Web Services
Coordination and Transactions

Kapitel 11
Workflows und Web Services

Motivation

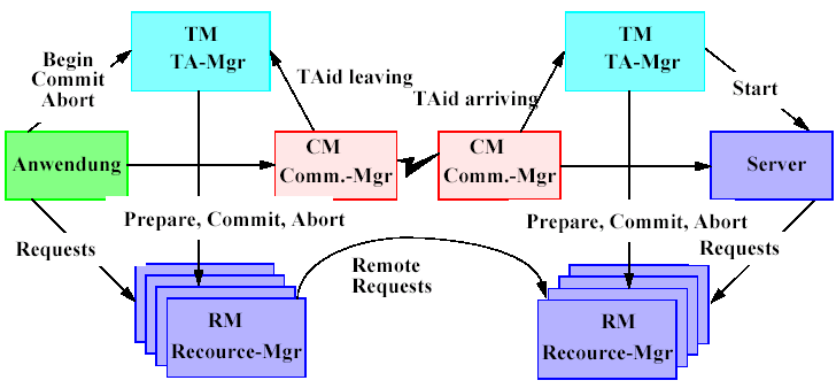
FACHBEREICH INFORMATIK

- Need to coordinate (sets of) activities or applications
  - Distributed
  - Running on different platforms using local coordinators
- Example
  - Reach consistent agreement on the outcome of distributed transactions
    - Atomic transactions
    - Business transactions/activities
- Web Services Coordination
  - Provides framework that enables an application service to
    - create a context needed to propagate an activity to other services and to
    - register for coordination protocols
  - Enables existing transaction processing, workflow, and other systems for coordination to hide their proprietary protocols or mechanisms and to operate in a heterogeneous environment
- Web Services Transaction
  - Uses WS-Coordination framework to define coordination types for
    - Atomic Transaction (AT)
    - Business Activity (BA)

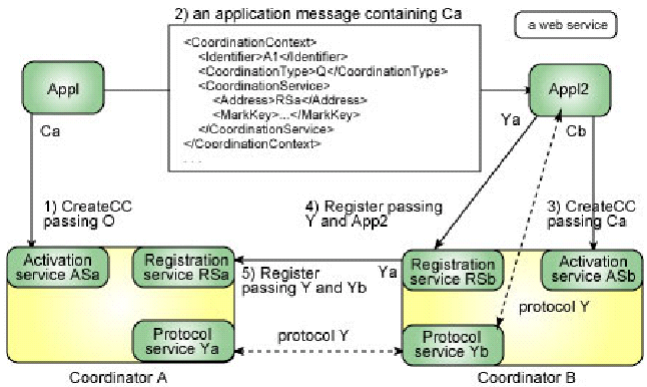
WS-Coordination

- The Model:
  - Coordination service (coordinator) consists of
    - Activation service
      - Used to create coordination context
    - Registration service
      - Enable application to register for coordination protocols
    - (set of) coordination protocols
      - Specific to coordination type
  - Additional coordinators can be interposed
    - Subordinate coordinators
  - Based on web services
- Coordination context
  - Used to exchange coordination information among different parties
    - Placed within messages exchanged between parties
- Extensibility
  - Publication of new coordination protocols
  - Definition of extension elements that can be added to protocols and messages
    - Exchange application-specific data on top of defined message flows

X/Open DTP revisited …



Exchanging Coordination Information



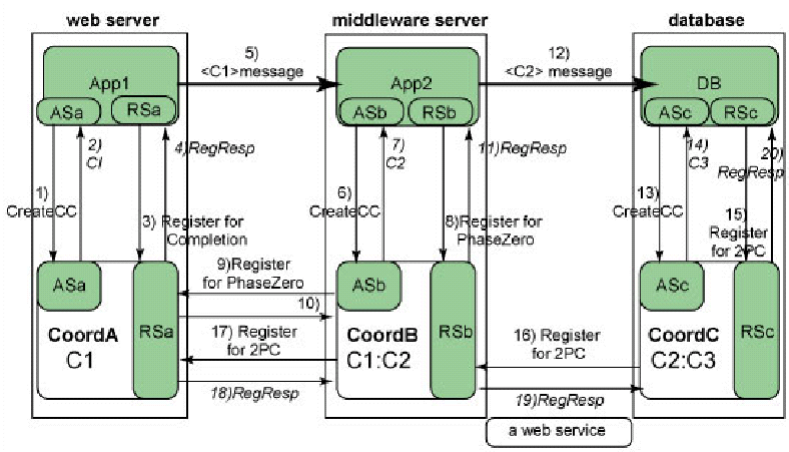
WS-Transaction – Atomic Transactions

- Atomic Transactions (TA) coordination type
  - Defines type-specific commit protocols
    - Completion: A participant (app creating the TA) registers so that it can tell the coordinator when/how to complete the TA (commit/abort)
    - CompletionWithAck: same as 'Completion', but coordinator must remember the outcome until receiving an acknowledgement
    - 2PC: a resource manager (RM) registers for this protocol to be included in the commit/abort decision
      - Hierarchical 2PC (local coordinators can be interposed as subordinate coordinators)
    - PhaseZero: a participant wants to be notified by the coordinator just before the 2PC begins
      - Example: participant caches, needs to communicate changes on cached data to DBMS before TA commits
    - OutcomeNotification: participant wants to be notified of TA outcome
  - Extension elements
    - Example: communicate isolation levels

Atomic Transaction – Example

```
<?xml version="1.0" encoding="utf-8"?>
   <S:Envelope xmlns:S="http://www.w3.org/2001/12/soap-envelope“
      <S:Header>
         . . .
         <wscoor:CoordinationContext
                  xmlns:wscoor=http://schemas.xmlsoap.org/ws/2002/08/wscoor
                  xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility"
                  xmlns:myApp="http://www.w3.org/2002/08/myApp">
            <wsu:Identifier>http://foobaz.com/SS/1234</wsu:Identifier>
            <wsu:Expires>2002-08-31T13:20:00-05:00</wsu:Expires>
            <wscoor:CoordinationType>
                  http://schemas.xmlsoap.org/ws/2002/08/wstx
            </wscoor:CoordinationType>
            <wscoor:RegistrationService>
                  <wsu:Address>
                        http://myservice.com/mycoordinationservice/registration
                  </wsu:Address>
                  <myApp:BetaMark> ... </myApp:BetaMark>
                  <myApp:EBDCode> ... </myApp:EBDCode>
            </wscoor:RegistrationService>
            <myApp:IsolationLevel>
                  RepeatableRead
            </myApp:IsolationLevel>
         </wscoor:CoordinationContext>
         . . .
      </S:Header>
      . . .
   </S:Envelope>
```


AT WS-Coordination Flow



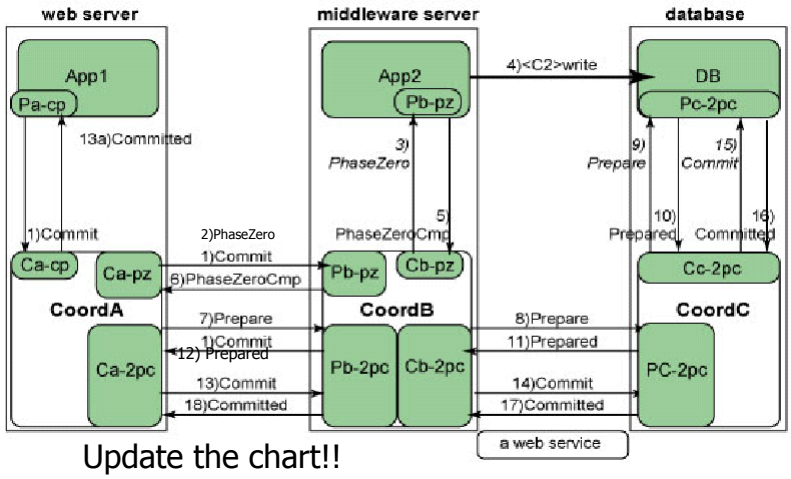
AT WS-Coordination Flow (cont.)

- App1:
  - sends a CreateCoordinationContext message (1) to its local coordinator's Activation service ASa
    - create an atomic transaction T1
    - gets back in a CreateCoordinationContextResponse message (2) a CoordinationContext C1 containing the transaction identifier T1, the atomic transaction coordination type and CoordA's Coordination PortReference RSa
  - sends a Register message (3) to RSa to register for the Completion protocol
    - gets back a RegisterResponse message (4), exchanging protocol service PortReferences for the coordinator and participant sides of the two-way protocol
  - sends an application message to App2 (5)
    - propagating the CoordinationContext C1 as a header in the message.
- App2:
  - decides to interpose local coordinator CoordB in front of CoordA
    - acts as a proxy to CoordA for App2
    - CoordA is the superior and CoordB is the subordinate
  - does this by sending a CreateCoordinationContext message (6) to the Activation service of CoordB (ASb) with C1 as input
    - getting back (7) a new CoordinationContext C2 that contains the same transaction identifier (T1) and coordination type, but has CoordB's Coordination PortReference RSb.
  - registers with CoordB for the PhaseZero protocol (8 and 11)
    - CoordB registers with CoordA for the PhaseZero protocol (9 and 10)
  - sends a message to DB (12), propagating CoordinationContext C2

AT WS-Coordination Flow (cont.)

- DB:
  - decides to interpose its local coordinator CoordC by sending a CreateCoordinationContext message (13), further extending the superior-subordinate chain
    - gets back (14) a new CoordinationContext C3 that contains the same transaction identifier (T1) and coordination type, but CoordC's Registration service PortReference RSc
  - registers with CoordC for the 2PC protocol because it is a resource manager (15 and 20)
  - causes CoordC to register with CoordB for the 2PC protocol (16 and 19)
  - causes CoordB to register with CoordA for the 2PC protocol (17 and 18)

AT Coordination Protocol Flows



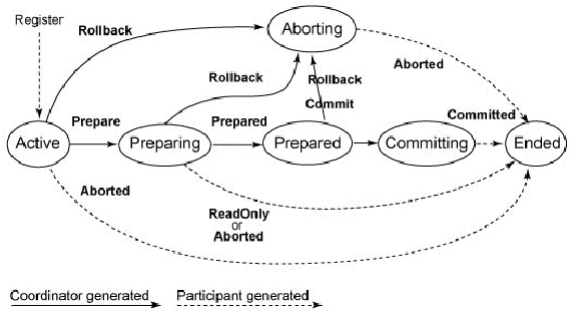
Update the chart!!

AT Coordination Protocol Flows (cont.)

- App1:
  - tries to commit the transaction using the Completion protocol (1)
- CoordA executes **PhaseZero protocol**
  - has 1 participant registered for PhaseZero (CoordB), sends a PhaseZero message (2) to CoordB's PhaseZero Participant protocol service Pb-pz
  - CoordB relays PhaseZero message to App2 (3)
  - App2 sends its cached updates to DB
    - application message (4) propagates the CoordinationContext C2
    - sends a PhaseZeroCompleted message (5) to CoordB
- CoordA begins the **2PC protocol**
  - sends a Prepare message (7) to CoordB's 2PC Participant protocol service Pb-2pc
  - CoordB sends Prepare message (8) to CoordC's 2PC Participant protocol service Pc-2pc
  - CoordC tells DB to prepare (9)
- CoordA commits
  - sends Commit message (13) to CoordB
    - Committed notification to App1 (13a) can also be sent
  - CoordB sends Commit message (14) to CoordC
  - CoordC tells DB to commit T1
    - DB receives the Commit message (15) and commits
  - Committed message returns (16, 17 and 18)

AT – 2PC Protocol

- Two-way protocol
  - Exchange of messages between coordinator and participant
- State Diagram
  - State reflects common knowledge of both parties



AT – 2PC Protocol (cont.)

- OnePhaseCommit
  - If only one participant has registered for 2PC, the commit/abort decision can be delegated to that participant
    - Send OnePhaseCommit message instead of Prepare message
  - Can be recursively applied by subordinate coordinator
- "Presumed abort" assumption
  - No knowledge of a transaction implies it is aborted
  - Allows for optimizations during commit phase
- "Read-only" optimization
  - After receiving a prepare message from the coordinator, participant can reply with a read-only message and skip the second phase
- Replay Message
  - Used by participant to solicit transaction outcome from coordinator after a failure

WS-Transaction – Business Activities

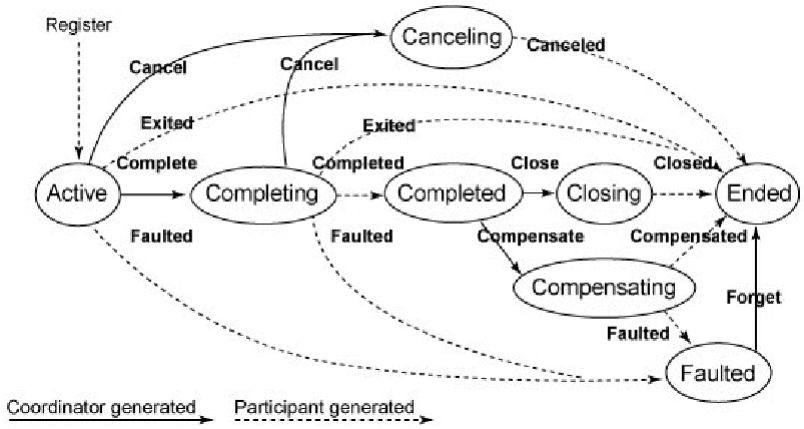
- Characteristics
  - Usually long-running
    - Responding to a request may take a long time
  - May consume lots of resources perform a lot of work
    - Loss of state of business activity cannot be tolerated
    - Forward recovery
- Design points
  - State transitions need to be reliably recorded
  - All request messages are acknowledged
    - Detect problems early
  - Response to a request is a separate operation
    - Not the output of the request
    - Avoid problems with timeouts of message I/O implementations

Business Activity (cont.)

- Business Activity (BA) coordination type
  - Create business activity, propagate coordination context
  - Interpose a coordinator as a subordinate
  - Create business scopes (see BPEL)
    - Can be nested
  - Register for participation in BA
    - BusinessAgreement protocol
      - Nested scope participant registers with parent scope coordinator
      - Parent scope can manage it
      - Nested scope must know when it has completed all the work for a business activity
    - BusinessAgreementWithComplete protocol
      - Nested scope relies on parent to tell it when it has received all requests for work in the business activity

Business Agreement Protocol

- BusinessAgreementWithComplete – State Diagram



Business Agreement Protocol (cont.)

- Parent sends application message to a child
  - Contains a business CoordinationContext
- Child registers with parent as participant of the business activity (Active state)
- Parent tells child when it has received all requests by sending the complete message (Completing state)
- Child finishes
  - Case 1: no more participation required (read-only, irreversible, …)
    - Child sends exit message (Ended state)
  - Case 2: continue participation
    - Completed: requires ability to compensate completed tasks
    - Child lives on until parent sends close or compensate message
  - Case 3: child fails while active (or compensating)
    - Send faulted message to parent
    - Parent replies with forget message
- Parent tells child to cancel
  - Child needs to abandon its work in "some appropriate way"