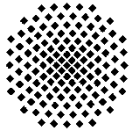
A Model-Driven Approach to Implementing Coordination Protocols in BPEL

Oliver Kopp, Branimir Wetzstein, Ralph Mietzner, Stefan Pottinger, Dimka Karastoyanova, Frank Leymann

Institute of Architecture of Application Systems



Universität Stuttgart
Germany

kopp@iaas.uni-stuttgart.de

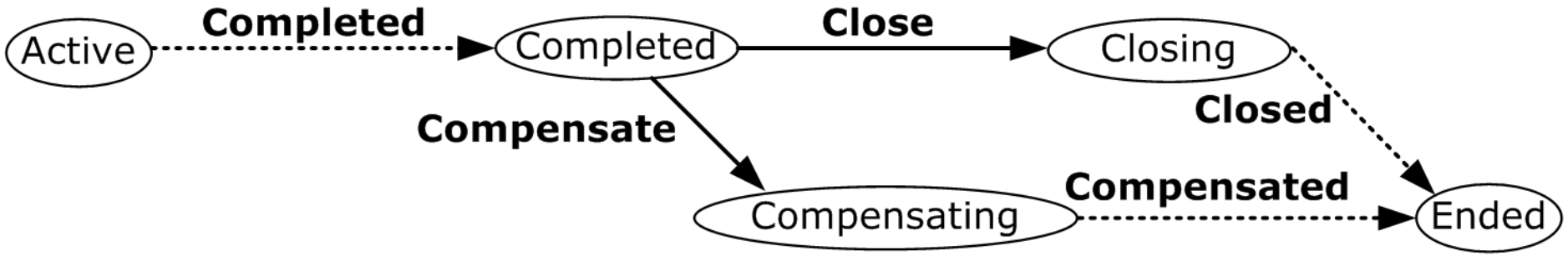
Agenda

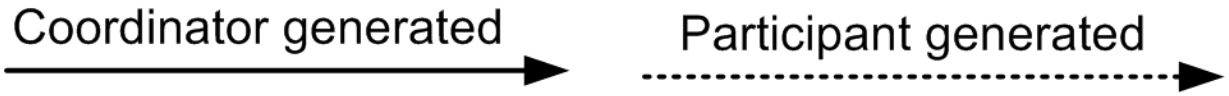
- Background
  - Coordination in a Web Service World
  - Transaction Research Roadmap of the IAAS
- The Approach
- Conclusion and Outlook

Modeling Coordination Protocols

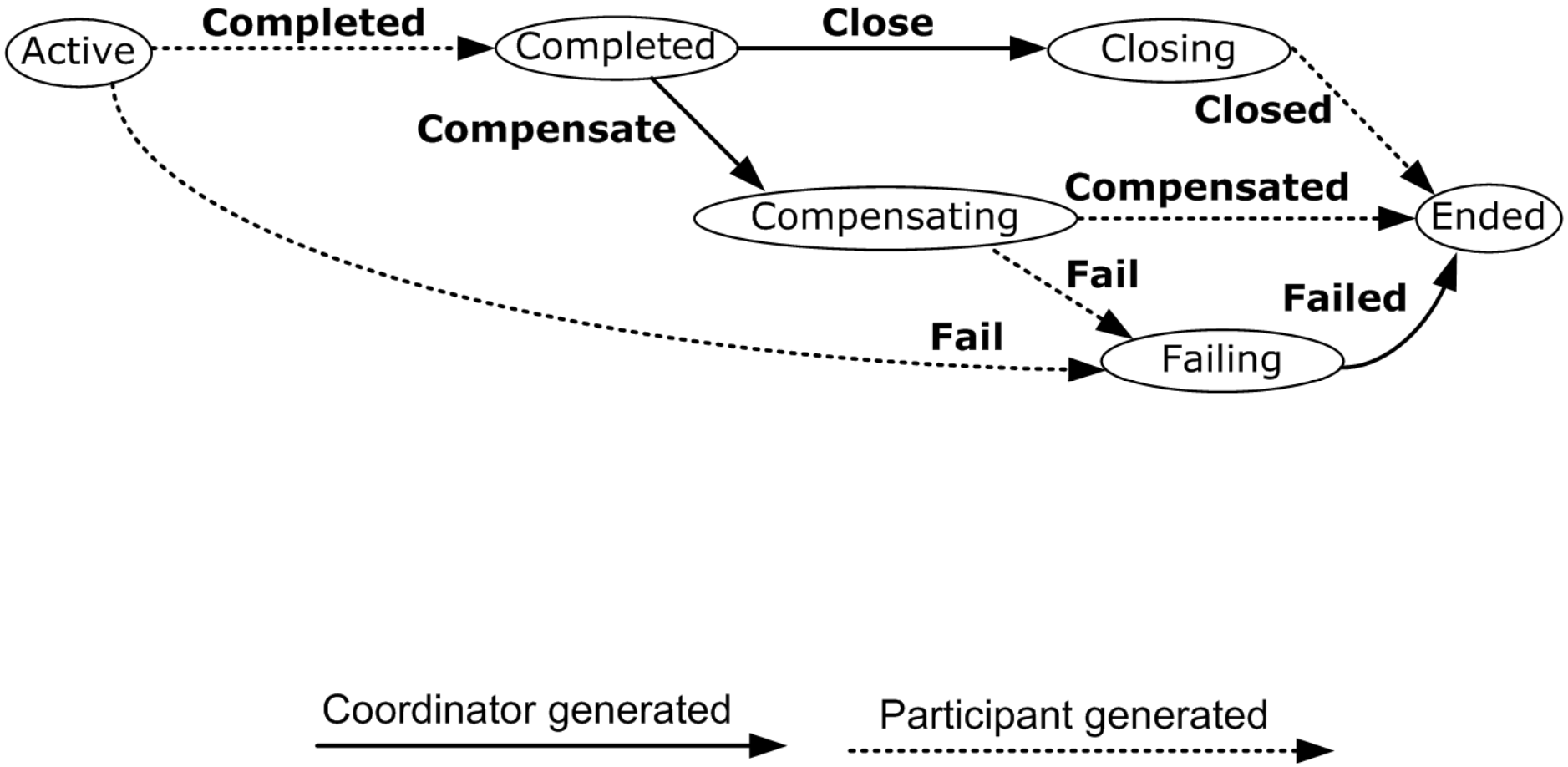
Modeling Coordination Protocols – 1/4

- Describe states and transitions of one participant

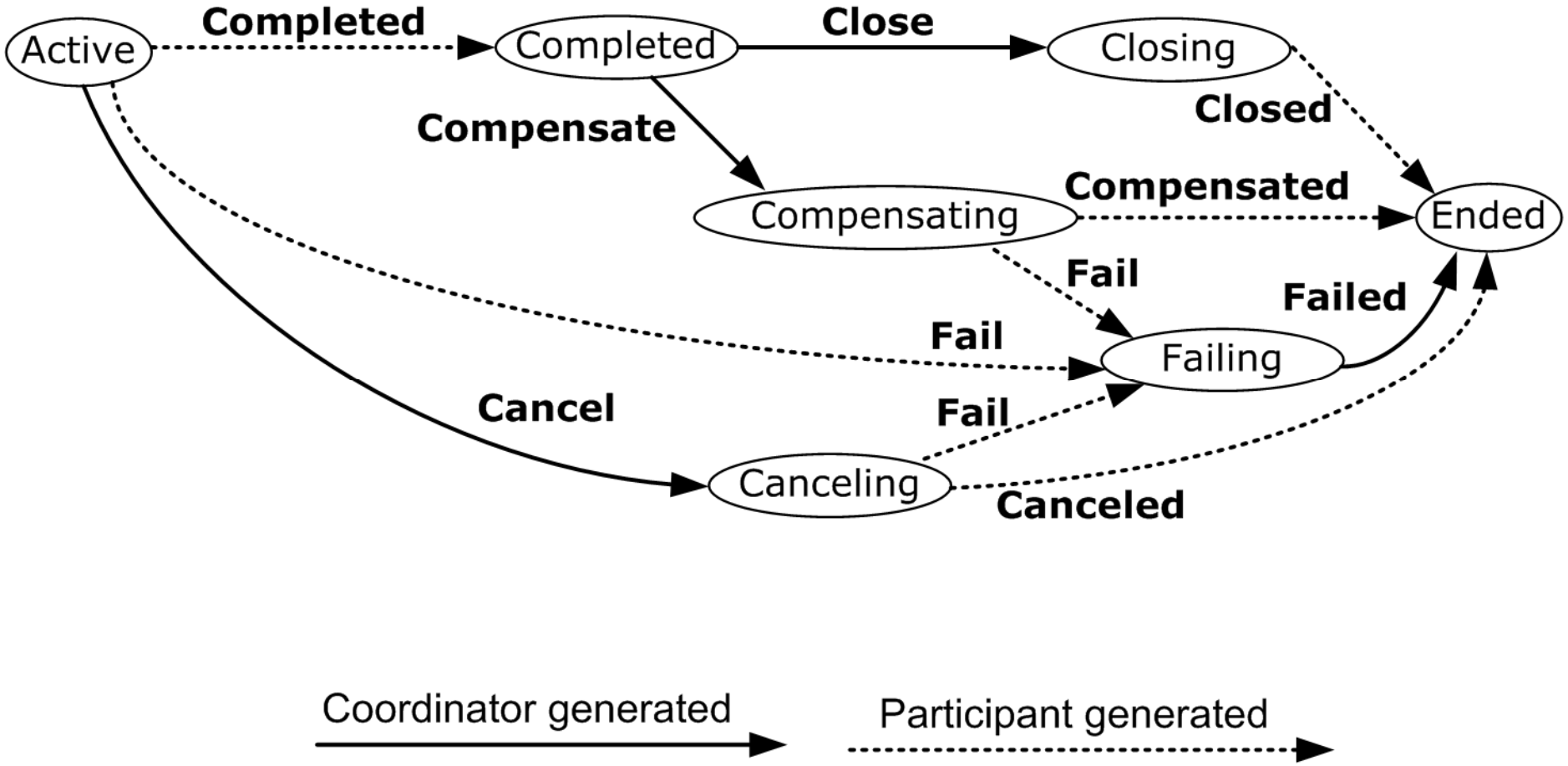




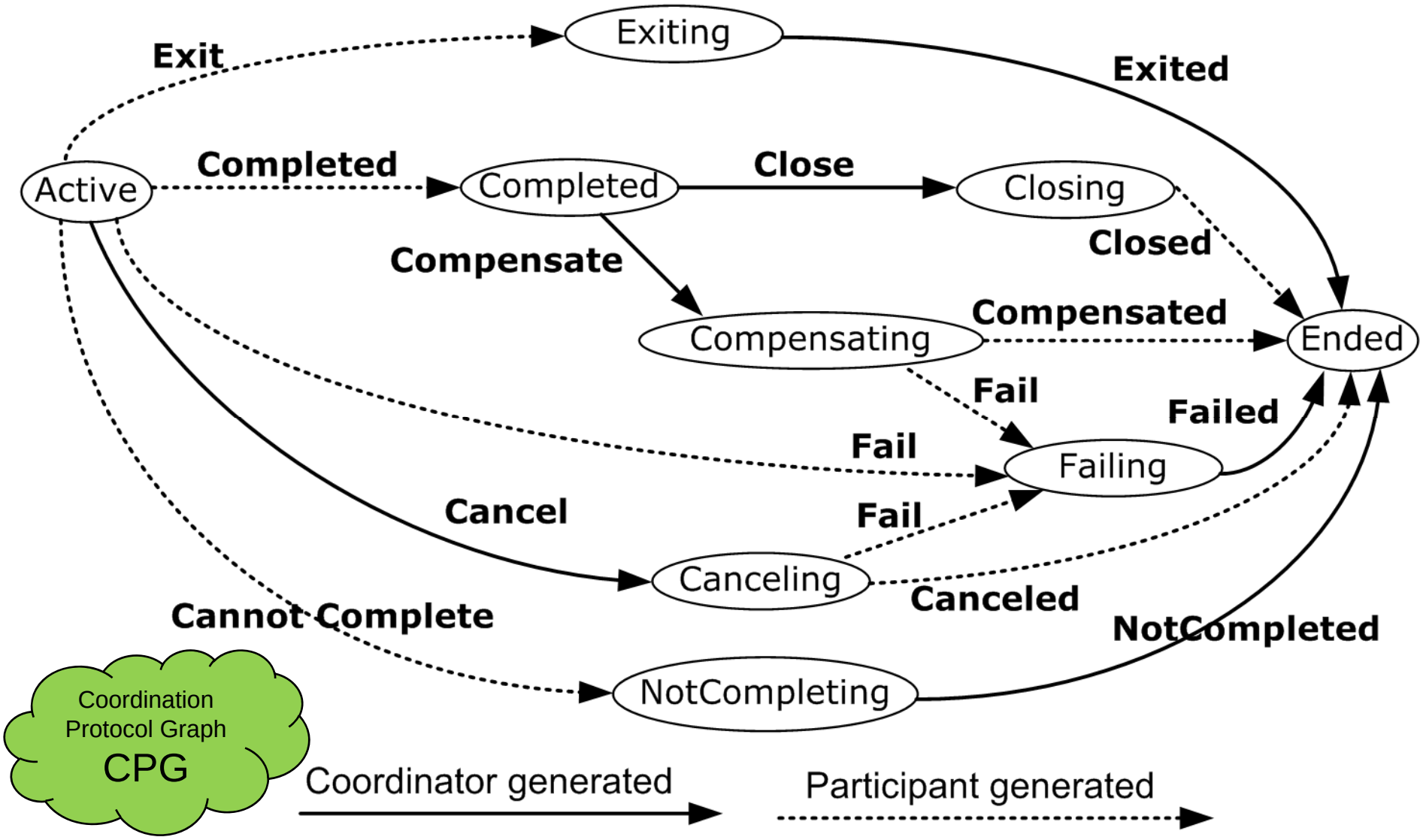
Modeling Coordination Protocols – 2/4



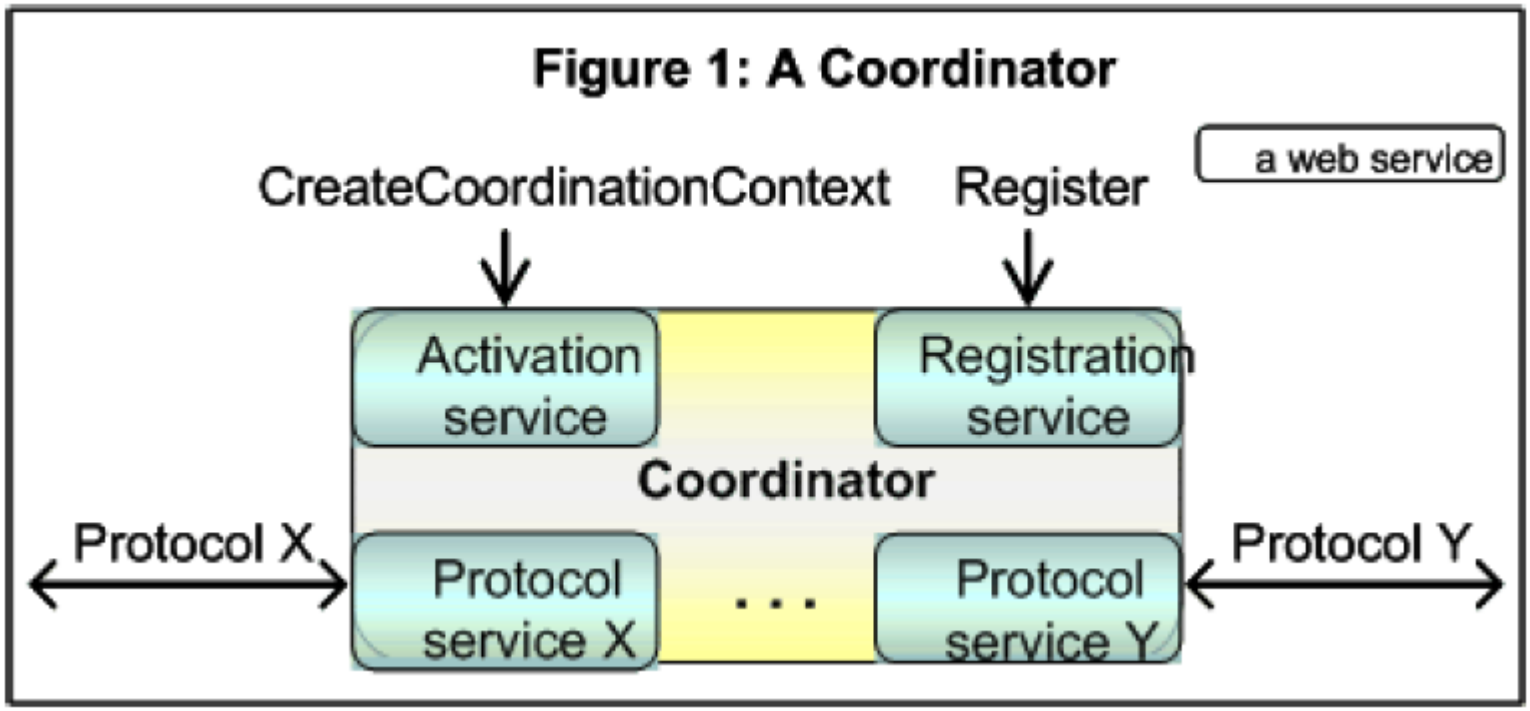
Modeling Coordination Protocols – 3/4



Modeling Coordination Protocols – 4/4: WS-BA



WS-Coordination



Web Services Coordination (WS-Coordination) Version 1.1

- One coordinator for many participants

Background

Research at IAAS

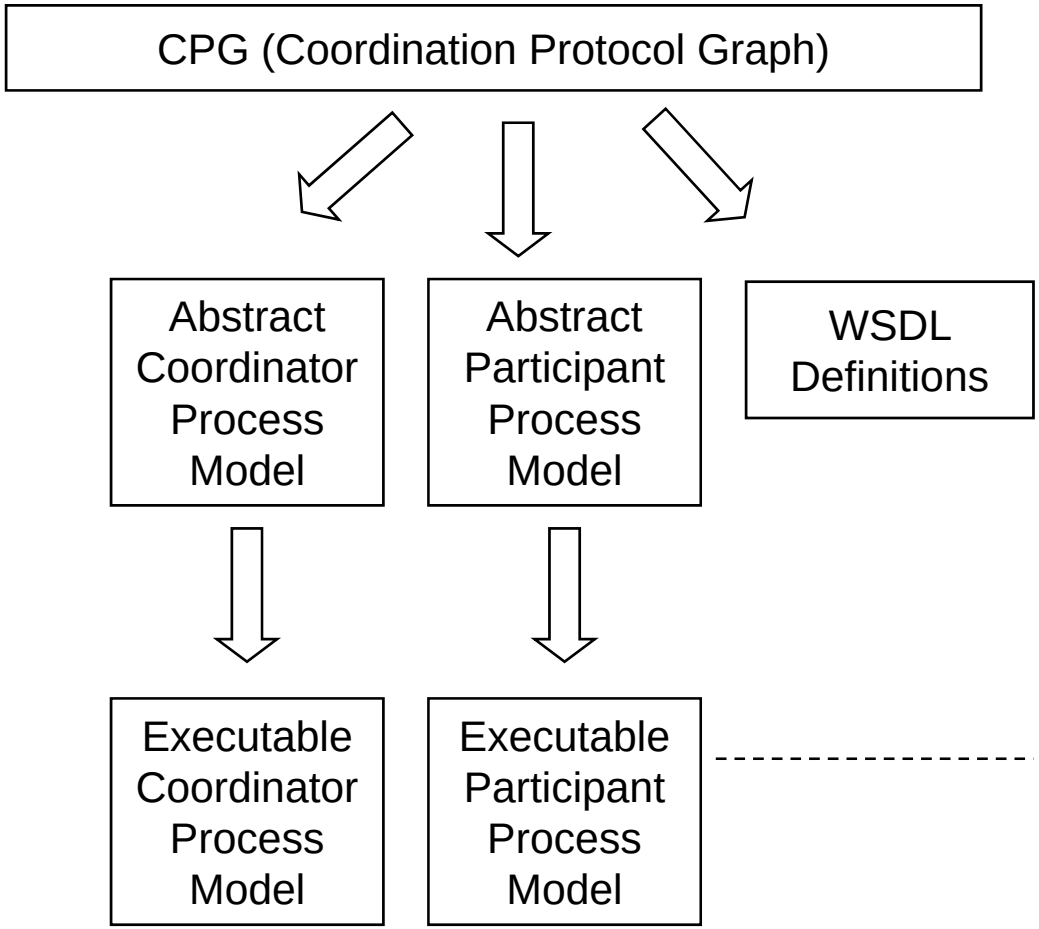
- Aim: Describe **any coordination protocol** by BPEL
  - Rania Khalaf's protocol for split loops and scopes
    Supporting Business Process Fragmentation While Maintaining Operational Semantics: A BPEL Perspective, Dissertation, University of Stuttgart
  - Auctions – e.g., multiple round sealed auction
    F. Leymann and S. Pottinger: Rethinking the Coordination Models of WS-Coordination and WS-CF, ECOWS'05
- Mapping of OASIS-BTP and WS-CAF to WS-Coordination (Master's Thesis)
- Replacement of BPEL's WS-BA transaction behavior by arbitrary transaction behavior
- Externalization of BPEL's transaction behavior
  - S. Pottinger, R. Mietzner and F. Leymann:
    Coordinate BPEL Scopes and Processes by Extending the WS-Business Activity Framework, CoopIS 2007

Why BPEL?

- BPEL is on a higher level than usual programming languages
- BPEL has native support for
  - Concurrency
  - Forward and backward recovery
    - Fault handler to catch fault and do alternative action
    - Phoenix behavior: If BPEL engine crashes, the state before the crash is restored
  - Scalability
- BPEL is supported by the most important vendors
  - IBM, Oracle, Microsoft, Sun, …

The Approach

Overview



- Domain-specific Language
- Platform-Independent Model
- Marking: WSDL names

- Automatic generation from CPG
- Abstract BPEL processes
- WSDL interfaces for coordinator and participant process

- Manual refinement from abstract BPEL processes
- Replacing opaque tokens with concrete BPEL code

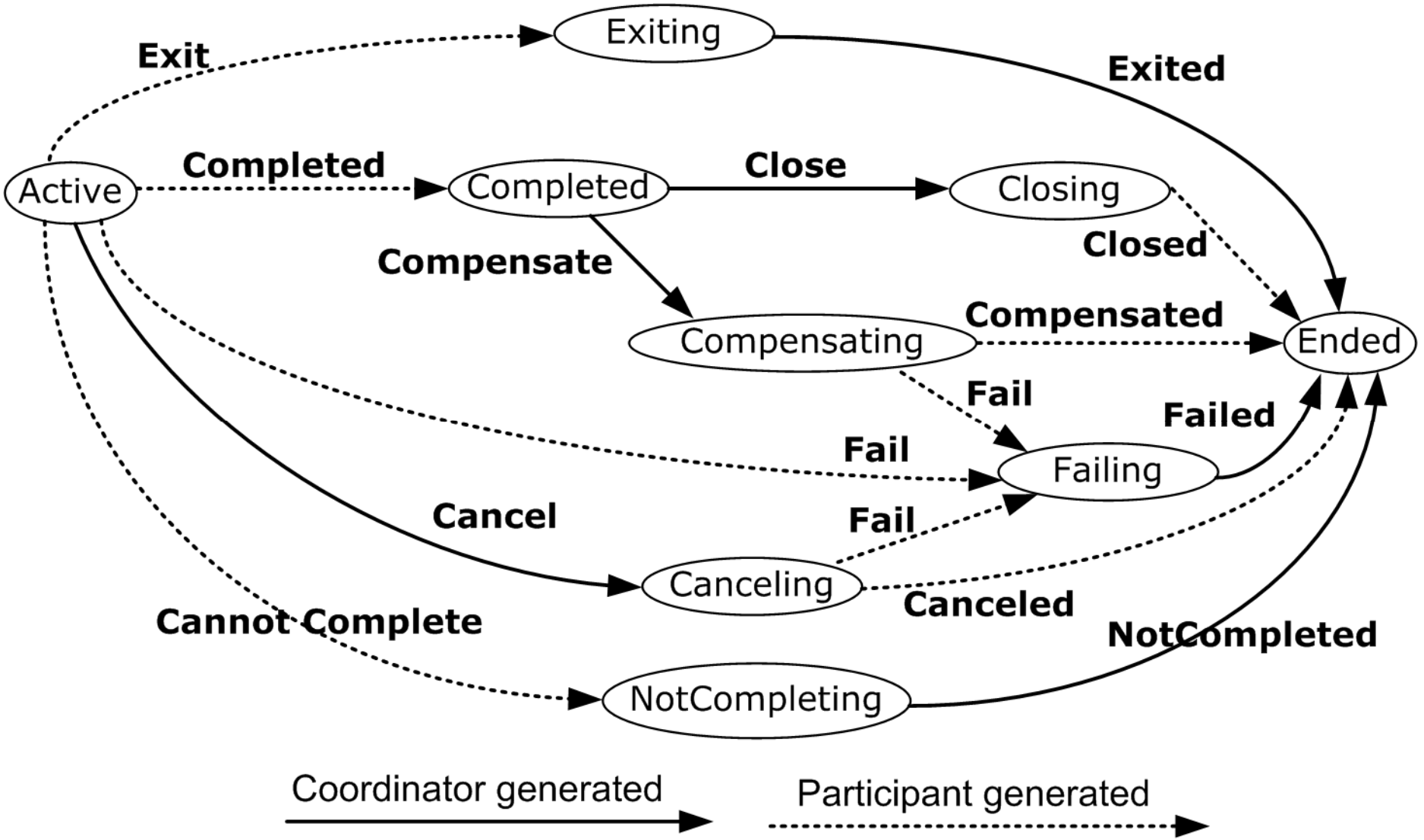
Participant

Observations and Basic Idea

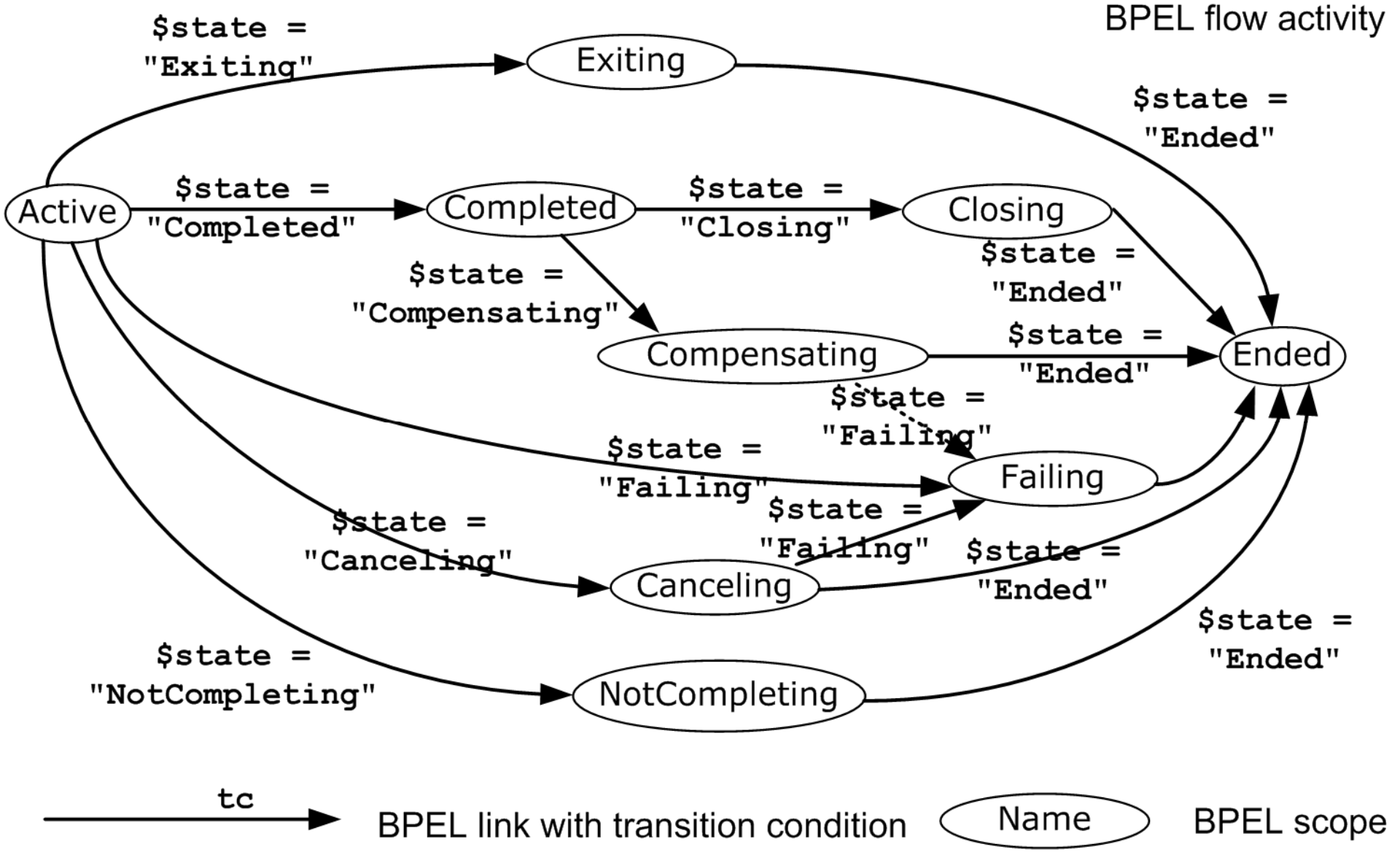
- Participant is in one state at a time
- BPEL supports graph-based programming
- A CPG is a graph

- Translate Graph Structure as Close as Possible to BPEL
  - “Element Preservation Strategy” in J. Mendling, K.B. Lassen, and U. Zdun. On the Transformation of Control Flow between Block-Oriented and Graph-Oriented Process Modeling Languages. IJBPIM, 3(2), September 2008.
  - No loops

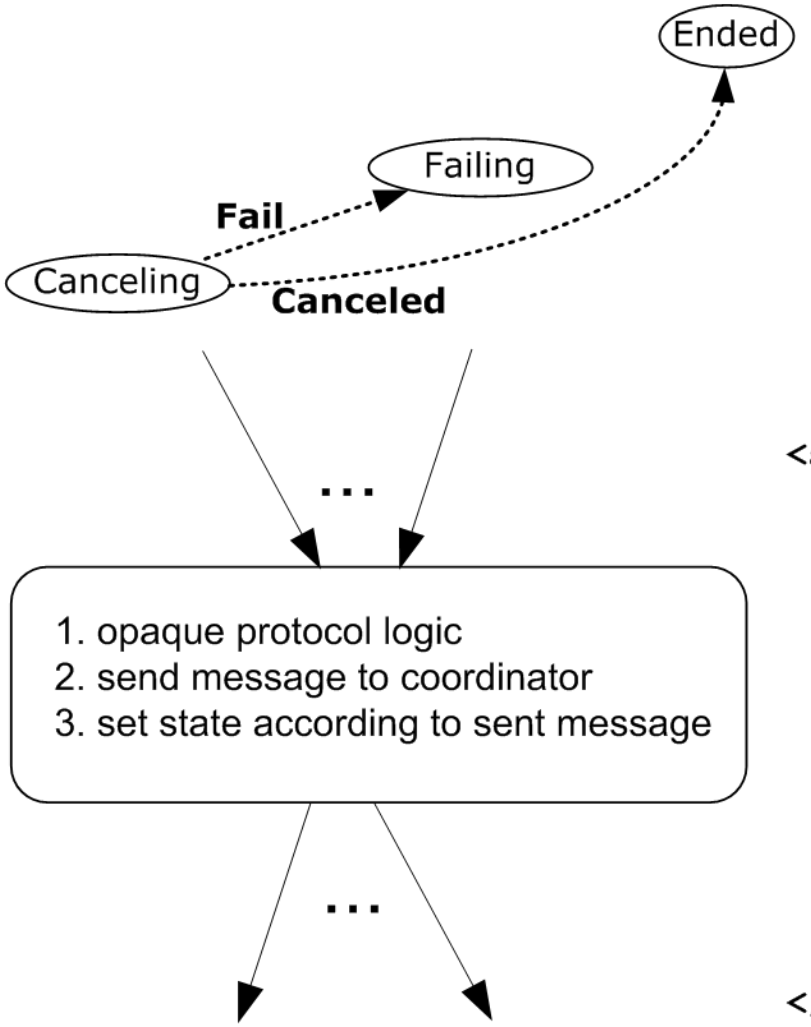
WS-BA with Participant Completion



Implementation in BPEL



State Canceling

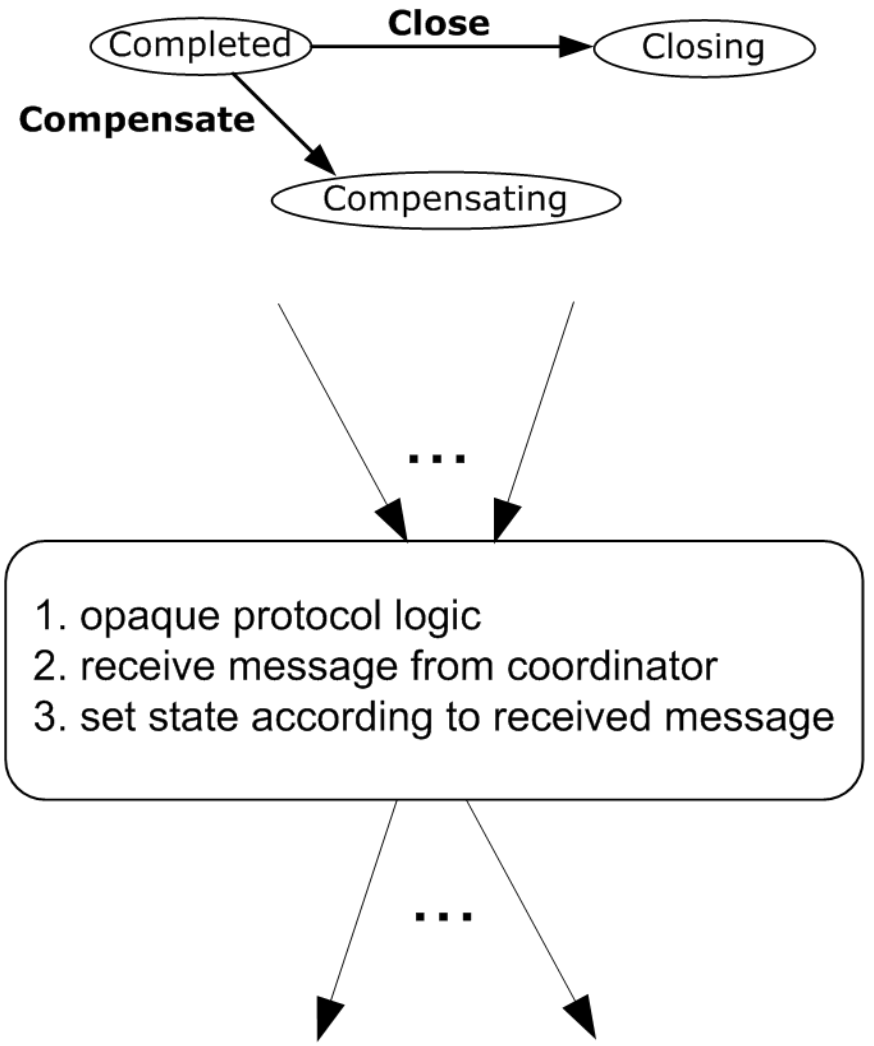



```
<scope name="state">
  <targets>…</targets>
  <sources>…</sources>
  <sequence>
    <opaqueActivity/>
    <if>
      <condition opaque="yes"/>
      <invoke coordinator, "Fail" />
      <else>
        <invoke coordinator, "Canceled" />
      </else>
    </if>
  </sequence>
</scope>
```

Conceptional behavior

Generated BPEL code

State Completed

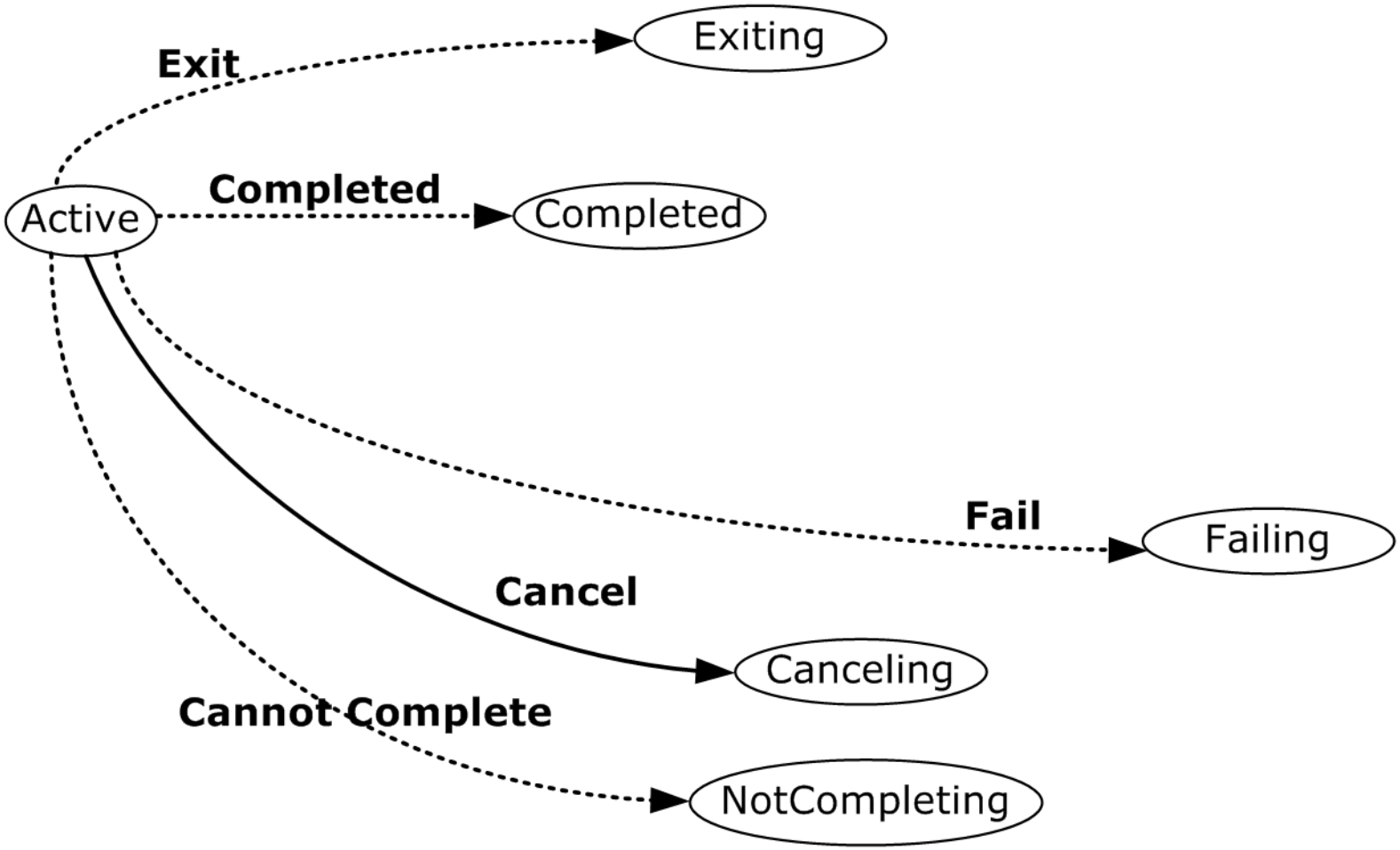


Conceptional behavior

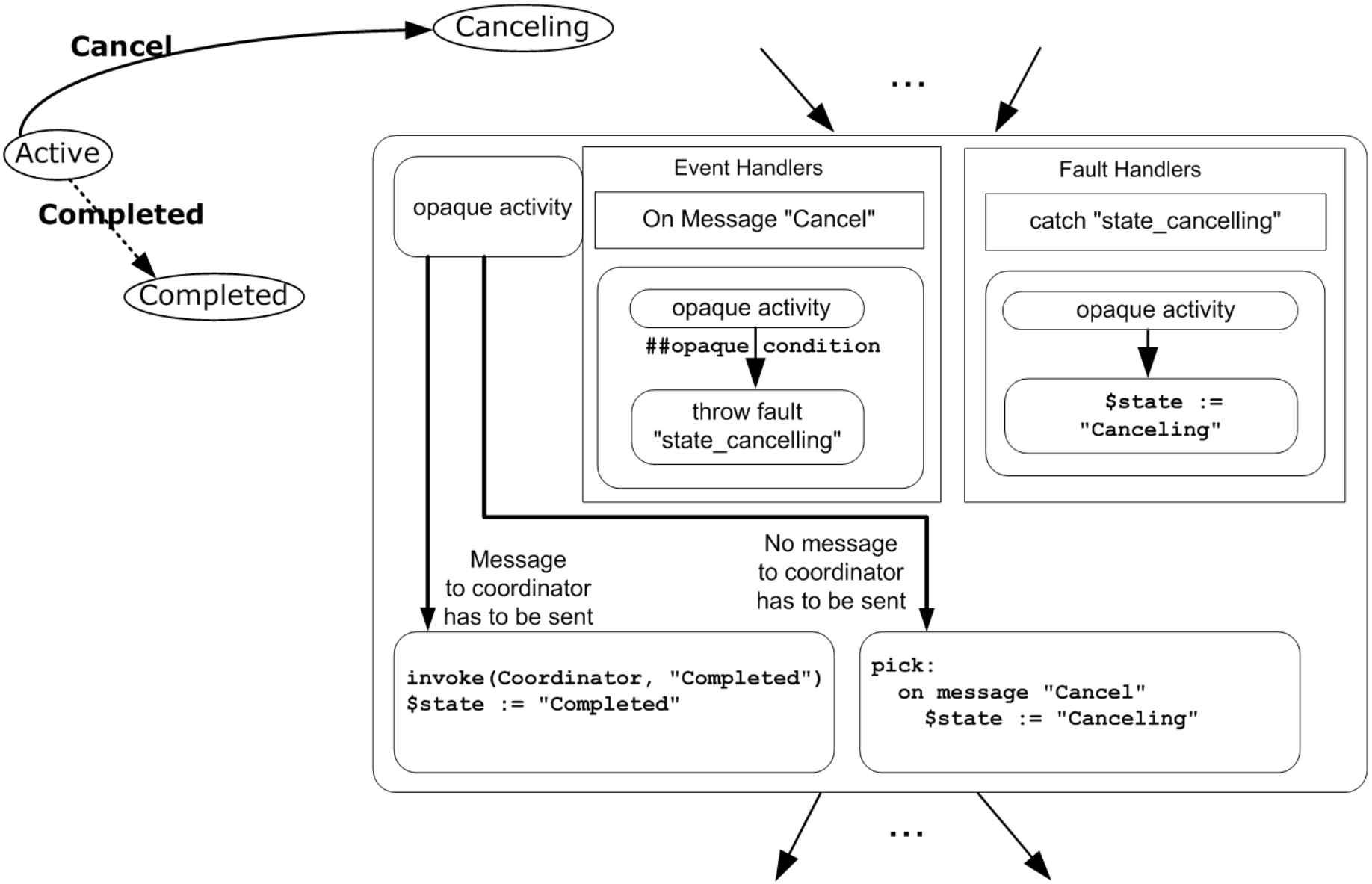
```
<scope name="state">
  <targets>…</targets>
  <sources>…</sources>
  <variables>…</variables>
  <sequence>
    <opaqueActivity/>
    <pick>
      <onMessage "Close">
          state := "Closing"
      </onMessage>
      <onMessage "Compensate">
          state := "Compensating"
      </onMessage>
    </pick>
  </sequence>
</scope>
```

Generated BPEL code

State Active



State Active simplified



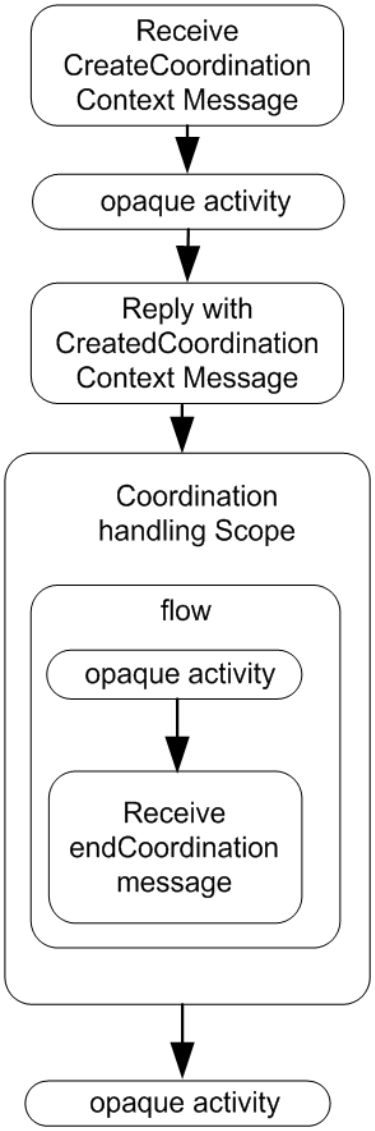
Coordinator

Observations and Basic Idea

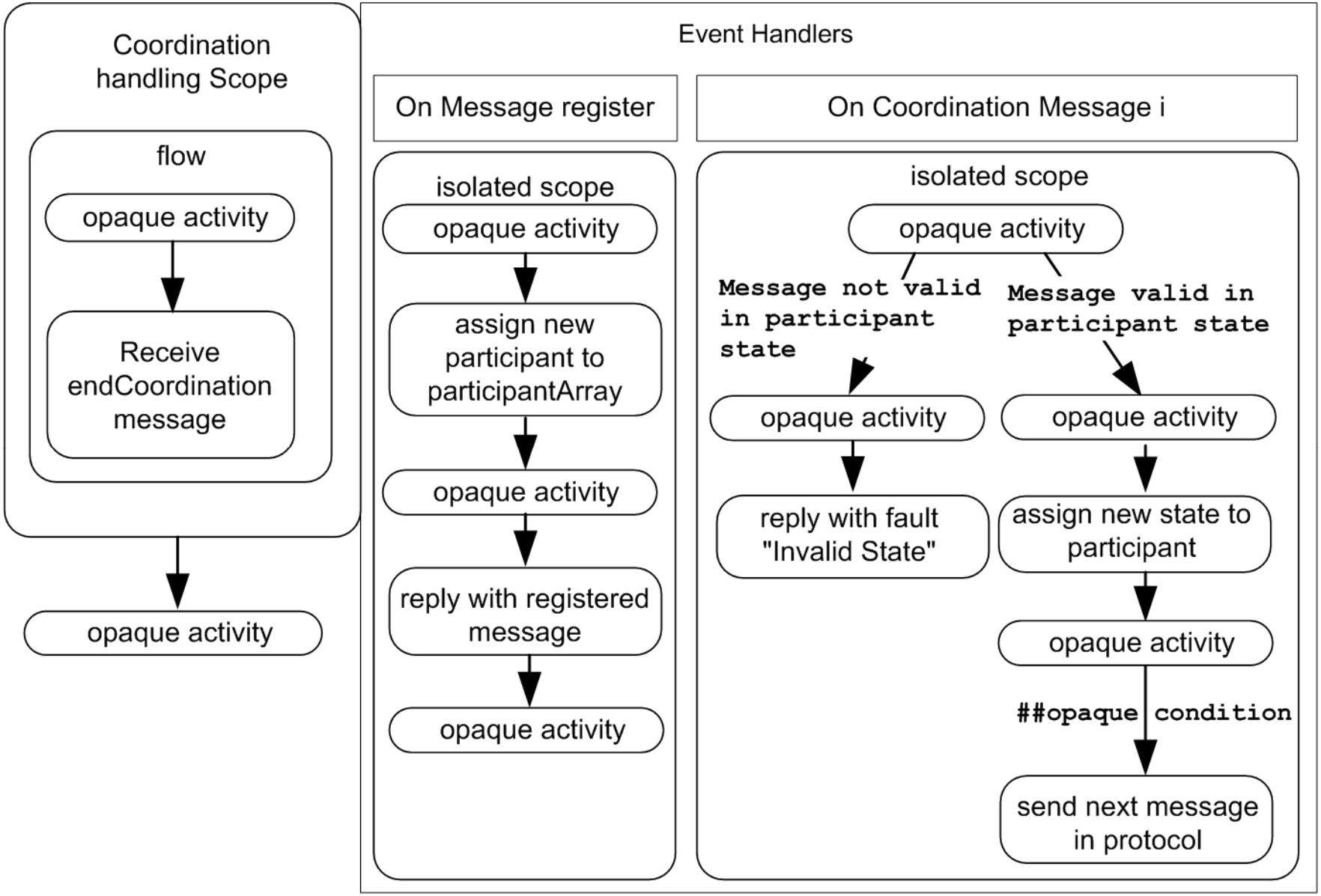
- Coordinator has to hold state for each participant
- Participant can register at any time
- Not specified how to end coordination

- Impossible to keep graph-structure
- Generate code for managing
  - Activation
  - Registration
  - End of Coordination
- Use Event-Condition-Action idea

Activation and Registration Services



Overview of the Coordinator



Reaction to message “fail”

- If sending participant in state “Active”
  - Set flag “failed” to true
  - For all participants in state “Active”
    - Send “Cancel”
  - For all participants in state “Completed”
    - Send “Compensate”
  - Send “Failed”
- Else
  - Fail came from Canceling or Compensating
  - No action required

Conclusion

Conclusion and Outlook

- MDA approach for WS-Coordination
  - Generated BPEL Skeletons
  - Participant easy to complete
  - Coordinator less easy to complete
- Loops
  - Jussi Vanhatalo, Hagen Völzer, Jana Koehler. The Refined Process Structure, BPM 2008
  - W. Zhao, R. Hauser, K. Bhattacharya, B. R. Bryant, F. Cao. Compiling business processes: untangling unstructured loops in irreducible flow graphs. International Journal of Web and Grid Services, 2006
- Support of Coordinator Hierarchy
- Support of mixed protocols at a coordinator
  - WS-BA and WS-AT

End of Document