Workflow Technologies and Reference Models
Workflow Applications & Standards

Complexity of Workflow Systems

- **Administrative Workflows** involve repetitive, predictable processes with simple task coordination rules (such as routing an expense report or travel request through an authorization process).

- **Ad-Hoc Workflows** involve human coordination, collaboration, co-decision, and often appear in office processes (such as product documentation or sales proposal); address exceptions and unique situations.

- **Collaborative Workflows** tend to be dynamic and can involve a number of iterations of the same step until agreement is reached or going back to a previous stage.

- **Production Workflows** involve repetitive and predictable (critical) business processes (such as loan applications or insurance claims). Production workflow encompasses an information process involving access to one or more information systems.
Workflow Application Spectrum

- **Production**
  - Loan Origination
  - Insurance claims
  - Accounting
  - “Transaction workflow”

- **Collaborative**
  - Tech doc creation
  - Product brand mgmt.
  - Software development
  - “Process management”

- **Administrative**
  - Expense reports
  - Purchase approvals
  - Budgeting
  - “E - Forms”

- **Ad Hoc**
  - FYI Routing
  - Review and Approve
  - “Groupware”

**High Value Business processes**

**Low value processes**

**Repetitive process** → **Unique processes**

CS 565 - Lecture 8
Workflow Application Spectrum

- Low Complexity
  - Low Structure
    - Ad-Hoc
    - Collaborative
    - Production
  - High Structure
- High Complexity
  - Administrative
State of the Art

- Current workflow products typically
  - have client/server architecture
  - are web-enabled based on web services
- The products can provide support for a large percentage of all possible workflow (work coordination) applications that
  - are relatively simple, repetitive
  - predominantly require human involvement (user/manual tasks), such as office automation
What is missing?

- Support for workflow applications that are typically mission-critical and hence of higher value, and require better support for
  - existing/legacy applications,
  - error handling, automatic recovery
  - scalability
  - adaptive workflows, configurable and dynamic WfMS
  - support for coordination and collaboration
  - mobility
Enabling Technologies and Standards

- Maturing infrastructure:
  - e-mail (early 90s-already mature)
  - Workgroup(Notes), OLE, ActiveX (1993 - now mature)
  - Distributed Object Management (CORBA) (1995 (R) - 1998 (P))
  - TP Monitors (1996 (R))
  - Web, Java, Javascript, (1995 (R) - 1998 (P))
  - Agents (1996 (R))
  - Services (2000 (R/P))

R: research prototypes
P: products
Infrastructure Component Technologies

- internet-enabled (HTTP/HTML), network computing and Java-enabled
- distributed computing services: X.500, LDAP
- application interoperability: OLE/ActiveX, OpenDoc
- Queued/Persistent message systems
- data exchange: EDI
- database access: ODBC/JDBC
- Work group support/enabler: e.g., Lotus Notes
- Distributed computing infrastructure: DCE, CORBA (ORB only or with services), DCOM
- Web (HTTP/CGI), Agent (Servlet, Mobile Agents)
- Transaction processing: Client/Server (X/Open, TxRPC, MS Transaction Server, Encina), distributed transaction processing (Orbix-OTM)
Building Blocks

- **DBMSs as building block:**
  - Products: DBMS to manage/store routing information, status, data.
  - Research: Usually using active DBMS (triggers, rules)

- **Groupware as building block:**
  - Supports client-server application development
  - Supports and interoperates across several hardware platforms
  - Supports multiple network protocols
  - Comprehensive email application included
  - Support for enhanced email and document routing
  - Provides a consistent interface for all applications written for Notes
  - Ease of use for end users
  - Existing user base
Building Blocks

• Groupware provides a good infrastructure to build a WMS for Administrative and Ad-Hoc workflows

• Disadvantages/deficiencies
  - no support for transaction processing or ACID properties
  - lack of locking mechanism
  - latency in replication
  - no native support for integrating legacy applications
  - support only proprietary database formats
  - no real-time collaboration support
  - no shared screen support
Building Blocks

- Transaction Monitors as building block:

- Advantages of second generation TP-monitors (e.g., Encina)
  - Ease of implementing fault tolerance: Transactional-RPCs, utilities such as Queue Managers or Structured File systems
  - Persistent messaging (message queues)

- Disadvantages:
  - Lack of interoperability between monitors
  - Products still lack good performance, stability, ease of use / administration
  - Problems with support for state persistence and multi-threading
Building Blocks

- The Web as building block
  - Standard user interface on heterogeneous platforms
  - HTTP protocol layer provides a well-defined communication medium across organizations
  - Multiple web servers can allow inter-enterprise wide cooperation; HTML forms, CGI scripts, DHTML used to interact with user and application systems
  - Relatively inexpensive infrastructure that can be shared for many applications
  - Problems: currently, limited support for distribution, error handling, reliability; unpredictable performance
  - Promises: recent and fast evolving support for direct database access, authentication, security, transaction support, ...
**CORBA**

- **CORBA** (Common Object Request Broker Architecture): a standard to define the architecture of an Object Request Broker (ORB). Proposed/promoted by OMG.
- Object Management Architecture:
  - **ORB**: provide the mechanisms by which objects **transparently** make requests and receive responses.
  - **Object Services**: Collection of system level services for adding functionality to ORB: e.g., Naming, Transaction, Event Handling, Security, etc.
CORBA as building block

Infrastructure for developing distributed enterprise-wide applications that are *robust, reusable, portable, interoperable*

- Distribution across heterogeneous hardware and software platforms
- Implementation Transparency: Unified high-level interface (IDL) for interactions between heterogeneous tasks
- Location transparency: allows decoupling of task and data location
CORBA as building block

- Integration of legacy application/information systems by \textit{wrapping} them as CORBA objects
- Built-in constructs for Exception handling
- Fault-tolerant communication paradigm (ORB):
  - automatic server start, rerouting of messages
- Suitable platform for incorporating recovery features into the workflow management system
- \textit{Object Services}: useful services for developing transactional workflows in a heterogeneous & distributed environment
CORBA as building block

CORBA makes it (relatively) easy to support

- **Error Handling**
  - support for system exceptions and user exceptions (variable level of support by different products)

- **Reliability/Fault tolerance**
  - Varied support, outside CORBA: ORBeline’s smart agent tracks all object & clients, notifies failures, reroutes lost data

- **Security**
  - Permission for every object; additional services (access permissions as user-programmed filters or at method invocation)
Workflow Management Standards

• Why standards?
  ■ for protecting investment in workflow software in the process of enterprise re-organization
  ■ for ensuring that essential criteria can be met and risk can be reduced
  ■ for enabling interoperability with intra- or inter-organization systems in order to support wide-area workflow management and distributed business processes

• Standardization Efforts in Workflow Management
  ■ Workflow Management Coalition (http://www.wfmc.org)
  ■ Object Management Group (http://www.omg.org)
The Workflow Management Coalition (WFMC)

- Non-profit international organization of workflow vendors, users, analysts and research groups
- Founded in 1993
- Mission: “...to promote and develop the use of workflow through the establishment of standards for software terminology, interoperability and connectivity between workflow products.”
- Achievements:
  - Workflow reference model
  - Workflow API (WAPI) specification
  - Workflow Interoperability specification
Workflow Reference Model (WRM)

- **High-level functional view**: WFMSs provide support for
  - **Build-time** functions (defining and modeling workflow process and constituent activities)
  - **Run-time control** functions (managing workflow process in an operational environment)
  - **Run-time interactions** (human users or IT applications and tools)
- **The Distribution view**: 
  - distributing work (from workgroups to inter-organization groups)
  - communication mechanisms and platforms supporting distributed interoperable runtime infrastructure
WFMS Functional View

- Business Process Analysis, Modeling & Definition
- Process Definition
- Workflow Enactment Service
- Applications & IT Tools
- Interactions with Users & Application Tools

Process Design & Definition
Build Time
Run Time
Process Instantiation & Control

CS 565 - Lecture 8 29/3/17
WFMS Distribution View

User Interface & Local Applications

Process Mgmt & Distribution Fns

Applications & IT Tools

Databases

Individual activity steps

BP
WFMC Interfaces

- WFMC defines:
  - Specification for process definition data and its interchange
  - Interfaces to support interoperability between different workflow systems
  - Interfaces to support interaction with a variety of IT application types
  - Interface to support interaction with user interface desktop functions
  - Interfaces to support monitoring functions for managing composite applications
WFMS Interfaces

Definition Tool

Process Definition

WF Control Data

Applications

Workflow Application Data

Workflow Enactment

Process Model

WFM Engines

Work List

Worklist Handler

User Interface

Applications

Workflow Enactment

Process Definition Tool
Design Decisions

- **Centralized vs distributed workflow enactment service**
  - Workflow engines are responsible for managing part of or all the execution of individual process instances.
  - In a centralized system, a single engine manages all process execution.
  - In a distributed system, several engines cooperate, each managing part of a process.

- **Worklist handler location and distribution mechanism**
  - Worklist handler manages the interaction between workflow participants and the workflow enactment service.
  - Controls work allocation (load balancing) and reassignment.
Workflow Enactment

Centralized Model

Process Definition

Supervisor

Distributed Model

WFM Engine

Workflow Control Data

Worklist

Invoked

Applications

WFM Engine

Workflow Control Data

Worklist

29/3/17
Workflow Enactment

- WFMC defines the **boundary** around the workflow enactment service; *functional components are accessible via a set of common APIs*
- Internal mechanisms for providing the functionality are not defined by WFMC; may include one or more workflow engines
- **Interoperation** between different enactment services is enabled by defining appropriate interfaces; composite (multi-vendor) workflow applications may execute parts of a process on different engines
Workflow Client Applications

- Worklists (i.e., queues of work items) provide for the interaction between workflow engines and worklist handlers.
- Several alternatives exist:
  - **host-based** model: worklist handler is at the host and communicates locally with the worklist at the workflow engine.
  - **shared-filestore** model: worklist handler is a client application communicating through a shared filestore that is accessible to both the engine and the client.
  - **e-mail** model: worklist at the client; communication for distributing work items through e-mail.
  - **procedure-call** or **message-passing** model: communication via RPC; worklist at the engine or worklist handler.
Worklist Handler Implementations

WFM Engine

Server Environment
- Local call
- Worklist access

Common API
- Workflow Client Appn

Host Based

Shared Filestore

Worklist

Procedure call or Message Passing

Workflow Client Appn

Client Environment
- RPC

WFM Engine

Worklist

X.400 etc

WFM Engine

Worklist

WFM Engine

Worklist

29/3/17
Workflow Reference Model

Focuses on a standardized set of interfaces and data interchange formats between generic components.

- Process Definition Tools
- Administration & Monitoring Tools
- Workflow API and Interchange formats
- Workflow Client Applications
- Invoked Applications
- Other Workflow Enactment Service(s)
- Workflow Engine(s)
Workflow Reference Model

- **WAPI** (Workflow APIs and Interchange Formats) comprises a set of constructs by which the services of the workflow system may be accessed.

- **Definition**: “The Workflow Enactment Service is a software service that may consist of one or more workflow engines in order to create, manage and execute workflow instances. Applications may interface to this service via the workflow application programming interface (WAPI)”

- Interaction with external resources occurs via two interfaces:
  - Client application interface (interface 2)
  - Invoked application interface (interface 3)
Workflow Reference Model

- **Client application interface**: the interface through which a workflow engine interacts with a worklist handler
  - Responsible for organizing work on behalf of user resources
  - Worklist handler selects and progresses individual work items from the work list
  - The use of application tools may be under the control of the worklist handler or the user

- **Invoked application interface**: enables workflow engine to activate server-based applications
  - If user interaction is needed, then it would be invoked via the worklist interface
Workflow Reference Model

- A **standardized interchange between workflow engines** is necessary for enabling the use of heterogeneous systems (interface 4)
  - Enactment service may transfer activities or sub-processes to another workflow enactment system for execution (workflow engine interchange)
- Administration and monitoring functions are realized through another common interface (interface 5)
Workflow Reference Model

- **Definition:** “A workflow engine is a software service that provides the run-time execution environment for a workflow instance”

- Workflow engine provides facilities for:
  - interpreting process definitions
  - controlling process instances (creation, activation, suspension, termination etc.)
  - navigation through process activities (seq./ parallel activities, scheduling etc.)
  - identification of work items for user attention
  - maintenance of workflow control data and workflow relevant data
  - invoking external applications
  - supervisory actions for control, administration and audit purposes
Workflow Reference Model

Homogeneous / heterogeneous workflow enactment

- **Homogeneous** workflow enactment involves one or more compatible workflow engines that provide the run-time environment
- A **heterogeneous** enactment service comprises two or more services that adhere to common interoperability standards
- **Conformance levels:**
  - common naming scheme across the heterogeneous domain
  - common process definition objects and attributes
  - workflow relevant data transfer
  - process / activity transfer between engines
  - common administration and monitoring functions
Workflow Reference Model

- Workflow enactment service can be viewed as a state-transition machine
  - Individual processes or activities change states in response to external events or control functions

![Workflow Diagram]

- Initiated
- Running
- Suspended
- Active
- Complete
- Terminated
- Iterate
- Initiate
- Start
- ReInitiate
- Suspend
- Resume
- Terminate / Abort
- Iterate
Workflow Reference Model

- Basic states:
  - initiated: process instance created but conditions to start its execution have not been fulfilled yet
  - running: process has started execution; any of its activities may be started (provided appropriate conditions are met)
  - active: one or more activities have been started (work items have been created)
  - suspended: no activity takes place until it returns to the running state
  - completed: conditions for completion have been fulfilled; process instance is erased
  - terminated: execution has been stopped before normal completion
Workflow Reference Model

- Workflow Application Programming Interface and Interchange
  - comprises 5 interfaces
  - some functions are common

- Data Interchange:
  - Definition: “Workflow Control data are internal data managed by the workflow management system and/or workflow engine”
    - not interchangeable via WAPI commands
  - Definition: “Workflow Relevant data are data used by the workflow management system to determine state transitions in a workflow process instance”
  - Definition: “Workflow Application data are application-specific and not accessible by the workflow management system”
Workflow Reference Model

- Interchange of workflow relevant data and application data is required across the WAPI to support functions of:
  - the worklist handler (interface 2)
  - invoked applications (interface 3)
  - workflow engine interchange (interface 4)
- WAPI command set includes calls to accept or return workflow relevant data from or to the enactment service across the WAPI
- Interchange can be direct (e.g., in e-mail based systems) or application-driven
- Conversions may be required (e.g., X.400 or MIME)
Workflow Reference Model

- Interface 1 - Process Definition Interchange
  - Meta Model
    - describes a meta-model that can be used to express objects, relationships and their attributes
    - Workflow Process Definition Language (WPDL)
      - WPDL is lacking constructs for: procedures, time modeling, events, organizational modeling
  - API calls
    - part of WAPI between workflow systems or between workflow systems and process definition tools
Workflow Reference Model

- Process Analysis
- Modelling & Definition
- Tools

- Process Definition
- File and API options

- Workflow Enactment Service
  - Workflow
  - Engine(s)

- Process Structure
- Activities & Navigation
- Roles & Participants
- Trigger Conditions
- Application
- Invocation
<WPDL File> ::= 
  WORKFLOW <process name> 
    WPDL_VERSION <wpdl version> 
    VERSION <source vendor id> 
    CREATED <creation date> 
    . 
    . 
    [Activity List] 
    [Transition Information List] 
    [Workflow Participant List] 
    [Workflow Application List] 
    [Workflow Process Relevant Data List] 
  END_WORKFLOW 
  [<WPDL>]
Workflow Reference Model

- Interface 2: Client Application Interface

Workflow Enactment Service

Standardized API’s & Interchange

Workflow Management

Client App

Invoked Apps

User Interface

Command Set
- Connect/Disconnect
- Process/Activity Control fn
- Process Status fns
- Worklist Manipulation commands

29/3/17
Workflow Reference Model

- Interface 3: Invoked Application Interface

![Workflow Reference Model Diagram]

CS 565 - Lecture 8
Workflow Reference Model

- Interface 4: WAPI Interoperability Functions

- Activity or sub-process invocation
- Process/Activity status/control
- Application/workflow relevant data transfer
- Synch-point coordination
- Process definition read/write
Workflow Reference Model

- Interface 5: Systems Administration & Monitoring Interface
Comments

- “Reference architecture”
- No dedicated component for administration of workflow-types
- No precise definition of semantics of interfaces
- “Glossary”- based on compromise
- High granularity of application-integration
- No consideration of many OMG design principles
- No provision taken for transactional workflows
- … but the recommendations have been adopted by many workflow vendors
The OMG Approach

- no need for an engine, workflows are just objects
- support for greater variety of workflow implementations
  - script-based
  - binary workflow objects
- advantages
  - simplicity
  - no architectural “break”
  - seamless integration of arbitrary business objects
  - fully exploit adopted and envisioned CORBA services and CORBA facilities

The Common Object Request Broker Architecture (CORBA) is a standard defined by the Object Management Group (OMG) designed to facilitate the communication of systems that are deployed on diverse platforms.
The OMG Architecture

Object Request Broker (ORB)

CORBAfacilities
- User Interface
- Systems Mgt.
- Information Mgt.
- Meta Object
- Task Management
- Rule Management
- Agent Facility
- Automation
- Workflow Management

CORBAservices

App. Interf.

Dom. Interf.
OMG-based WFMS

• **ORBWork**: Fully distributed ORB-based Workflow Enactment System for METEOR$_2$
  http://lsdis.cs.uga.edu/workflow

• **WorCOS**: Fully distributed, modular workflow management system using CORBA services
  http://wwwdb.inf.tu-dresden.de/WORCOS

• **Reliable Workflow System**: Workflow management on top of an ORB and the Arjuna transaction system
  http://arjuna.ncl.ac.uk/WorkflowSystem/index.html

More on OMG activities and standards at http://www.omg.org
Updated/Modern Reference Model

- Embraces main strengths of WFMC’s Workflow Reference Model & remedies the drawbacks

Main Strengths/Drawbacks of WFMC Workflow Reference Model:

- **Separation of abstract from implementation view of a BP**
  - Concrete bindings to each abstract interface were defined to specific technologies that can deliver respective functionality
    - Initially C, then IDL & CORBA and finally Web Services & XML

- **Lifecycle view of BP**
  - Three broad phases joined up (overall consistency + common representation model): model & define, operational implementation and analyze & improve
    - Change supported via process repository
    - Audit and analysis tools operate on common audit data specification
Updated/Modern Reference Model

- Main Strengths/Drawbacks of WFMC Workflow Reference Model:
  - Lifecycle view of BP
    - Separate query & audit functionality from analysis & reporting tools
    - More emphasis should be put on process fragments & their consolidation to support dynamic BPs
    - Develop something beyond primitive choreography capability to support process fragment interactions
  - 3 main dimensions: process, information & organisation interlinked
    - Takes a simplistic view on information dimension
      - Recognized only 3 data classes: workflow control, workflow relevant & application data
      - Weak in information marshalling within BP
        - Define activity attributes to determine incoming/outgoing information flow
Main Strengths/Drawbacks of WFMC Workflow Reference Model:

- **Data co-ordination & recovery** somehow considered
  - Two-phase commitment and/or transaction compensation can be used
  - Just a concept of exception transitions was defined to allow user-defined failure handling or compensation activities
  - No standards were proposed for their specification while at least one mechanism supported by respective WFMSs
Updated/Modern Reference Model

- Functional BPM component model to enable product interoperability
Updated/Modern Reference Model
Updated/Modern Reference Model

- **Internal Process Definition:**
  - Use of standards to integrate different sw tools
  - Not all aspects of internal process behaviour need to be standardized or made visible to external boundaries

- **External Process Definition:**
  - Interoperability covering specification of permitted interactions between different BPMSs
  - Upper layers concerned with standards to support E2E process modelling & choreography
  - Lower layers define semantics of process interoperability
    - Abstract service definition for WfMC interface 4 & different interoperability models
    - Standards defining service end-points & data formats in interactions are below
    - Significant overlap in standards particularly in choreography area
External Process Execution:

- Standards stack for process execution
  - Standards for discovery of external interoperability services
  - Standards for process schemas to support interoperability
  - Standards to support runtime process interoperability

- Standard for true process interoperability is Wf-XML

- Different forms of messaging protocols (e.g., in different vertical industries like HL7 for healthcare) could be integrated if clearly linked to process semantics
  - Need to address process id handle for asynchronous interactions to allow for queries on remote process state or actioning dynamic change in process execution
    - A generic process interoperability protocol could be used to carry industry-specific context data payloads
Internal Process Execution:

- Standards provide a common framework to support workflow execution and a basic level of functionality to achieve a common interpretation.

- Highest-level: model of process & activity notation to underpin most execution activities, including API functions, audit data collection and query state:
  - Notation allows local extension of basic WfMC states.
  - All other standards use these state models covering:
    - Audit data collection related to various state change events, including internal resource assignments.
    - Process or activity status query.
    - APIs for consistent access to BPM functionality from client applications to query or set process, activity or worklist control data.
  - Subset of information might be made visible to external boundaries of process execution environment to enable visibility on internal states and audit data in a defined manner in cooperating external systems.
Recommended Reading

- “Workflow Management: Models, methods and systems” by van der Aalst and van Hee
- http://www wfmc org/resources -> XPDL, Wf-XML, and Interface specifications of WFMC