Workflow Management

• Topics
  – Architectural Issues
  – Transactional Workflows
  – Cross-Organizational Workflows

• Readings
  – Lecture Notes
  – Workflow Management Coalition. Workflow Reference Model

What do WFMSs provide?

• With the introduction of WFMS a facility came up that allows for both:
  – Composing large distributed application systems out of smaller pieces which can be developed independently
  – Supporting real world business process concurrently performed by many different users exploiting various tools in a network

Architectural Issues

• Generic product structure
• Process definition
• Process definition meta model
• Client applications
• Invoked applications
• Interoperability
• Administration and monitoring

Source: Workflow Management Coalition [www.wfmc.org]
**Process Definition**

- Process Analysis
  - Modeling & Design Tools
  - Process Definition
  - Interchange Formats & APIs

- Process Structure
  - Activities & Navigation
  - Roles & Participants

- Application Definition
  - Application Invocation

---

**Process Definition Meta Model**

- Workflow Engine
- Command Set
- Interchange Formats & APIs

- Invoked Applications
- Application Specific Interfaces
  - (Local or Remote, Many Variants)

- Workflow Engine
- Workflow Enactment Service

- Connectors
- Direction

---

**Client Applications**

- Workflow Enactment Service
- Invoked Applications
- User Interface
- User Application
- Application Engine

---

**End User's View**

- User interacts with program windows
- Worklists show assigned activities
- User sees workflow and program windows

---

**Invoked Applications**

- Workflow Engine
- Application Specific Interfaces
- Invoked Applications

---

**Interoperability**

- Interface 4
- Activity state
  - Process execution context
- Application/Workflow execution context
  - Execution data transfer
- Task completion
  - Process definition and design
**WFMS – DBMS**

<table>
<thead>
<tr>
<th>Component</th>
<th>DBMS</th>
<th>WFMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meta-Model</td>
<td>(Relational DBMS) Table with rows and columns and SQL operations</td>
<td>No Industry Wide Standard</td>
</tr>
<tr>
<td>Build Time</td>
<td>Functions to define user specified constructs in terms of the meta model</td>
<td>Functions provided by DDL</td>
</tr>
<tr>
<td>Run Time</td>
<td>Functions provided by DML</td>
<td>Creating, navigating and controlling processes</td>
</tr>
<tr>
<td>Database</td>
<td>System catalogue and user defined tables</td>
<td>User-defined constructs (such as process models) and instances of these constructs (instances)</td>
</tr>
</tbody>
</table>

**WFMS 3 Tier System Structure**

- **Tier 1**
  - GUI
  - WFMS Client
  - WFMS Server

- **Tier 2**
  - DBMS Client

- **Tier 3**
  - DBMS Server
  - Database

**WFMS User Types**

- End Users
- Process Modellers (Business Analysts)
- Process Administrators
- System Administrators
- Customer Support
- External Users

**How Users Work with the WFMS**

- **GUI Variants**
  - WFMS propriety interface
  - Operating System Metaphors (Windows Explorer)
  - Desktop of another system (Lotus Notes)
  - Custom developed

**Working with Worklists**

- Pull Mode
- Push Mode
- Grab Mode

- Activity Functions:
  - Start
  - Restart
  - Re-execute
  - Finish
  - Suspend
  - Resume
  - Terminate
Working with Processes

- Start
- Suspend
- Resume
- Query
- Update

Working with Activities

- Create a workitem
- Reschedule
- Query
- Transactional actions (repair, commit, compensate, …)

Application Programming Interface

WFMS provides access to all data and functions via APIs
- Worklist API
- Operation API
- Administration API
- Process API
- Audit Trail API
- Buildtime API
- Container API

Workflows and Transactions

- In the context of transactions, workflow is a (high level) activity, that consists of a set of tasks with a well-defined precedence relationship between them
- The workflow (activity) is typically long-duration
- Workflow tasks (sub-transactions) are allowed to commit individually, permitting partial results to be visible outside the workflow
- Relaxation of ACID properties

Classification of Transactions

- Distribution
  - Transactions in Centralized DBMSs
  - Transactions in Distributed DBMSs
- Duration
  - Short-life transactions
  - Long-life / long duration transactions
- Processing
  - On-line / interactive transactions
  - Batch transactions
- Grouping of Operations
  - Flat transactions
  - Nested transactions

Nested Transactions

- Grouping of operations into hierarchical structures
- A set of sub-transactions that may recursively contain other sub-transactions

| Begin-transaction Reservation |
| Begin-transaction Airline |
| End. (Airline) |
| Begin-transaction Hotel |
| End. (Hotel) |
| Begin-transaction Car |
| End. (Car) |
| End |

Programming Language

Messaging Interface
Types of Nested Transactions

- **Closed Nested Transaction**
  - Sub-transaction begins after the root and finishes before
  - Commit of sub-transaction is conditional upon the commit of the root
  - Top-level atomicity

- **Open Nested Transactions**
  - Relaxation of top-level atomicity
  - Partial results of sub-transactions visible
    - Sagas
    - Split transactions

Advantages of NTM

- Higher level of concurrency
  - Objects can be released after sub-transaction
- Independent recovery of sub-transactions
  - Damage is limited to a smaller part, making it less costly to recover
- Creating new transactions from existing ones

Sagas: Open and Long-duration Transactional Model

- A collection of actions that form a long duration transaction
  - A collection of actions
  - A graph whose nodes are either actions, or one of {Abort, Complete} called the terminal nodes
  - An indication of the start called the start node

Concurrency Control in Sagas

Concurrency control is managed by two facilities
1. Each action “A”, is itself a (short) transaction
2. The overall transaction which can be any path from the start node to one of the terminal nodes
   - Uses the concept of Compensating transactions
   - A Compensating transaction “rolls back the effect of a committed action in a way that does not depend on what happened to the database between the time of the action and the time of the compensating transaction
   - If A is any action, A⁻¹ is its compensating transaction, and it is any sequence of legal actions and compensating transactions, then $A \alpha A^{-1} \alpha$ is a compensating action.

Workflow Transactions

Various concepts introduced to overcome problem of dealing with sub-transaction commit
- Compensating tasks
- Critical tasks
- Contingency tasks

Transactional Workflow

- Streaming
  - Support a run-and-gun environment
- Atomicity
  - Parts of a workflow once started must be completed
- Compensation
  - Implemented as work-item streams
    - Micro-script streams
    - Transaction streams
    - Work package streams

Over-loaded term!
Transactional Workflow

- Streaming
- Atomicity
- Compensation
  - Support all-or-nothing semantics
  - Encapsulation of re-usable (transactional) functions to form global transactions
  - Concept of atomic spheres
    - All activities transactional
    - All activities have same predecessor (or none)
    - All activities commit or all abort

Transactional Workflow

- Streaming
- Atomicity
- Compensation
  - Support a business-oriented unit of work
  - Consider non-transactional functions as well
  - Concept of compensation spheres
    - Invoke corresponding compensation activities in reverse order
    - Invoke compensation activity for the compensation sphere

Emerging Concepts

Virtual Organizations and E-Services
- An enterprise that outsources everything
- An entity composed of geographically dispersed workers (processes) who share their work and communicate only by electronic means
- Appears like a traditional enterprise to its customers, but the services and products it provides rely on the core business processes and resources of multiple constituent enterprises
- The member enterprises may be participating in a strategic alliance, or may collaborate only for the duration of one electronic commerce transaction

Collaborating Enterprises

Integration Technologies

- Business to Business (B2B)
  - or Application to Application (A2A)
  - or Enterprise Application Integration (EAI)

- Semantic Integration or Technical Integration

Basic B2B Integration Concepts

- Event
  - Occurrence of business data that is of interest
- Trading Partner
  - Organization that participates in event exchange
- B2B Protocol
  - Specification of event exchange between trading partners over a network
- Back end Application
  - Application within trading partner that is source/destination of events
- Process management
  - Definition and execution of business processes describing integration behaviour
Problem Space of B2B Integration

- B2B protocols
  - EDI, RosettaNet, Swift, …
- Network transport
  - SOAP, ebXML, EDIINT, …
- Security
  - Certificates, Authorization, Key Management
- Back end application integration
- Process management
  - Between and within trading partners

Cross-Organizational WF's

- Cross-organizational, but integrated solution for business process management
- (Public) workflow composition from pre-existing (private) component processes owned and developed by independent organizations
- Presents modelling challenges beyond stand-alone workflow systems

Process Interactions

- Chained
- Nested
- Joint Invocations
- Cross Synchronization

Process Interactions

- Chained
- Nested
- Joint Invocations
- Cross Synchronization
Process Interactions

- Chained
- Nested
- Joint Invocations
- Cross Synchronization

Heterogeneous WFMSs

- Homogeneous WFMSs
  - Support the same meta model
  - WFMSs can exchange data directly since no translation is required
- Heterogeneous WFMSs
  - Need at least a common denominator in terms of interfaces
  - Mapping of WFMSs meta models

Challenges

- Modelling
  - Composing global processes
- Execution
  - Interoperability between heterogeneous WFMSs
- New area, many open questions
- Interested research students welcome!