Business Processes Modelling

MPB (6 cfu, 295AA)

Roberto Bruni
http://www.di.unipi.it/~bruni

06 - Methodology
Objective

Coarse-grained methodology for developing business process management solutions

Guidelines for process designers to plan and conduct business process management projects

Ch.1.3, 2, 8 of Business Process Management: Concepts, Languages, Architectures
Levels of business processes

- **Strategy and Organization**: Focus on overall operations of the company.
  - Determine the operational goals, such as focus on customers, cost efficiency, leadership in innovation or quality or price.
- **Business Process Methodology**: Focus on individual business processes.
Levels of business processes

Focus on overall operations of the company

Focus on individual business processes

Determine the operational goals, such as focus on customers, cost efficiency, leadership in innovation or quality or price
Levels of business processes

long-term company strategies
to develop sustainable success in the market
Some business strategies

**Competitive Advantage:** Gained over competitors by offering consumers better value

  You increase value by lowering prices or increasing benefits and services to justify the higher price.

**Cost Leadership:** Compete for the largest number of customers through price.

  Minimize costs to the company and minimize costs to the customer without decreasing profits.

  Standardization needed: generic goods or services sold at the lowest prices.

**Differentiation Strategy:** Product or service with distinctive qualities valued by customers

  Attract customers because products are set apart from the competition.

  Leading scientific research needed:
  
  highly skilled and creative product development team;
  
  a strong sales and marketing team.

**Focus Strategy:** serve a limited group of customers better than any competitor

  Concentrate on a particular customer, product line, geographical area.

  A focus strategy works well for small but aggressive businesses.
Levels of business processes

long-term company strategies to develop sustainable success in the market

define operational goals that contribute to the realization of the business strategy
Operational goals

Management implements the business strategy by defining operational goals.

**Profitability:**
to increase revenue while limiting expenses

**Customer Service:**
to improve response time to customer complaints

**Efficiency:**
to implement a new shipping procedure that improves delivery time
Levels of business processes

Long-term company strategies to develop sustainable success in the market define operational goals that contribute to the realization of the business strategy.

High-level processes in textual form: input, output, expected results, dependencies.
# Organizational process

Top-level: Form-based description of organizational business process  
(Black-box view, internal structure not shown)

<table>
<thead>
<tr>
<th>Process Name:</th>
<th>Product Development Process</th>
<th>Responsible Process Manager:</th>
<th>Dr. Myers</th>
</tr>
</thead>
<tbody>
<tr>
<td>From:</td>
<td>Requirements</td>
<td>To: Rollout</td>
<td></td>
</tr>
<tr>
<td>Process Results:</td>
<td>Integrated and completely tested innovative product with complete documentation</td>
<td>Customer Processes:</td>
<td>Order Management Process, After-Sales Service Process</td>
</tr>
</tbody>
</table>
Intra-organizational process

No interaction with business processes performed by other parties (single organization processes)

Primary focus:
streamlining of internal processes,
eliminating activities that do not provide values,
allocating activities to persons who are competent and skilled enough

Orchestration!
Inter-organizational process

Business-to-business process (multiple organizations)

Primary focus: communication aspects, legal matters, interoperability of heterogeneous software infrastructures

Choreographies!
Levels of business processes

- **Business Strategy**: long-term company strategies to develop sustainable success in the market
- **Operational Goals**: define operational goals that contribute to the realization of the business strategy
- **Organizational Business Processes**: high-level processes in textual form: input, output, expected results, dependencies
- **Operational Business Processes**: activities and relationships are specified, but implementation aspects are disregarded

Informal & semiformal techniques: plain text + diagrams + forms-based

BPM level

- **Business Strategy**: determines, realize
- **Operational Goals**: determine, realize
- **Organizational Business Processes**: determine, realize
- **Operational Business Processes**: determine, realize
Guidelines

Recursively:
Collect observations
Classify information
Validate findings with stakeholders
Refine artefacts
Guidelines

Gather information (in textual format) about the business process environment, including project goals, project team and legislative regulations

Prepare a domain ontology to fix a common understanding of terms and concepts in the application domain
Functional decomposition

Functions at finest granularity level are called activities (rounded boxes)
Who is the customer?

Each business process starts and ends with a customer who requests a product and who receives the product as a result of the business process

(remind that a customer can be internal to the organization, e.g. a department)
Who is the owner?

Each business process is assigned a process owner, who is responsible for the process

(individual in charge of making sure that process instances are conducted correctly and that business goals are met)
Which tasks and roles?

Each business process comprises a set of activities needed to realize the business goals.

Each task may need some specific abilities to be carried out.
Which dependencies?

Execution constraints are used to order activities in the business process in a way that enterprise resources are used efficiently and at the same time the business goals are met

(process orchestration language are used to express process execution constraints)
Guidelines

Represent the (textual) gathered information as business process model(s)

Exploit the model as a communication basis to consolidate findings and improve the organizational/technical environments (e.g., acquire new skills, move to service-orientation)
Structuring business processes

Textual process descriptions are ok for coarse-grained functions

Operational business processes are ok for fine-grained functions
Repetitive vs collaborative

Highly repetitive processes, fully automated, no human involvement: process automation can pay off (e.g. online airline ticketing)

On the opposite, for rarely enacted processes it is questionable if the effort of modeling can pay off (e.g. vessel design: cost per instance too high)

Collaborative business processes, low degree of repetition, involved persons are at the centre of attention: allow to track relationships (human activities, no cost for automation)
Levels of business processes

long-term company strategies to develop sustainable success in the market

define operational goals that contribute to the realization of the business strategy

high-level processes in textual form: input, output, expected results, dependencies

activities and relationships are specified, but implementation aspects are disregarded

executable/technical/organizational environments (from written policies and procedures to enactment platforms)
Platform selection

Select the platform on which the business process will be enacted and possibly enhance the process model with additional information to make it executable.

It can be a technological platform but also a non-technical one (e.g., written business policies, manual procedures, service-oriented architecture).
Guiding principles

**Modularity** and information hiding
(encapsulation, interfaces, reuse, maintainability, response to change)
Software Architecture

**Definition:** A *software architecture* defines a structure that organizes the software elements and the resources of a software system.
A **hype cycle** is a (branded) graphic representation of the maturity, adoption and social application of specific technologies.
Early (architectures)

- Monolithic applications developed from scratch
  - Porting required redevelopment
  - Data dependency and consistency issues

1970

- Application code and (textual) user interfaces still entangled
  - Data management as a primary concern

1980

- Application and (textual) user interfaces still entangled

1990

- Human interaction made easier

Advanced user interfaces

Application

Physical data independence

DBMS

OS

GUI

Application

DBMS

OS

OS

Database

Database
Enterprise Applications

OS + DBMS + GUI + Networking capabilities = more and more elaborate information systems could be engineered

Typically hosting enterprise applications (customers, personnel, products, resources)

From individual to multiple information systems (needs integration)
Individual enterprise application

Lack of Integration!

Data redundancy!

Data dependencies!
Changes

Changes were hard to implement!

Hard to track data dependency and replication

Any modification of an application was a complex and error-prone activity, with domino effect (e.g. change of customer address format)
Enterprise Resource Planning (ERP) systems to deal with the increasing complexity of changes

Basic idea
integrated database that spans most applications, separated modules provide desired functionalities, accessed by client applications
Enterprise resource planning systems

- Integrated and consistent (centralized) database
- Two-tier client-service
- Remote data access

ERP (Enterprise Resource Planning) System Diagram:

- Human resources
- Financials
- Manufacturing

Client 1 -> ERP Server Application

Client 2 -> ERP Server Application

... -> ERP Server Application

Client n -> ERP Server Application

ERP Server Application

- DBMS
- OS

ERP Database

CRM and SCM

New types of sw entered the market around 2000

**Customer Relationship Management (CRM) systems**

**Supply Chain Management (SCM) systems**

**Goal**

to support the planning, operation, and control of supply chains, including inventory management, warehouse management, management of suppliers and distributors, and demand planning

**Problem:** different vendors, separately developed
Siloed enterprise applications

Data Integration would provide valuable information

Lack of Integration!
Data redundancy!
Data dependencies!
(on a larger scale and complexity than before)

Customer Relationship Management System

Supply Chain Management System

Connected on local network, but not logically integrated
Definition: **Enterprise Application Integration** (EAI) is defined as the use of software and computer systems architectural principles to integrate a set of enterprise computer applications.
Point-to-point integration (of silos)

N × N hard-wiring problem!

Too many interfaces to develop!

Does not respond well to changes!
(Reprogramming an interface requires considerable resources, typically)

\[
\sum_{i=1}^{N-1} i = \frac{N(N-1)}{2}
\]

Middleware technology (dedicated system integrators)

- ERP System
- CRM System
- SCM System
- Data Warehouse
- Human Resources Application
- Inventory Management

Message-oriented middleware

Messages must be encoded and decoded

Point-to-point connection problem does not diminish that much

Cooperation realized in the integration application

Offers some guarantees (e.g., message delivery)
The Hub-and-Spoke paradigm is based on a central hub and a number of spokes attached to it.

The Application Integration middleware represents the hub, and the applications to be integrated represent the spokes.

Interactions between any two applications must pass through the hub.
Hub-and-spoke integration

Configuration and management of adapters and message brokers can become cumbersome.

From $N \times N$ to $N$ integrators

Message brokers

Publish/subscribe mechanism

ERP System

CRM System

SCM System

Centralized Enterprise Application Integration Middleware (Hub)

Data Warehouse

Human Resources Application

Inventory Management

adapters
EAI implementation pitfalls

70% of all EAI projects fail (2003). Most of these failures are not due to technical difficulties, but due to management issues:

Constant change
Shortage of EAI experts
Competing standards

Loss of detail: Information unimportant at an earlier stage may become crucial later
Data protectionism
From (data-models and) data-integration

To (process-models and) process-integration
Value Chains and Process Orientation

Two major factors fuelled business process management

Value chains as a means to functionally break down the activities a company performs

Process orientation as the way to organize the activities of enterprises
Definition: a single-application workflow consists of activities and their causal and temporal ordering that are realized by one common application system.
Multiple-application workflow system

**Definition:** A multiple-application workflow contains activities that are realized by multiple application systems, providing an integration of these systems.
**System workflow**

**Definition:** A system workflow consists of activities that are implemented by software systems without any user involvement.
Workflows fit well with hub-and-spokes EAI
Limitations in workflow management

Technical integration problems:

Scarcely documented applications

Different levels of granularity

Tight coupling of applications (direct invocation)
Enterprise service computing

Main idea:

Business functionalities exposed as services

Services are equipped with usage information

Customers can find services and use them
Definition: Services are loosely-coupled computing tasks that can be dynamically discovered and invoked over the network.

Each service comes with a service description that can be published in service registries by the service provider.

Service registries can be queried by service requestors.

Service descriptions provide a level of detail that facilitates service requestors to bind and invoke them.
Service-oriented architectures

Definition: Service-oriented architectures (SOA) are software architectures that provide an environment for describing and finding software services, and for binding to services.
Service-oriented architectures
Service enabled application system

ERP Enterprise Services
service specification must be decoupled from implementation and legacy system
ERP System

standard interface

DBMS

OS

ERP Database

Composite service based application

Intra-company well-expressed as business processes

Local registry
Manual search (absence of dynamic matchmaking)
Advantages of SOA

Reuse of functionality at coarse level of granularity

New applications can be built with less effort

Existing applications can be efficiently adapted to changing requirements

Reduced maintenance and development costs
Products as services

Corporations are increasingly perceived by the set of services they provide

These services exposed to the market can be realized by enterprise services (provided by the back-end application system)

Also services provided by third parties can be integrated so that better end used services can be provided to the customer