Workflows as Commodity: Key Issues for the Future

(Nancy, France, September 2005)
Agenda

Web Services as Commodity
Application Structures
Utility Computing
Flow Optimization
Summary
EAI – The Problem

- SAP Order Entry
- Siebel Update Customer Record
- Oracle Update Available Credit
- CICS StockHold_Tran
- DB2 Log High-Value Order
- ERP
- TPM
- DBS

Application Bridges

- Application A
- Application B
- A→B Bridge

Application Bridges:
- CICS EXCI
- SAP RFC
- OS CALL
- ...

© Frank Leymann
Adapters And Queuing: Advent of QoS

Source Environment
Application A
Source Adapter

Message Queuing

Target Environment
Application B
Target Adapter

Advent of Brokers

Message xform, Side effects, ...
("Side Effects")
Flows For “Side Effects”

- GET msg FROM queue OrderValue > 10k€
- XFORM into adapter_1 format
- PUT msg INTO Q1 (target adapter_1)
- STORE msg into DB
- APPEND to ErrorLog
- SEND eMail to Admin
- XFORM into DB format

Queuing-Based Integration Architecture

- Application i
  - (Source) Adapter
  - Broker
  - Message Queuing System
- Application j
  - (Target) Adapter
- Application k
  - (Target) Adapter
- µFlow Engine

Analyze msg & call corresponding µflow
Application Server-Based Connector Architecture

AppServer Core

Extensions For Connector Plugability

Application specific Connector

Source Application

Target Application

µFlow Engine

JMS

Wrapper: Some Sort Of Adapters

DBMS

SELECT...FROM...WHERE

DBMS Extensions for wrapper plugability

Wrapper

Wrapper

Wrapper

Wrapper

Other DBMS

Standard Application

File System

Legacy Application

© Frank Leymann
Flows & Wrappers In Databases

SELECT...FROM ..., T, ... WHERE...

DBMS

SQL

Wrapper

T

WFMS

ERP, PDM, HR, ..., RTO

Flows & Channels In B2B Interactions

Enterprise A

Public Process

Enterprise B

Public Process
And So On?

- Well, this integration technology is confusing:
  - Adapters
  - Connectors
  - Wrappers
  - Channels
  - …
- Where does it stop?

→ Web Services !!!

Web Service Technology: The Key Thing!

1. Web service technology provides a “virtual component model” for using components in a loosely coupled manner

2. When using a Web service the supporting container hides its “middleware idiosyncrasies“
   (component model behind the implementation of the Web service, the invocation protocol etc.)

3. Web service technology does not provide a new model for implementing components
   (with the exception of BPEL 😊)
The Ordering and Guiding Principles

- All standard efforts should foster…
  1. an environment which is
     a. heterogeneous,
     b. distributed,
     c. interoperable,
     d. loosely coupled,
     e. multi-vendor
  2. secure, reliable, recoverable interactions
  3. a single uniform usage model for “components”
  4. business criteria to discover components needed
Matching Endpoints Based On Policies

Service Discovery

Selected Port

Candidate List

Don't Care! Virtualizing Services

I want ...
But I don't care about a particular provider: Chose the one who is at this point in time „the best“ for me!
Distributed Heterogeneous Environment

The Grid: Virtualizing Resources
Agenda

Web Services as Commodity

Application Structures

Utility Computing

Flow Optimization

Summary

Reminder: Process-Based Applications

Business Process Models

Business Functions

.EXE  CICS  MQ  EJB  (D)COM
Modeling Business Processes

Microflows Are Business Processes
Business Processes Between Web Services

Business Processes As Web Services
Summary: Aggregating Web Services

The Notion Of “Orchestration”
Programming Model

Runtime: The Need For Deployment
Programming Model: Refinement

- **Programming in the Large**
  - Application Flow
  - Workflow System

- **Deployment**
  - DeploymentDesc
  - Service Bus

- **Programming in the Small**
  - Components
  - Application Server

Supporting People Activities

- BPEL4People

- (abstract organizational group)
Supporting People: Deployment

Organisation Database

Application Artifacts

Deployment Descriptor

... staff query...

Task

Workflow Language Genealogy (incomplete)

BPEL

BPEL4People

WS-CDL

(W3C)

BPML

(WfMC)

WSCI

(SUN, SAP, Oracle)

BPSS

(OASIS ebXML)

FDML

(IMM)

FDL

(IMM)

XLANG

(Microsoft)

XPDL

(WfMC)

WSFL

(IMM)

BPXL

(bpm.org)

BPELJ

Research

YAWL

... GSFL

(bpm.org)

(IMM, Microsoft, SAP, BEA → OASIS)
Model Representations and Semantics

- Process model often has different representation at business level and IT level
- Semantic descriptions help to identify/match them

Deriving Services From Semantics

- Selection of appropriate port types may be done base on semantic descriptions at the business level
Deriving Semantics of Process Models

- Semantic description of (fragments of) process model may be derived from semantic Web services and (causal,… relations expressed via) process model itself

Business → IT Transformation

- Fragments of process models may be translated into IT variants based on semantic matchmaking
A solution is a package containing all the software artifacts required to solve a business problem.

- A solution does not assume additional programming in order to become executable.
- But some of the artifacts of a solution might require binding/deployment such that the solution becomes executable.

Application Template

- …is a package that enables customization.
- …might even require implementation work to result into a solution.
  - For example, process model may only be partially defined, message structures may be only course grained, rules may only be sketched,…
    → BPEL “opaque”
- …may represent “best practices” that have to be refined at a particular site.
- Points-of-variability specify what can be changed without jeopardizing integrity or that must be completed/refined.
What Is Typically Customized

“Points-of-Variability: Application Logic”

Customizing Application Environment

Environment Customization Tool (SLAs, Deployment Descriptors, ...)

Environment Element

Environment Element
Finally: Solution And Environment

For Business Process Components Hardware DBMS Operating System Middleware

Environment Adaptation

Even Then: Runtime Adaptability Needed

Logic adaptation (Ad hoc, instance migration,...)

Environment Adaptation

Organizational adaptation (Staff assignment, abstract roles,...)

Binding adaptation (Deployment, late binding, semantics,...)

Who? What? With?

Accept Order

Good Customer?

Bad Customer?

Reject Order

Check Customer

Accept Order

Get Order

Always!

Credit Card Information

Customer Address

Customer

Order

Accept Order

Reject Order

Good Customer?

Bad Customer?

Check Customer

Accept Order

Reject Order

Credit Card Information

Customer Address

Customer

Order

Always!
Manageable Resource

Manageability Interface

- Allows to plug resource into a management system.
- It makes the resource manageable (e.g. lifecycle, monitor, metadata, state...)

Resource Interface

- Resource specific functions, e.g. mountDisk, openDatabase, allocateCPU, ...
**Manageability: Sensors And Effectors**

- **Sensors** enable monitoring by providing queries ("pull") on properties (actual state, metadata and data) as well as change notification ("push")

- **Effectors** enable explicit state management by providing lifecycle operations and update operations on state and data

---

**Resources Are Web Services 😊**

- **Concept**
  - Manageability Interface
  - Manageable Resource
  - Resource Interface

- **Specification**
  - Port Type
  - Properties (status, data, metadata)

---

© Frank Leymann
Notification: Base For Monitoring

WS-Notification*

Notification Source ➔ Subscriber ➔ Notification Sink Y

Subscribe sink Y to events X
Publish notification about X
Deliver notification message

Broker

Bus

MAPE and Service Bus

Analyze ➔ Plan ➔ Execute

Monitor ➔ Publish/Subscribe Features ➔ Service Bus ➔ Invocation Features

1. Resource

2. Sensors

3. Effectors

4. Execute

5. Plan

6. Analyze

Invocation Features

© Frank Leymann
Management: Using The Bus

Monitor  Operate  Configure

Service Bus

Monitoring  Operating  Configuring

Computing On Demand

In 3 hours I need the following environment for 45 minutes: 253 machines, 1.7 TB storage, WebSphere, DB2, and the following ERP applications...
The Ingredients Of On Demand

- Business Processes
- Grid
- Autonomic
- Web Services
- On Demand

Outsourcing
(Information Technology)
Terminology

- **Outsourcing**
  Allocation of IT orders to external companies (e.g. running of [parts of] IT infrastructure including application functionality, development of application functions)

- **Off-Shoring**
  Special case of outsourcing: external company is in foreign country

Situation

- **ASP* model was never really successful:**
  - Model could not exploit economy of scale
  - Interoperability problems
  Technology was not ready for this!

- **Recent advancements in Web service-, Grid-, autonomic-, and workflow-technologies are about to remove these obstructions**
  - Recently, outsourcing of both, application functions and business processes is rising sharply
  - Number of intermediate aggregators is rising

➤ **Shift from ASP model to Utility model!**
The Change

- Providers become more cost-effective based on being able to exploit economies of scale
  - Just like traditional public utilities do since decades
  - The key enabler is “dynamic provisioning”
    - I.e. making resources available only when needed
      - “Resources” = hardware, storage, networking, middleware, application functions, business processes
- Consumers get more trust
  - They only have to pay what they use
  - Dependency on third-party – i.e. provider lock-in – is significantly reduced
    - Web service interoperability allows to change providers without major re-write of software
      - Message architecture, security, interfaces, policies, processes, SLAs,…

Reasons For Outsourcing

- They are manifold
- They and their priority change(d) over time
- Cost-savings is mostly ranked high
  - Maintenance of infrastructure and software
    - Purchase cost often not so critical
    - Keeping it current is expensive
  - Skilled IT personnel is rare
    - Operators, systems management personnel
    - Complexity of IT is constantly raising → Training (costly) to avoid (costly) outages,…
    - Programmers
- Focus on core competencies is important
  - IT is not the proper business of companies
- Quality improvement is important
  - Application backlog is obstruction to proper functionality
  - Skills of own personnel in some areas not appropriate
Outsourcing Applications: It Is Done!

“Software As A Service” (SaaS)
Sample SaaS

AppShore Delivers Web Based
On Demand Applications And Services
Everything Your Business Needs To Succeed

Features Tour Online Demo Free Trial Subscribe

AppShore Free Edition
- Contact Management
- Lead Management
- Accounts
- Opportunities
- Activities
- Forecast
- Dashboards
- Case Management

AppShore Professional Edition
- All applications offered in Free Edition, plus
  - Project Management
  - Document Management
  - And many more...

Aggregate Service Providers

GRAND CENTRAL
COMMUNICATIONS

A rich ecosystem of services for you to leverage:
In addition to Grand Central’s desk phone “Buy for Success, Not Deep for Success” approach, one of the most powerful aspects of our Business services network comes from our resellable and rapidly growing collection of business services.

Our customers are able to easily integrate services from leading solutions like salesforce.com and onboard with their own internal systems like Quickbooks and Oracle.

Through Grand Central's service discovery capability our communities of developers, partners, and customers can easily discover and share services.

Central private services
High-quality service management services enable administrators to have full control over who can access, terminate, and manage their services and business services.
Outsourcing Highly-Customized Applications

A Pessimistic Scenario

- Your headquarter
- "Your" compute center (business processes)
- "Your" compute center (application functions)
- "Your" programmers

- Your headquarter
- "Your" compute center (business processes)
- "Your" compute center (application functions)
- "Your" programmers

© Frank Leymann
Geographic Distribution of Application

But There Is Hope: Shift of Skills

Your and your application

Your headquarter

Your IT architects

Your process specialists

“Your” Operators

“Your” programmers
And There Are Counter-Arguments !!!

- "Off-shoring of infrastructure is high risk"
  - Low-income countries sometimes in political, geographical,… risky areas of the world
  - I.e. higher probability of outages of infrastructure

- "Outsourcing of programming is brain-drain"
  - Company is loosing important skills
  - You become dependent on third party

- "Salaries in low-income countries are raising"
  - Outsourcing/off-shoring is only temporary; try to avoid it

- "Outsourcing/off-shoring of programming has hidden costs"
  - Functional requirements must be documented down to very low level → Lot’s of meetings, reviews,…
    - Increased costs for remaining staff
  - Quality is not up to standards used too

- "Legal aspects are painful"
  - SLAs negotiation is very critical
  - Are electronically negotiated contracts legally binding?
    - Automatic, dynamic select of providers assume corresponding contracting & negotiations
  - Cross-country contracts (off-shoring) produce additional costs requiring special lawyer skills

- …

Pragmatics

Reality will be somewhere in between 😊
Agenda

Web Services as Commodity
Application Structures
Utility Computing
Flow Optimization
Summary

Remember: Some Years Ago…

Go with the flow!
Today…

Everybody wants to go with the flow!

We Saw Just a Few Minutes Ago…

- …the use of flow technology in:
  - Enterprise application integration
  - Database technology
  - Provisioning
  - Packaged application structures

- …and don’t forget the “classical” usage of workflow technology
…And There Is Much More “Demand”!!!

- Portal personalization
- Warehousing
- Adaptive Applications
- Peer-To-Peer Flows
- Grid applications
- Grid Transaction-Management
- ...

Thus, Soon…

Nobody can go without flow!
A Déjà Vu – Nearly 😊

Panta Rei (≈ Everything flows)
(Heraklit, 550 B.C. - ~480 B.C.)

Consequence

- Flow technology becomes extremely critical in terms of performance
  (= overall runtime overhead)!!
  - WFMS must become “homeopathic” 😊

- Remember:
  Availability and scalability already addressed by today’s production workflow systems
The Conclusion

- Flow technology is now where RDBMS technology was about 20 years ago:
  
  Initial success resulted in huge demand from broad spectrum of application areas

- Different application areas require different optimized behavior → This is what we have to focus on too:

  **Flow Optimization!**
Flow Optimization: Next Steps

- As in RDBMS this has two major aspects

- Algebraic optimization
  - Syntactic rewrite of request – in flow speech:
    - Graph surgery based on Behavior Spheres

- Heuristic optimization
  - Statistics and cost based plan derivation – in flow speech:
    - Audit trail based graph rewrite using analysis and mining technologies

And This Is A Long-Lasting Vision?

- Yes, it is!

- Flow technology has matured, it becomes mainstream – finally

- The optimization vision is long-lasting
  - Will take a decade of research & development!
  - Complex technology, challenging jobs!

- It will be “blood, sweat, and tears” !!!
Conclusion

- Web service technology and workflow technology is about to become a commodity
- Service Bus & WfMS are underlying middleware
- Process Models as base of new tradable artifacts
- “Optimization” must become focus of workflow research and development
- Autonomic and On Demand based on the above
- Socio-political impact
  - Programming-in-the-small in “low-income” countries
    - Outsourcing, offshoring
  - Programming-in-the-large “anywhere”
    - Especially in “high-income” countries & close to using companies
  - Impact on (university) education
The End!
Thanks
For Your Attention...