13. How Models and Patterns Evolve; Emerging Technologies

DE + IA (INFO 243) - 5 March 2008

Bob Glushko

Plan for Today's Class

The big trends in the information and service economy

Web services and SOA

E-Government Architecture for Ireland

The Big Ideas of Chapter 5 (and of the Information-Powered Economy)

- Business architectures co-evolve with technology
- Information technology has radically changed the structure of firms
- Information about goods becomes a good (or a service?)
- Business models are shifting from forecast/schedule-driven to demand/event-driven
- Business relationships/architectures shifting from tightly to loosely coupled
- Business models are shifting from proprietary to standard models with reusable components

Co-evolution of Business Models and Enabling Technologies

- Business patterns are continuously evolving, mostly as a result of changes in information and communications technology
- Businesses don't just select a pattern and follow it; they may have to adapt a pattern or change to a different pattern to succeed
- New technologies pose predictable problems for the business models of incumbents (as opposed to new firms) in an industry

"The Nature of the Firm" – Coase (1937)

Why do firms exist at all? Why does an entrepreneur hire people instead of "renting" them in the marketplace?

A transaction costs analysis says that firms are created when hierarchical coordination of internal processes is more efficient than carrying out the same processes externally "in the market"

The marketplace sets prices and coordinates the actions of self-interested buyers and sellers through the "invisible hand" (Adam Smith), but it also imposes "transaction costs"

When transactions are brought inside, the administrative coordination with the "visible hand" of management and authority can reduce transaction costs

"Transaction Costs"

SEARCH – Discovery of potential business partners

INFORMATION ANALYSIS – Determining what products and services are offered and whether the partner is appropriate on other dimensions

BARGAINING – Proposing the terms of a business relationship

DECISIONMAKING – Agreeing on the terms and ensuring their fit with other business processes

MONITORING – Ensuring that the terms and conditions are being met

ENFORCEMENT – Taking corrective action if they are not

"The New Industrial State"

The size of General Motors is in the service not of monopoly or the economies of scale but planning…and (thanks to) this planning—control of supply, control of demands, provision of capital, minimization of risk—there is no clear limit to the desirable size (of the company.)

Size is the general servant of technology, not the special servant of profits. Small businesses have no need for technological innovations and can hardly afford to keep up with new technologies (as big businesses do) and therefore struggle to survive in the economical whirlwind of production and profit. The enemy is advanced technology, the specialization and organization of men and process that this requires and the resulting commitment of time and capital.

John Kenneth Galbraith (1957)

The Hierarchical Firm

The traditional industrial corporation of the mid-to-late 20th century was large, vertically integrated, and hierarchically organized to produce standardized products for mass markets

In 1960 all but two of the world's largest companies based in US

General Motors earned as much in profits as 10 biggest firms from France, UK, Germany combined (30 total)

US firms produced 50% of world output; this amounted to more than the next 9 industrial nations combined

Example: Ford's River Rouge Plant

The ultimate in vertical integration - with docks on the Rouge River, 100 miles of interior railroad track, its own electricity plant, and ore processing, raw materials were turned into running vehicles within this single complex

1.5 miles (2.4 km) wide by 1 mile (1.6 km) long, including 93 buildings with nearly 16 million square feet (1.5 km^2) of factory floor space

Over 100,000 workers worked in this single complex in the mid 1900's

River Rouge -- 1940s Aerial View

River Rouge -- 1940s Tool and Die Works

Transaction Costs and New Technologies

New technologies (e.g. telephone, mainframe computer) reduce coordination costs so firms can get bigger...

But what if new technologies reduce the external costs proportionally more than internal costs?

As communication, coordination, and monitoring costs decline because of new technology and more organizational autonomy it becomes possible to outsource non-essential functions

And makes it cheaper to work with new business partners on shorter term, more ad hoc relationships

Technical standards for product description and document exchange can also be seen as technology that reduces transaction costs

From Hierarchy to Network

Today, the large vertical integrated firm of the mid- to late- 900s has been transformed into a more "network" form, no longer driven by command-and-control

IBM, Cisco and other large firms are repositioning themselves as comprehensive "service networks" whose business units are both more autonomous and collaborative

Competition is increasingly between entire supply chains or ecosystems, not just between firms

This requires large amounts of formal and informal information exchange

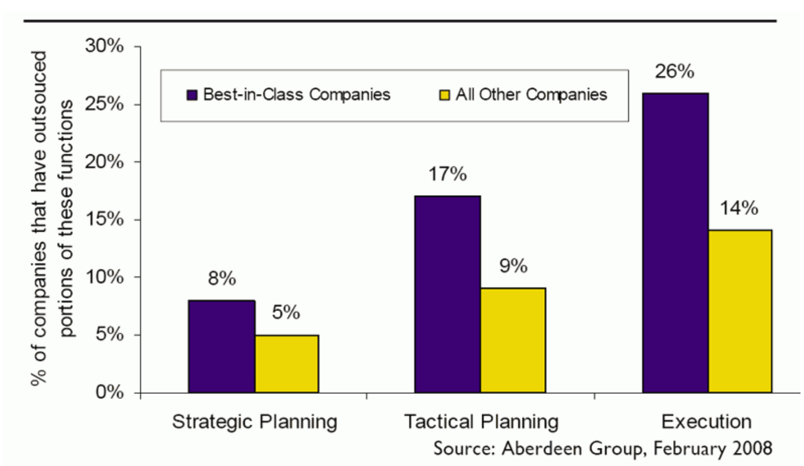
Information About Goods Becomes a Good

Information about the supply chain is taking on independent value

Information about where products are, who uses them, and when and how they are used can be worth more than the products themselves

Example: UPS Supply Chain Solutions

Smart Firms Outsource Their Logistics



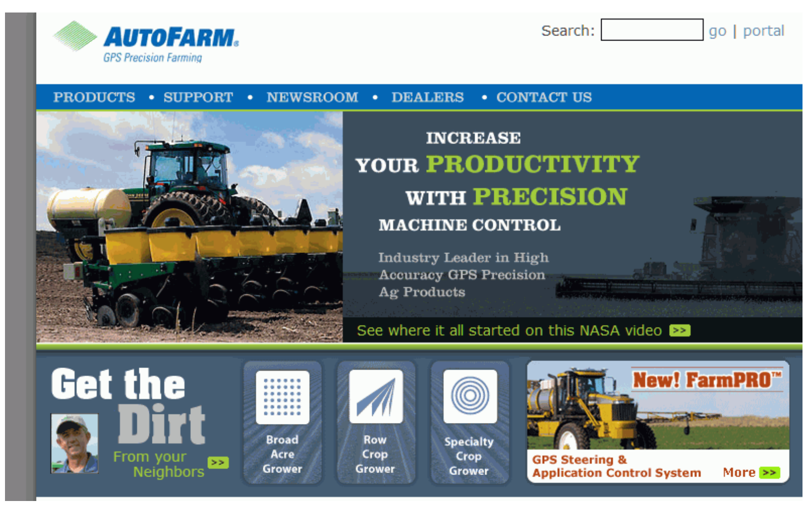
Toward On Demand/Event-Driven Business Models

No forecast can ever be as accurate as actual sales and demand information

The key to supply chain optimization isn't moving things faster according to plans, it is moving things smarter according to actual demand

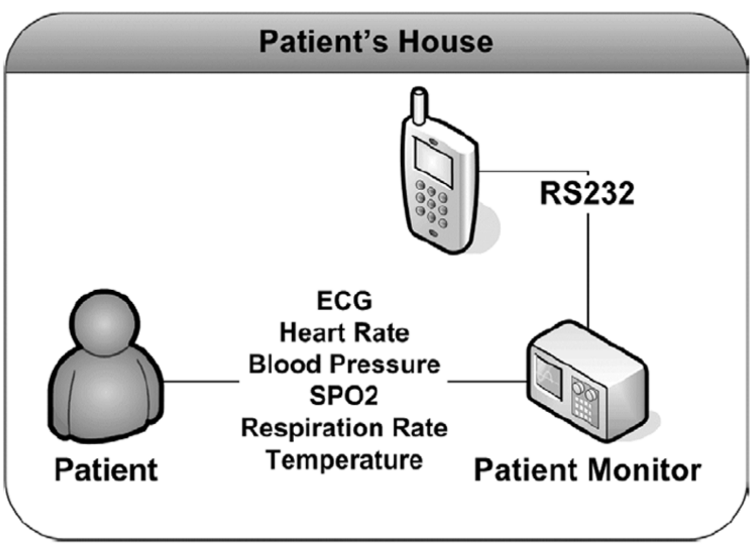
"Information-driven decisions" can be make more reliably and with less latency when sensor networks collect information

Example: GPS & Sensor-Driven "Precision Agriculture" [1]



Example: GPS & Sensor-Driven "Precision Agriculture" [2]

Example: Mobile Telemedicine for Home Care and Patient Monitoring



Example: Mobile Telemedicine -- Patient Monitor

A3 Colour Patient Monitor (M 3929 A)

A3 Colour Patient Monitor (M 3929 A)

Medical Technologies - Compact portable patient monitor Philips (Formerly Agilent/Hewlett Packard). Easy to use with large numbers and colour TFT graphical display.

EDF+ Data Format

```
<vital>
 <patient id='1234' name='Anonymous'
      birthday='20/04/1981' gender='M'>
  <historic>
   <evaluation date='02/12/2003' time='12:03:29'
      id_responsible='123' id_equipment='12'>
    <signal type='heart_rate' value='65'
      unit='/min' />
    <signal type='spo2' value='98' unit='%' />
    <signal type='mean_arterial_pressure'
      value='100' unit='mmHg' />
    <signal type='temperature' value='36.5'
      unit='C' />
   </evaluation>
  </historic>
 </patient>
</vital>
```

Tight Coupling

"Tight coupling" between two businesses, applications or services means that their interactions and information exchanges are completely automated and *optimized* in performance...

... by taking advantage of knowledge of their *internal* processes, information structures, technologies or other *private* characteristics that are not revealed in their *public* interfaces

... and usually implemented with a custom program that fit only between the two of them

Tight coupling is most often used, and usually limited to, situations in which the same party controls both ends of the information exchange

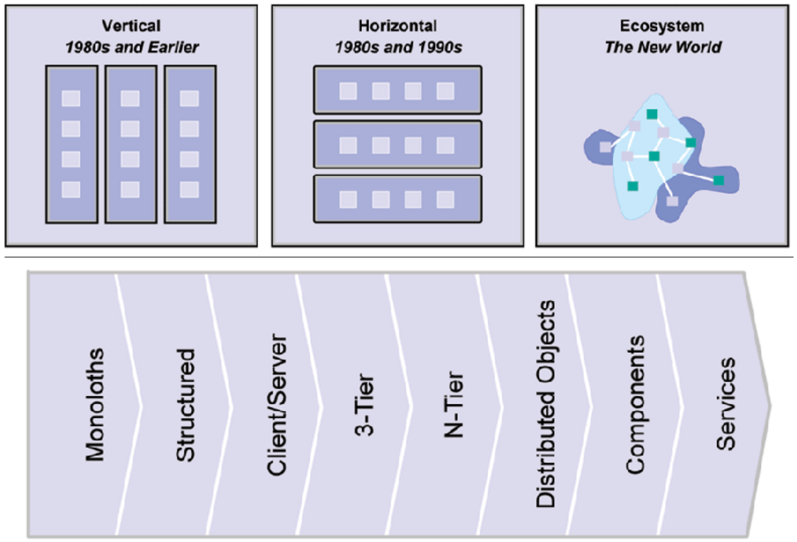
The Integration Challenge

Can we have integration and loose coupling at the same time?

The idea of service-oriented integration says we can

But we can get there from here?

Co-Evolution of Business and Technology Architecture



Document- or Service-Oriented Integration

Internet protocols and XML are enabling "loosely coupled" architectures and "coarse-grained" information exchanges that make far fewer (or no) assumptions about the implementation on the "other side"

When integration is done with loose coupling, the two sides can make (some) changes to their implementations without affecting the other

This is even more true when they communicate through an "integration hub" which can further abstract their implementation by doing transport protocol/envelope/syntax translation for them

The particular integration technology for loose coupling is less important than the philosophy or business model that requires it – treating different organizations, applications, and devices as loosely-coupled cooperating entities regardless of where they fit within or across enterprise boundaries

Service Oriented Architecture - A Conceptual Perspective and Design Philosophy

Business processes are increasingly global and involve widely dispersed parts of an enterprise or multiple enterprises

A business needs to be able to quickly and cost-effectively change how it does business and who it does business with (suppliers, business partners, customers)

A business also needs more flexible relationships with its partners and "assets" to handle variable demands

Web Services {and,vs} Service Oriented Architecture

Web services are an important PHYSICAL architectural idea and a set of standards and techniques for loose coupling

Service Oriented Architecture is a CONCEPTUAL architectural perspective and design philosophy for loose coupling

MBAs and CIOs talk about SOAs, software architects and developers talk about web services

Web Services

Web Services -- with a capital "S" -- generally means a particular set of specifications for doing service-oriented integration with XML documents as the "payload" that conveys the information required by the service interface

(Or put another way -- the interface is specified using an XML schema that defines in a formal way the information the service expects and how it should be structured)

The most important Web Service specifications are those for a service's public interfaces (Web Service Description Language) and for the messaging protocol used to send and receive XML documents through those interfaces (SOAP)

The Service Discovery Myth

Many discussions about services highlight the concept of service discovery and a specification called UDDI (Universal Description, Discovery and Integration)

UDDI was proposed as a kind of services "white" and "yellow" pages directory that would enable services to be registered by their providers and discovered by potential users, all by automated means

But UDDI is mostly used for "internal" service directories and rarely for "public" ones

Most service relationships are established "offline" and then the information about how to access the service is built into the service requestor's implementation

WS-* ("star" or "splat")

The major platform and enterprise software vendors have developed and "standardized" a few dozen specifications for extending the basic Web Services specifications to handle issues that emerge in complex distributed applications and service systems

These specifications cover things like security, multi-hop addressing, process choreography, policy assertion, performance management, ...

Their proponents argue that these additional specifications are essential for service oriented computing to be viable for enterprise-level applications and services

But they've made Web Services (with a capital "S") seem needlessly complex for a great many applications where they might have been useful

Many services are being implemented today with simpler protocols

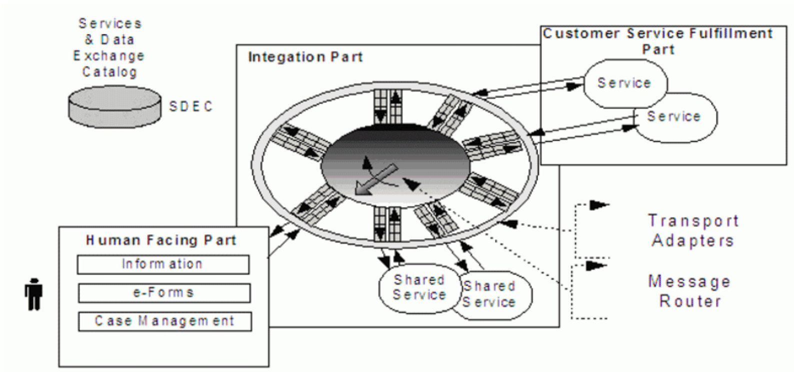
Web-based Services

This is a category coined by Erik Wilde for his courses at the I-school to mean "Web Services and any services that use any Internet protocol"

This includes services implemented using the basic HTTP protocol and its mechanisms for providing "better service" using content negotiation (provide different information to the client based on the type of browser, etc.)

This broader category makes it easier to understand and make tradeoffs in the design and implementation of services

E-Government Architecture in Ireland



SOA and Web Services with a "hub" architecture to implement e-Government services in Ireland

Abstractions of "service," "message," and "hub" make it easy to understand technology and business benefits and their interrelationships

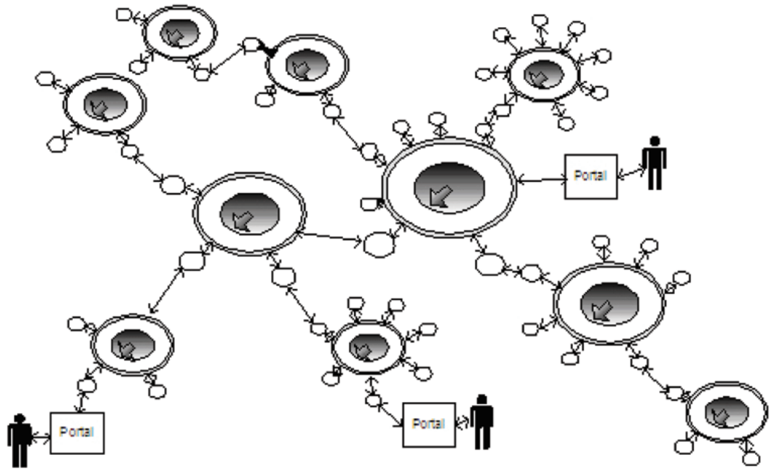
The PSB Architecture

The "Service & Data Exchange Catalog" is a registry that contains the models that define the structure and semantics of the messages exchanged to request and satisfy a service

The PSB hub is "hollow" in functionality, doing little more than message routing and security functions -- but it provides the platform into which other value-added services are "plugged in"

Among the most important of these plugged-in services are the adaptors that transform message content and protocols to ensure interoperability and composability of services

A Network of Hubs



Services from the Customer's Perspective

A "customer facing service" is split into three distinct components:

- The user interface through which people interact with the service
- The integration part (at the hub) that orchestrates the routing of messages to the (one or more) services that provide the information to the user interface
- The fulfillment part -- "where the work gets done" to create / compute / process information that gets put into a message

Some Example Messages and Integrated Services

Messages like:

- This child has been born...
- Please look up the following car registration...
- Is the person with the following identifier entitled to a free telephone allowance...

Instead of requiring a new mother to apply for benefits when her child is born, the "new child message" can be directly sent to the agency responsible for child benefits

- This made 30,000 annual applications for benefits unnecessary

Principles of E-Government Architecture

What are some of the ways in which public sector concerns and priorities differ from those of commerical enterprises?

How do these differences influence architectural decisions in e-government?

It has been said that "most software business models are based on lock-in" -- why is this a concern for governments?

Why are many governments endorsing XML?

Readings for March 10

Chapter 8 of Document Engineering

Karl Wiegers, "Habits of Effective Analysts," Software Development (1 October 2000)

B. Tommie Usdin, "When "It Doesn't Matter" Means "It Matters"", Extreme Markup Languages 2002