Adaptive Software Engineering
G22.3033-007

Session 4 – Sub-Topic 3
Business Process Management
and
Related Standards

Dr. Jean-Claude Franchitti

New York University
Computer Science Department
Courant Institute of Mathematical Sciences

Part I

Business Process Management (BPM)
The little story of BPM

My grand-father the workflow…

• Starts in the late 70s
• Theoretical foundation
  – Petri nets
• Dedicated to document management
• Principle
  – Manage the lifecycle of resources in a distributed environment and the state changes that happen over long periods of time (> a user session)
My first steps …

• Born around 1997-1998 in the rage of B2B
  – I am an « adaptor » between real-time B2B transactions and slower, often batch oriented enterprise systems
  – The fundamental principle is the same, « documents » become XML documents
• Standardization in August 2000: BPML
  – Dozens of startups are rushing to the gold…
• New theoretical foundation
  – Invented in the late 80s by Robin Milner
  – Pi-calculus

My teenage crises …

• Re-org: wander in the land of EAI after the B2B bubble burst
• Disappearance of my prefered standard: BPML, for a glass of WSCI and a plate of BPEL
• Isolation
  – My barrier of entry is too high (Big Bang effect)
  – Impossible or almost impossible … to get into a J2EE or .NET architecture
• Controlled by the big guys for the better or the worst
  – Microsoft, IBM, …
**Grown-up…**

- Who am I?
- Where am I going?
- What am I doing?

- :-( … after 6 years of efforts we cannot talk about success )-:

**But, Where is the Software Industry Going?**
What’s happening today...

- Verticalization of the application architecture
  - The big five: IBM, MS, OpenSource, Oracle, SAP
  - « full-stack »
- Aggregation of all architecture concepts
  - B2B, EAI (webMethod only independent vendor left)
  - BPM
  - Portals…
- Transformation of business applications
  - We just finished writing these applications and we now need to change the infrastructure …
    • Take advantage of the web
    • More federation…
    • But also cost of infrastructure software (Open Source or home grown)

Software Industry is in transition

- Technologically, we are at the end of the « web-based » lifecycle
  - An HTML browser is not a viable client
    • It is an auxiliary client for very specific purposes
  - We are inventing the « rich client » which brings together the advantages of the browser and native client
    • Zero administration
    • Federated UI (can hit multiple « servers »)
    • UI is interpreted not compiled
  - The J2EE vendors start to spread their offer, in complete anarchy to differentiate from each other and respond to the pressure of .NET which start to go beyond « web-based »
Software Industry is in transition

- Architecturally, the complexity is such that:
  - We cannot fit all the information for a given role in a single system
    - The borders of application disappear
  - IT cannot absorb new technologies
    - How would you go about changing all the plumbing and electricity of a skyscraper? … Go wireless?
  - Users do not use Business application
    - It is now sometimes more productive to work by hand when dealing with so many systems which are incompatible and not integrated, let alone cannot capture or report on the information that they need
  - It is harder and harder to maintain and evolve business applications
    - Every changes has global impact and a high cost, often generating negative ROI and terrible delays

At the same time, technology convergence has accelerated
And the Customer Needs Have Never Been so Complex …

• Fundamental changes in the enterprise
  – Specialist => Generalists « without borders »
  – Commercial Success of the model « built-to-order » or « engineer-to-order »
  – Cycles: shorter and shorter
  – Geographical Distribution: increasing way beyond manufacturing
  – Budgets: as low as they can be
  – Re-engineering: constant for production facilities, information systems and organization

• « Real-time » Enterprise
  – All possible « touch points » must be available securely
    • information and execution
    • Internal and external
    • Human and systems

• Complexity of the tasks
  – Deal with a much larger volume of data, information and knowledge
  – Embedded in complex processes (task coordination)
  – Resource access is « highly distributed »
And the Customer Needs Have Never Been so Complex …

- Decrease of Product Lifecycles
  - Establish processes for
    - Design
    - Production
    - Logistic
    - Support
    - Maintenance
  - Global Collaboration via virtual organisations
    - “Network engineering and manufacturing”
    - Supply chain optimization
    - Demand chain optimization

All this results in a globalization of the information and processes
We just reached the stage of «massively connected systems »

- 2002-2003 are a turning point for business application and their architectures
  - Everything that could be networked has been networked
- We are going to spend the next 10 years exploiting this connectivity bounty
  - B2B
  - Real-time
  - …

This is going to be supported by the « Web-centric » model
All our beliefs have been swept away, we are at the maximum confusion point

- Continuum Business object … Document
  - Consumer oriented (no more producer contracts)
- Messages and Services
  - No more « distributed objects »
  - Service grids
- Asynchronous communications
- Federation and Collaboration
  - As opposed to « Integration »
  - Business application borders are disappearing
- Language(s)
  - Semantic (syntactic)
  - Declarative (Procedural)
  - Improving the productivity going from functional specification directly to execution

We are reaching the concept of Loose Coupling

Fortunately the first web-centric « bricks » have been laid out…

- XML
- « peer-to-peer » asynchronous communications
- The remaining problem in a massively connected architecture
  - Managing state changes and in particular the alignment of these changes between all the entities
  - In complete security !!
BPM in Application Architecture

A roadmap… so we don’t get lost

- BPM and UI
- BPM and business logic
  - Web Services
  - Service Oriented Architecture
- BPM and EAI
- BPM and B2B
User Activities in a Business Application are Asymmetric

• This is critical to realize that to understand the role of BPM in an application architecture
  – Queries to « pull » business objects
  – Tasks such as
    • Create/Delete business objects
    • Change their state
    • Establish relationships between business objects
• The link between BPM and the user interface happens at the task level
• BPM is not a « tier » of the architecture, it is rather a brick

The Tasks are the Touchpoints Between Users and Business Processes

• Architecturaly, we cannot separate the two aspects of the user interface
  – A Business Process Engine is then by definition « embedded » completely separated from the user interface (tasks and queries).
  – A user becomes a « service » like any other which exchanges messages with the business process engine
  – Inbox concept to keep the system asynchronous
• This approach allows for decoupling the « user session » (short lived) from the lifecycle of business objects (long lived)
  – A purchase order is going to go through various states: created, approved, ordered, received, paid, archived…
  – These state changes happen upon events (B2B, user, legal…)
The Model-View-Controller pattern revisited

We are going to reach the point of a complete separation of the business logic and UI
But… what about the concept of a « central » process engine?

- Enterprise-wide and omnipotent
- Well…
- This notion has been introduced for the B2B where it is acceptable and accepted
- Does not completely work for EAI integration scenarios
- Does not fit at all the application architecture
- This concept is slowing down the development of BPM technologies

A business process does not have a “center”…

OAGIS 8.0 Scenario 41
IT is *de facto* “peer-to-peer”

But…what about the web service vision?
One Must also Differentiate between Orchestration and Choreography

Orchestration Provide the Perfect Model for Long Running Business Logic of Services
This is exactly what « BPEL Java embedding » is doing … mix code and Orchestration

On the other hand Choreography Manages the Links between the Long Running Business Logic in a SOA

• However, loose coupling can only be reached if we provide
  – Mapping
  – Routing
• Orchestration
  – J2EE, .NET
• Choreography
  – EAI, B2B
  – SOA
So …

- **Orchestration**
  - « … is an emerging [concept] that would give programmers a way to formally describe processes underlying business applications so that they can be exposed and linked to processes in other applications »
- **Choreography**
  - Is a concept that specifies how these processes are linked together across the enterprise
  - Choreography can be « active » when mapping and routing are necessary

And what about Collaboration?

- The inter-enterprise message exchange is sufficiently specific to justify the use of protocols for:
  - Business transactions
  - Business state alignment
  - Business object state synchronization…
- A choreography specifies the message exchange, a collaboration and associated protocols specifies the meaning of these message
  - Signal
  - Notification
  - Contract
  - Sending a shared business object…
Putting it all together: Concepts and Standards

BPM

Collaboration  BPSS

WS-transaction  Protocols  BPSS  BTP

Orchestration Composition  BPEL

Choreography Coordination  WS-CHOR  WS-CAF

BPM = protocol://Collaboration(Choreography) + (Orchestration*) + (Task*)

Business Process Execution Is Distributed and Not Centralized

Integration Server (Decoupling for B2B and EAI)
Using BPM technologies today…

- Ok, that’s a bit of science fiction…
- You can get started today
  - Design and use web services that will potentially fit in business processes later
  - Deploy embedded engines (commercial or open source) to manage the lifecycle of business objects
  - Of course, can do some EAI and B2B today

The ultimate BPM standard…
« Standards », who said « standard »?

<table>
<thead>
<tr>
<th>OASIS</th>
<th>ws-bpel</th>
<th>ebBP</th>
<th>ebXML</th>
<th>ebMS</th>
<th>ebRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDDI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>W3C</th>
<th>ws-chor</th>
<th>ws-dl</th>
<th>XMLP (SOAP)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Others</th>
<th>OAGIS</th>
<th>WS-I</th>
<th>ebCC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UDDI</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

« Le Stack »

<table>
<thead>
<tr>
<th>SOA</th>
<th>Service</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Business Integration</td>
<td>Messaging</td>
</tr>
<tr>
<td></td>
<td>Application</td>
<td>Content</td>
</tr>
<tr>
<td></td>
<td>Discovery</td>
<td>Transport</td>
</tr>
<tr>
<td></td>
<td>Quality of Service</td>
<td></td>
</tr>
</tbody>
</table>

| | Business Integration Languages | Choreography Languages |
| | Business Agreements Languages | Interaction Languages |
| | Orchestration Languages | Security |
| | Transaction | Coordination |
| | Context | Description |

| | XMLP (SOAP) | URI |
| | Reliable Messaging | Routing, Addressing |
| | XML, t. e. XML Schema, ... | XML |
| | HTTP, SMTP, IIOP, ... | |

Runime | Design time

© ebxml.org, 2003
Conclusion

• Business application architecture is undergoing major mutation
  – Chopping off the user interface
  – Integration capabilities are now THE value of these applications
• BPM is the third fundamental brick of web-centric SOA architectures after XML and peer-to-peer Web Services
• BPM technology still need to evolve dramatically
  – Orchestration, Choreography, Protocols and Collaboration
  – Powerful tools
  – Distributed engine as opposed to centralized
• You can start today applying these architecture principles, no need for Big Bang.
• The future will belong to whom can master massively connected systems (SOA, Grids)

References

• http://www.opengroup.org/onlinepubs/007679899/toc.pdf
• www.ebpml.org
• http://groups.yahoo.com/group/India-egov/message/1010
• http://www.yared.com/
Part II

Business Process Modeling Notation (BPMN) and BPML / BPEL4WS

See also:
(BPMN standards)
http://www.bpmi.org
http://www.processwave.net/Articles/SoftwareProcess/BusinessModelingArticles.htm
http://cde.berkeley.edu/resources/bpmn/
(BPMN tools)
http://www.popkin.com/
http://www.popkin.com/customers/customer_service_center/demos/demos_with_overview_process_map.htm

Procedural Knowledge Road Map

<table>
<thead>
<tr>
<th>Type of knowledge/Levels</th>
<th>Conceptual model</th>
<th>Deployment methodology</th>
<th>Interoperability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process instance level</td>
<td>Process instance concept</td>
<td>BPMN+</td>
<td></td>
</tr>
<tr>
<td>Process definition level</td>
<td>Process definition concept, WfMC meta-model+</td>
<td></td>
<td>XPDL+</td>
</tr>
</tbody>
</table>

- BPMN - Business Process Modelling Notation, BPMI, 11.02
- XPDL - XML Process Definition Language, WfMC, 07.02
- Wf-XML - Workflow Interoperability Binding, WfMC, 11.01
ICONS’ conceptual models

• Process definition concepts - WfMC’s workflow meta-model extended with
  – time modeling
  – flexible Workflow Participant Assignment [Momotko2002]
• Process instance concepts
  – advanced process instance and activity instance behaviour models (timing and criticality behavior)
  – advanced time management

BPMN and XPDL extensions

• BPMN extension (BPMN+)
  – idea - standard well-known notation for both process definition adjusted to the needs for process instance
  – ICONS’ approach - a BPMN extension to visualise process execution
• XPDL extension (XPDL+)
  – performer relationships (WPA)
  – pre & post conditions
Process definition - an example

Possible process execution