important ways the manner in which e-business companies innovate, collaborate within supply chain networks, and communicate with their markets.

Architecture and Communication

Companies communicate with their markets in many ways. They introduce their value proposition, they sell products and services, and they deliver information components of products and services.

Communication channels — the basic components constituting business to consumer, or e-commerce initiatives — are diversifying rapidly. Such channels consist of a medium (Web, WAP device, digital TV), content (e.g., information service, transaction-related data), and a process (the way exchange of information in a channel is structured and synchronized in time) (see Figure 1). Effective competition in the e-commerce space implies effective creation, modification, and utilization of channels. In technical terms, this translates to efficient creation, maintenance, and operation of information systems constituting the channel technical infrastructure. Thus it is not the current excellence of an enterprise communication channel infrastructure that gives it a competitive advantage. The advantage comes from the ability to manage changes in this infrastructure more effectively than competitors do.

Architecture and Collaboration

The next important issue is collaboration in supply chain networks. To collaborate in a virtual supply chain means to be able to extend your business processes beyond the physical limits of your enterprise. Products are still being manufactured, ordered, delivered, paid for, and serviced, but the process must be negotiated, implemented, and optimized across many business entities. Surviving in the world of virtual supply chains requires you to be able to integrate your back-office activities and supporting information systems with the activities of your business partners. It includes issues such as sharing of planning and status information and optimizing the information-handling overheads within the whole supply chain — and all this in an environment where the rules and participants of the chain are subject to frequent and unpredictable changes. Again, the ability to quickly adapt

your back-office automation to the rules and standards of your supply chain is the key to survival, and this translates directly to excellence in managing changes in your IS architecture.

Architecture and Innovation

In the era of electronic business, there is no such thing as sustainable competitive advantage. Today products are built on relatively cheap, standardized, and widely available platforms (such as PC technology, smart cards, telecommunications, and international transport infrastructure). Thus the main area of differentiation and unique customer value lies in the services and relationships built around the platforms. These allow for virtually unlimited innovation, but they are easy to replicate (if successful).

All that remains of any online service or product is a history of market conversation and a bit of space in the minds of customers (usually short-term memory). The electronic business strategy must therefore concentrate on building the relationship with the customer and leveraging it through a continuous innovation stream. Innovation is an investment that consumes information resources. New processes must be automated, new communication channels may have to be implemented, and existing data must be used to support the decision about the scope, lifecycle, and size of opportunity. If your systems portfolio is a mess, it may effectively bar you from innovation — a very dangerous situation in the information economy.

WHERE THE I.S. SUPPLY CHAIN BREAKS

Despite the gloomy conclusions of the famous and often-cited CHAOS Report or the statistics recently revealed by Michael Mah on the Cutter IT Forum, there are some things IT people are fairly good at. Given a

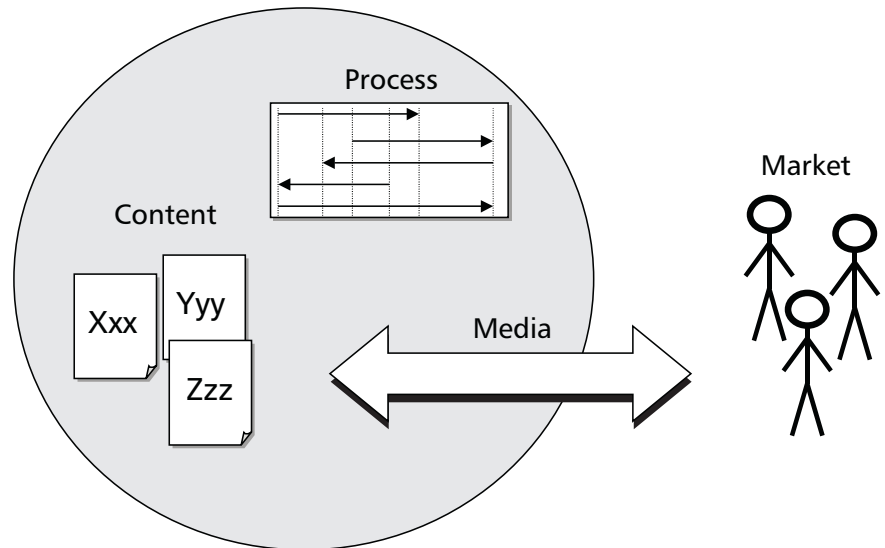


Figure 1 — Digital communication channel components.

reasonably well-written and stable specification, they can design and produce a reliable and functional piece of software on time and on budget. Software engineers are fairly effective in overcoming technical obstacles; we know how to organize to solve technical problems. Cases of successful offshore development usually involve situations where requirements are clear and stable enough to build working change management procedures around them. One of the approaches that helps to further improve the engineering part of the software supply chain is design patterns. Patterns — collections of objects with defined relationships, semantics, and an underlying mental model — help application designers to found the structure of the software on well-understood “forms,” creating a language of design and promoting the reuse of good design practices. Representing design patterns as physical components in object frameworks adds the benefits of reusability in the production and testing threads of application development as well.

If your systems portfolio is a mess, it may effectively bar you from innovation — a very dangerous situation in the information economy.

However, the low level of granularity and somewhat abstract nature of design patterns make them impractical in solving the most difficult puzzle in corporate IT management, which is translating the strategic themes of the business into a meaningful high-level IT concept that could be used well for a reliable project definition. Your company wants to implement a customer relationship management (CRM) strategy? Fine, but what should you do — buy an extension to your ERP package? Recommend building a call center? Design a customer-oriented data warehouse? Build your own application? I would like to suggest the idea of “macro patterns” — an extension of design patterns into the world of enterprise IS architecture planning — as a potential solution to this problem.

The difference between the strategic view of the corporate systems architecture and the architecture of an individual IT system is sometimes depicted as the difference between planning a city versus designing a building. City planning defines key aspects of the way a city will live — its communication infrastructure, areas for different “modes” of city life (e.g., offices, residential neighborhoods, green areas), and so on. Designing a building, while involving a lot of creativity, is nevertheless constrained by many elements of urban infrastructure. A building is defined not just by its purpose (e.g., a condominium for middle-class families with kids, a shopping mall) but also by its surroundings, the cost of real estate, and access to media and utilities, among other things. Finally, there is one more thing that translates a purpose to the design of a building — the concept of “style.” The style of a building may constrain the work of an architect to a class of forms, preferred materials, and — in many cases — particular technologies and methods of organizing the work.

MACRO PATTERNS

Macro patterns are to enterprise IS architecture planning as style and form are to disciplines of art. Choosing a “typical” style and a form for a musical work — for example, a classical symphony — suggests the number and sequence of movements and defines the basic rules that should be applied in melody, harmony, and tempo. Further, it defines rough estimates of how long the resulting piece will be and the basic set of instruments for which the score will be written. Finally, an experienced composer has a good chance of estimating how long it will take to compose the work and the price he or she expects for this future achievement. A macro pattern may therefore be best defined as an organizing principle that includes a typical layout of information systems components, a set of enabling technologies, and a set of best practices concerning the implementation of the pattern.¹ Patterns are general in nature, and their implementation is specific to the business strategy context. The idea may be best shown in a couple of examples.

Data Warehouse and Integration Broker

There are two important macro patterns relating to application integration strategies — data warehouse and integration broker. The main rationale for building a data warehouse is to create a unified, quantitative view of key business processes and events in order to support the decisionmaking activities of knowledge workers, such as business analysts, managers, business planners, and so forth. Deciding to use the data warehouse pattern to solve specific decision support problems translates into clear design objectives and project scope. At the same time, there is a set of design and implementation practices in the area of data warehousing that gives us reasonable confidence that once we know what

¹The concept of macro patterns has been developed in the IT Managers Training Centre (an educational initiative of InfoViDE) by a team including Artur Krol, Jan “Kuba” Moszczynski, Wojtek Ozimek, Piotr Fuglewicz, and myself. The thinking behind macro patterns has also been inspired by some ideas expressed in Bernard Boar, *Constructing Blueprints for Enterprise IT Architectures* (John Wiley & Sons, 1998).

information should be delivered, we can tell how to design and build a system to achieve it. Furthermore, we can assess the costs of investment and make an informed decision about the preferred scope, risk, timing considerations, and expected business impact.

The integration broker is another pattern that addresses application integration issues. Its value lies in two essential capabilities. The first capability — referred to as the “Zero Latency Enterprise” strategy — is the ability of the enterprise to synchronize all its information assets around the “current state” of business processes. The second is the ability to link information assets of disparate organizations into a single value chain. These capabilities are again generic and can be used for many different purposes. The underlying technology is mature, and there are many fairly established design practices that can be used in broker implementation.

Digital Communication Channel

Another interesting example of a macro pattern is a digital communication channel. This pattern (described earlier in this article) is emerging along with the diversification of

digital media that can be used to interact with the customer. Creating a new communication channel involves the selection of the content, media, and process (protocol) of communication. This act is — in essence — a creation of a new value proposition based on company information assets (data, application, integration broker services, etc.). Therefore, an essential practice associated with channel development is the verification of the value proposition (see Figure 2). This verification should answer three key questions: Is the new value proposition significantly different? Does the difference in value have its market? Is this market willing to pay a feasible price for the value provided? This inclusion of marketing practices in IT-based initiatives seems to be the natural course of things in the digital economy.

Table 1 gives a brief overview of some of the patterns useful for strategic systems architecture planning.

MACRO PATTERNS AND ENTERPRISE ARCHITECTURE FRAMEWORK

How can you apply a macro pattern? The answer is simple — you should use it as a

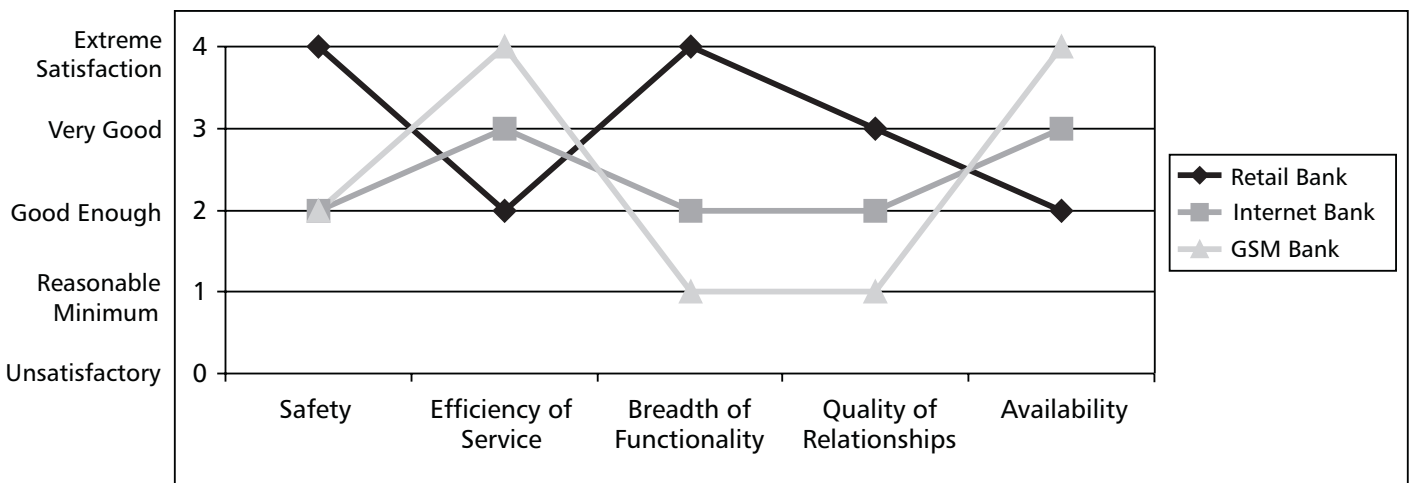


Figure 2 — Value spectrum analysis for a digital communication channel.

Table 1 — Sample Macro Patterns

Pattern	Pattern components	Sample best practices
Data warehouse	Central data warehouse Subject-oriented data marts Meta data repository Extract/transform/load engine User access layer	Corporate data model as the foundation for architecture Information requirements modeling Structured process models Architecture-based iterative design and implementation
Message broker	Message repository Message transformation and routing engine Application adapters Workflow engine Monitoring and management	Legacy wrapping Event response and business process analysis as a foundation for architecture Architecture-based iterative design and implementation Component-based development techniques
Business portal (may be externally or internally oriented)	Personalization layer Application integration layer Web application server Communication channel Client layer (this may range from a typical Web browser, to WAP mobile phones, to Internet-enabled home appliances or wearable devices)	Use cases and user classification Personalization as the key to usability Build on top of application integration solution Rule-based content selection
Electronic communication/distribution channel	Communication media Content management layer Customer access device	Channel value modeling
Workflow management	Workflow rules repository Workflow engine Workflow client layer Document repository	Workflow modeling techniques
ERP package	Subject area modules (finance, controlling, treasury, sales and distribution, asset management) Application server Database server	MRP II as the underlying management concept Standard implementation methodologies Phased module-oriented approaches

¹For more on the Enterprise Architecture Framework, see J.A. Zachman, "A Framework for Information Systems Architecture," *IBM Systems Journal*, Vol. 26, No. 3 (1987) and J.F. Sowa and J.A. Zachman, "Extending and Formalizing the Framework for Information Systems Architecture," *IBM Systems Journal*, Vol. 31, No. 3 (1992).

tool that lets you translate the strategies typical for electronic business into a well-defined information systems architecture.

John Zachman defined in his Enterprise Architecture Framework the major levels of granularity that are necessary to fully describe a concept of an enterprise information system and the major questions that

must be answered to avoid any "white spaces" in your concept of a strategic information systems portfolio.² Speaking in terms of the Enterprise Architecture Framework (see Figure 3), macro patterns help you to get from row 2 (Owner's perspective) to row 3 (Architect's perspective). Here is how you might do it for a typical e-business strategy — CRM. This strategy is based on

Layers of Enterprise Architecture
(by John Zachman)

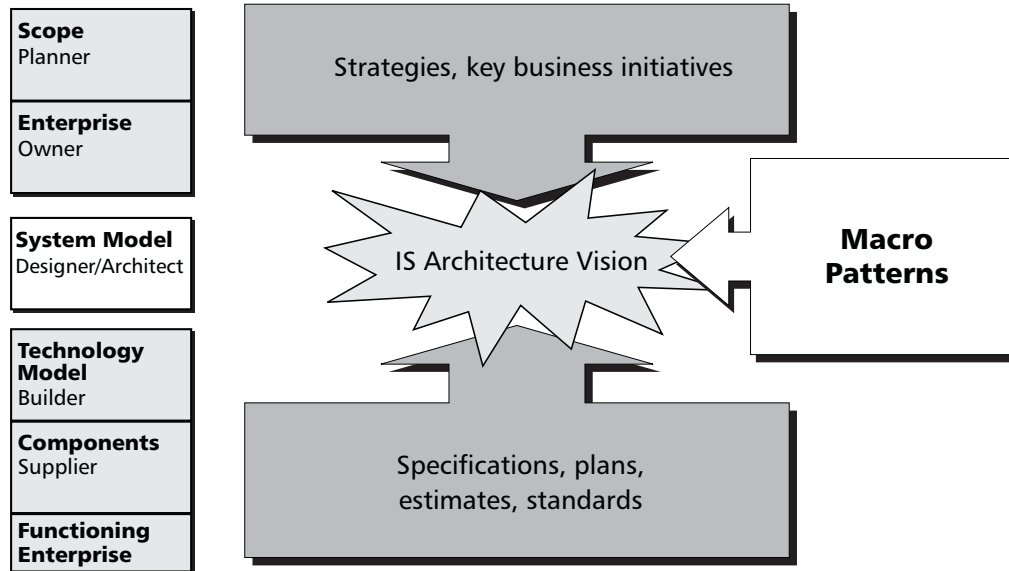


Figure 3 — Macro patterns and the Enterprise Architecture Framework.

defining a desirable balance between three main customer-related measures: customer satisfaction, customer profitability, and market share. Most of the implementations of CRM strategy target customer loyalty as a key goal that ensures both profitability and reliable market share. Customer loyalty can only be achieved if the value perceived by the customer significantly exceeds expectations. This, however, requires caution about the costs of support, so that your cherished group of customers may remain profitable. Let us try to construct a generic recipe for how to use each of the patterns to support such an approach (see Table 2).

Patterns can be assigned to many other e-business strategies, such as knowledge management (achieving strategic advantage by using information technology to support the creation, sharing, and implementation of knowledge), virtual marketplaces (online marketing, sales, and support), or supply chain management. Patterns can be

combined; for example, the information broker pattern can be used as a feeding mechanism for a data warehouse, and a corporate portal can be implemented as a personalized data access layer for the data warehouse (see Figure 4). In any case, the main advantage of using macro patterns in strategic IS planning is the fact that the resulting structure of applications, data, and network reflects good practice in high-level IS architecture design. This should yield an application portfolio that is reliable and relatively easy to extend and maintain.

Macro patterns are not a revolutionary concept, but it is important to realize that they do exist and to consider how they may help in planning IS architecture. Defining an enterprise-specific strategy-to-pattern map covering the key e-business strategies may help IT managers to rule their realm better, in terms of both the quality of systems they build and the alignment of IT infrastructure and business strategies.

Borys Stokalski is a cofounder and principal of InfoViDE, a Polish company specializing in consulting and training services for the IT community, ranked among the top IT consulting organizations operating in the Polish market. He began his career in 1985 as a research worker and then as a lecturer in the area of human-computer interaction at Warsaw University.

Today his primary responsibilities include involvement in the development and implementation of InfoViDE corporate strategy. Mr. Stokalski takes an active part in InfoViDE initiatives targeted at new service areas and consulting products, such as electronic business strategies and architectures, economic analysis of IT projects, knowledge management, and training programs for IT managers. He occasionally takes the opportunity to perform some high-level consulting for selected InfoViDE customers.

Mr. Stokalski frequently publishes articles on software engineering and business applications of information technology and is a frequent speaker at conferences and seminars.

Mr. Stokalski can be reached at InfoViDE, Kolejowa 5/7, 01-217 Warszawa, Poland. Tel: +48 22 534 74 00; Fax: +48 22 534 74 02; E-mail: bstokalski@infovide.pl.

Table 2 — Macro Patterns Application to a CRM Strategy

Pattern	Possible application/constraints
Data warehouse	Use to gather and analyze measures used to manage customer relationship (e.g., satisfaction index, response times, personnel data). Use to store and analyze customer-oriented quantifiable data (profitability, profile, relationship history).
Message broker	Use to update customer-related data on the basis of interaction happening in digital communication channels, call center operations, and activities of field representatives.
Enterprise portal	Use to communicate with the customer. Use as an interface to integrated customer data for all customer interaction activities (telemarketing, sales, clearance, support, etc).
Workflow management	Employ to automate routine handling of customer-related events.
ERP package/ packaged software	Assess the feasibility of the CRM-related functions, and consider using the package if it fits well with a significant set of your requirements and is easy to extend and integrate with other key components of your CRM architecture.

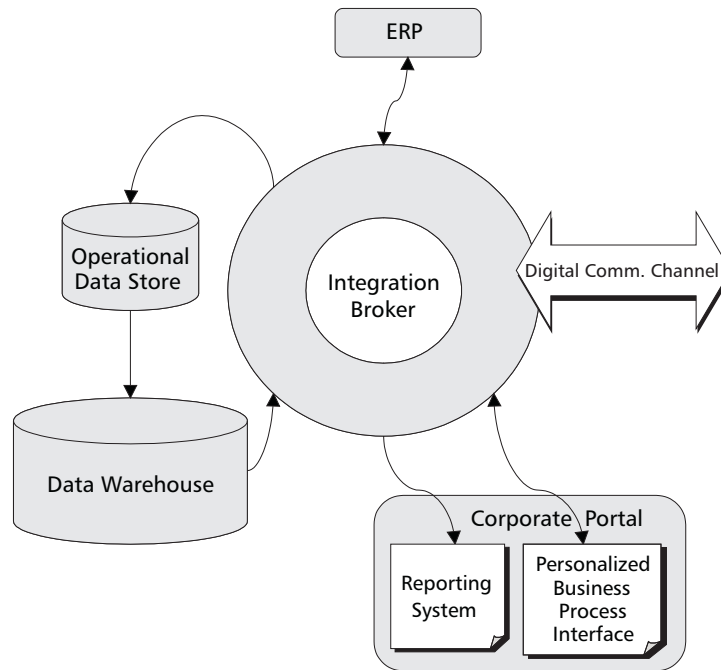


Figure 4 — Combining macro patterns.

This article originally appeared in Vol. 13, No. 8 of *Cutter IT Journal*.
Copyright 2000 by Cutter Information Corp. All rights reserved. Unauthorized reproduction, including photocopying, faxing, and scanning, is illegal.

The monthly *Cutter IT Journal* is published by:
Cutter Information Corp., 37 Broadway, Suite 1, Arlington, MA 02474-552, USA.
Tel: +1 781 641 5118, or +1 800 964 5118; Fax +1 781 648 1950 or +1 800 888 1816;
E-mail: info@cutter.com; Web site: www.cutter.com/consortium/.