A Software Engineering Approach and Tool Set for Developing Internet Applications

Presented by David A. Marca

to the 22nd International Conference on Software Engineering at Limrick, Ireland

Internet-ready business process for maximum control, on-line learning, real-time monitoring and selective opening
Electronic connections can be defined by assessing their effect on the supply chain.

- **Market Focused**
  - Link a base of customers to a vertical market of suppliers
  - Decreasing marketing costs + increasing orders = lowering prices
  - Requires brand recognition of web site and products

- **Production Focused**
  - Outsource a portion of the value of a business
  - Others perform the activity that create the value
  - Requires monitoring partner performance
Business factors and rules of engagement for a B2B connection can be managed by both parties.

- Use a software engineering approach and tool set.
- Create an architecture of the business relationship.
- Develop adaptable processes within that architecture.
- Support both with an adaptable technical infrastructure.

All designs constantly mirror and support the business partnership.
Adaptable e-Business connections arise when all of the following are created in order:

1. **Business Architecture** for the partnership(s).
2. **Business Design** that defines all essential policies.
3. **Process Architecture** designed for adaptability.
4. **Process Design** for the activities of the connection.
5. **Technical Architecture** that enables adaptability.
6. **Technical Design** that supports process activities.
#1 Business Architecture
Business Structure + Rules

Define boundaries around major business units, fundamental policies and external market forces.

- **Boundary** defines the roles for each business unit.
- **Policies** establish cycle times.
- **External Forces** are events that affect roles and cycle times.
- **Software Engineering principles used:**
  - Modularity
  - Well-defined Subsystem Interfaces
Define how each business unit responds to each external business event it receives.

- **Context** is one box on the Business Architecture model.
- **Events** = input arrows. **Responses** = output arrows.
- **Policies** that govern how to respond = control arrows.
- **Software Engineering principles** used:
  - Event-Response Design
  - Requirements Definition -- for Service Level Agreements (SLAs)
Business Architecture
Context for Business Design

- Business Strategy (C1)
  - Business Management (A1)
    - Margin
    - Business Plan
      - Product R&D (A2)
        - Product Characteristics
        - Product Specs
      - Time to Market
    - Raw Materials and Components (I2)
      - Orders (I1)
        - R&D Costs
      - Payments (I3)
        - Product Costs
  - Product Mfg and Distribution (A3)
    - Service Level Agreement
    - Orders and Order History
    - Filled Orders
  - Order Fulfillment (A4)
    - Sales Costs
    - Filled Orders
    - Billings (O2)
  - Customer Billing (A5)
    - Product Revenue
  - Shipments (O1)

- Margin
- Business Plan
- Product R&D
- Raw Materials and Components
- Orders
- Payments
- Product Mfg and Distribution
- Sales Costs
- Product Revenue
- Billings
- Shipments
Business Design
First Level

- Product Specs (C1)
- Time to Market (C2)
- Orders and Order History (C3)
- Product Costs (I2)
- Raw Materials and Components (I1)
- Product Plans and Budget
- Manufacture Product (A33)
- Manage Product Costs (A31)
- Manage Product
- Manage Capacity (A32)
- Manufacture Capacity
- Operational Capacity
- Manage Inventory (A34)
- Sales Volumes
- Sales Trends
- Product Orders
- Product Supply (O2)
- Shipments (O1)
- New Capacity
- Product Packages
- Distribute Product (A35)
#3 Process Architecture
Adaptable Operations

Ensure each party can adapt its operations to changing: events, cycles, market conditions.

- **Context** is one box on the Business Design model.
- **Decompose** to identify the atomic events and responses.
- **Operating Steps** create responses to events within service level agreements.

- **Software Engineering principles used:**
  - Data Transforms
  - Coupling / Cohesion
Define accountabilities, intended outcomes, expected rules of engagement, response times.

- **Process Description** for each operational step.
- **Process Template** categorizes and provides consistency.
- **Inter-linked Hypercode** versions of all process descriptions.
- **Software Engineering principles used:**
  - Pseudo-code
  - Data Repository Design
Process Architecture
Context for Process Design

- **M1** Manufacturing Line
- **C1** Line Assembly Schedule
- **A3331** Produce Detailed Parts
- **A3332** Assemble Parts
- **A3333** Major Assembly
- **I1** Product Materials
- **O1** Manufactured Products

- **Assembly Line**
- **Part Line**
- **Product Line**
- **Part Instructions**
- **Assembly Instructions**
- **Production Instructions**
- **Part Production Schedule**
- **Product Production Schedule**
- **Part Production Schedule**
Process Design

Policy

Process Policy:
2. Orders determine how much Raw Materials and Subparts to order.
3. Billings match, one-for-one, Shipments.
4. Customer Payments are matched to their respective Billings.

ISO9000

Inputs

Orders

Payments

Raw Materials and Subparts

Product Revenues

Links:

file:///C:/process\MCSE\Glossary/Orders.txt
file:///C:/process\MCSE\Glossary/Payments.txt
file:///C:/process\MCSE\Glossary/Raw Materials and Subparts.txt
file:///C:/process\MCSE\Glossary/Product Revenues.txt

©OpenProcess, Inc.
#5 Technical Architecture
Mirrors the Process

Define thin-client solution, preserve boundaries, isolate policies and break up applications.

- Dedicated Server for each business unit, for each event-response pair.
- Process and Data Allocation across the computer topology.
- Small GUIs so steps use only the application logic they really need.
- Software Engineering principles used:
  - Adaptable Client/Server Architecture
  - Graphical User Interface Design
Technical Architecture

Client

Internet Browser

Server

Wed Site Generated by the OpenProcess™ Tool

User Interface

User Interface

User Interface

Stored Procedure

Stored Procedure

Stored Procedure

Mainframe

Data Access

Data Access

Data Access

Data Access

Database
Each Business Event Requires a Response
People Respond by Following Their Particular Process
☆ That Particular Process is Part of the Whole Process
Process is Supported by Applications via Web Forms
Application Logic and Web Forms are Always Separated
Some Application Logic Performs the OLTP
Some Application Logic Performs the OLAP
#6 Technical Design

Links to the Infrastructure

Define keyword for each distinct end-user action, needed business data, policy, instruction, etc.

- One Web-based GUI for each end-user action.
- One “Live” File for each data, policy, instruction, etc.
- One Hyperlink to each Web-based GUI and Live File.

Software Engineering principles used:
- Graphical User Interface Design of Web-based Forms
- Structured Programming (of Office Suite Applications)
Technical Design
Hyperlink to Live Data

Hyperlink:
- External Website
- Intranet Page
- Web Form
- Application
- Live File
Assign different colors to inputs, outputs, controls and mechanisms. Customize at run-time.

- **Novice Users** do not customize their browser
  - They use the multi-colored links to learn the process

- **Infrequent Users** keep process and data cues visually present
  - Override with just one color -- thereby creating reminders

- **Expert Users** read process descriptions as ordinary text
  - Turn off all link colors and turn on the hover color
Conclusion

Software engineering principles are applicable to business-to-business electronic connections.

- **Design before Implementation**
  - The business partnership is designed correctly

- **Rapid Application Development**
  - The connection is built fast enough

- **Highly Modular Architecture**
  - The operations can quickly adapt to changing business direction