

Model Driven Design of Distribution Patterns for Web Service Compositions

Ronan Barrett and Claus Pahl

`ronan.barrett@computing.dcu.ie`



School of Computing,
Dublin City University,
Dublin, Ireland

Motivation

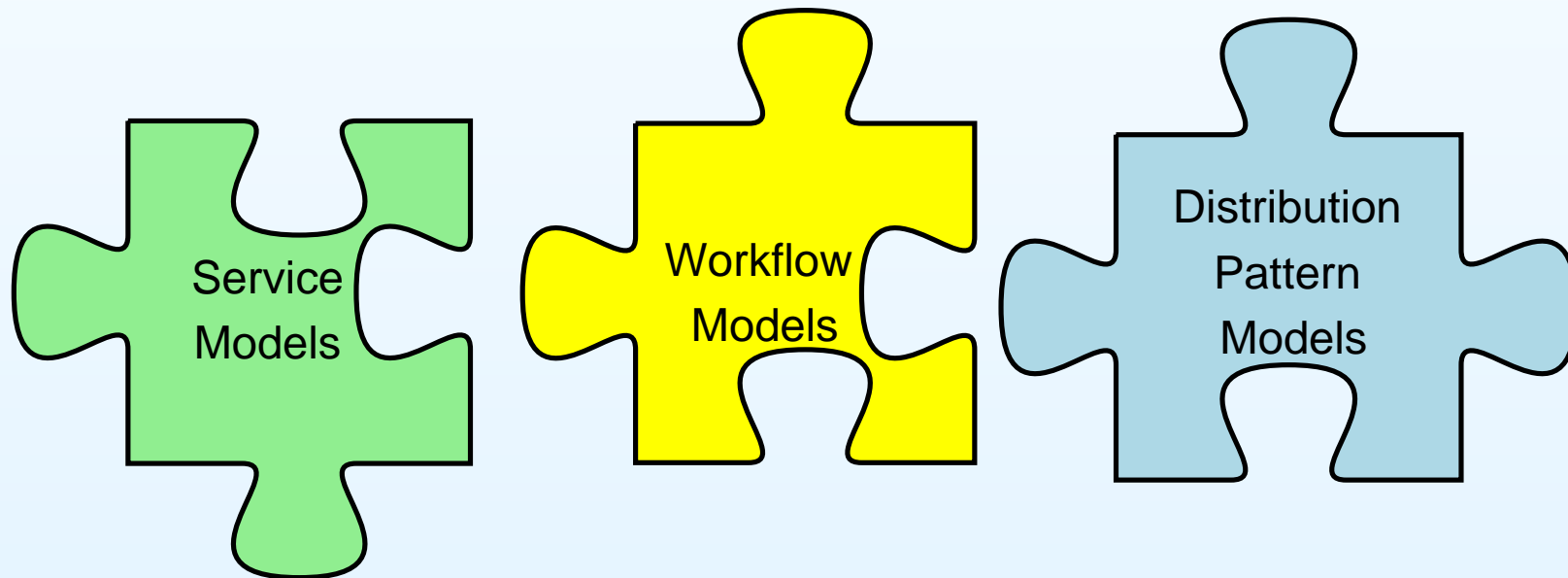
- Web service compositions are inflexible, can this inflexibility be reduced?
 - Coding effort is considerable
 - Architecture is fixed
 - QoS dependent patterns are lost
- Goal: Increase flexibility by explicitly modeling the compositional aspects
 - No coding effort
 - Model based development
 - Flexible architecture
 - Make QoS dependent patterns visible

Outline

- Modeling aspects
- Distribution patterns
- Modeling approach & transformation technique
- Tool Implementation
- Enhancements & future work
- Conclusions/Questions

Modeling Aspects

- Service modeling * e.g. WSDL as UML
- Workflow modeling † e.g. WS-BPEL as UML
- Distribution pattern modeling



*Armstrong

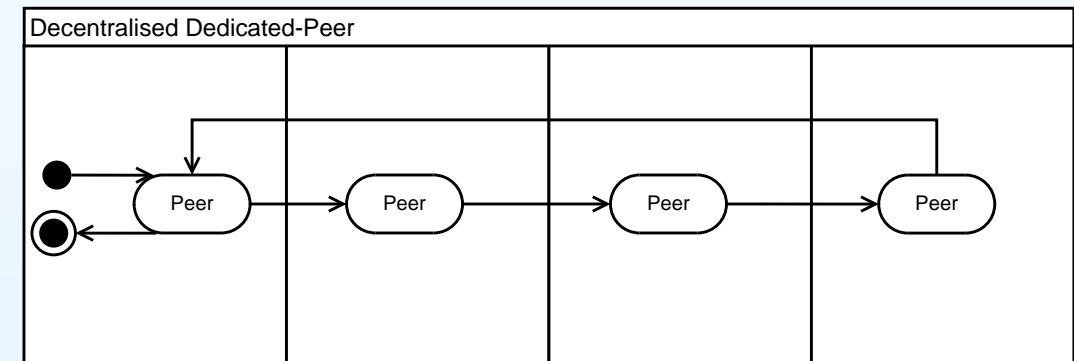
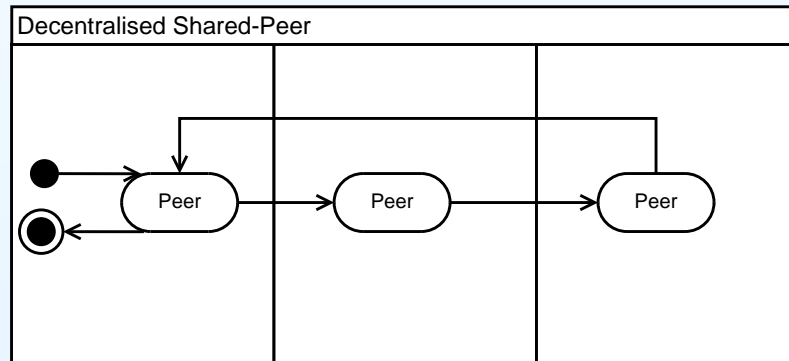
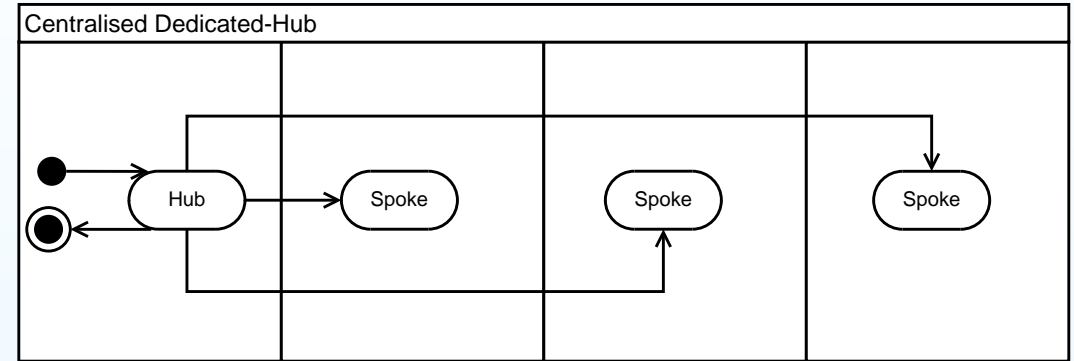
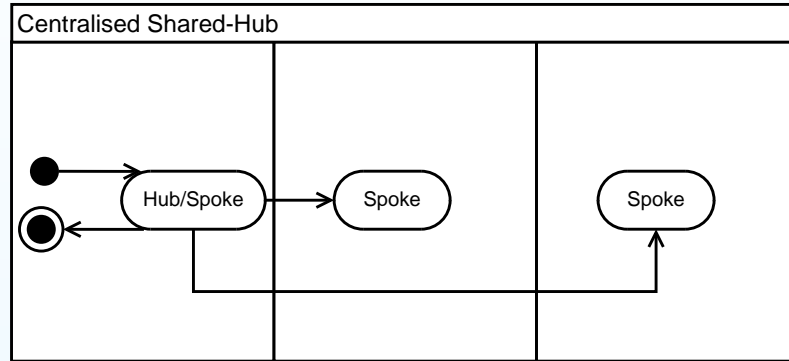
†Grønmo & Solheim

Distribution Patterns

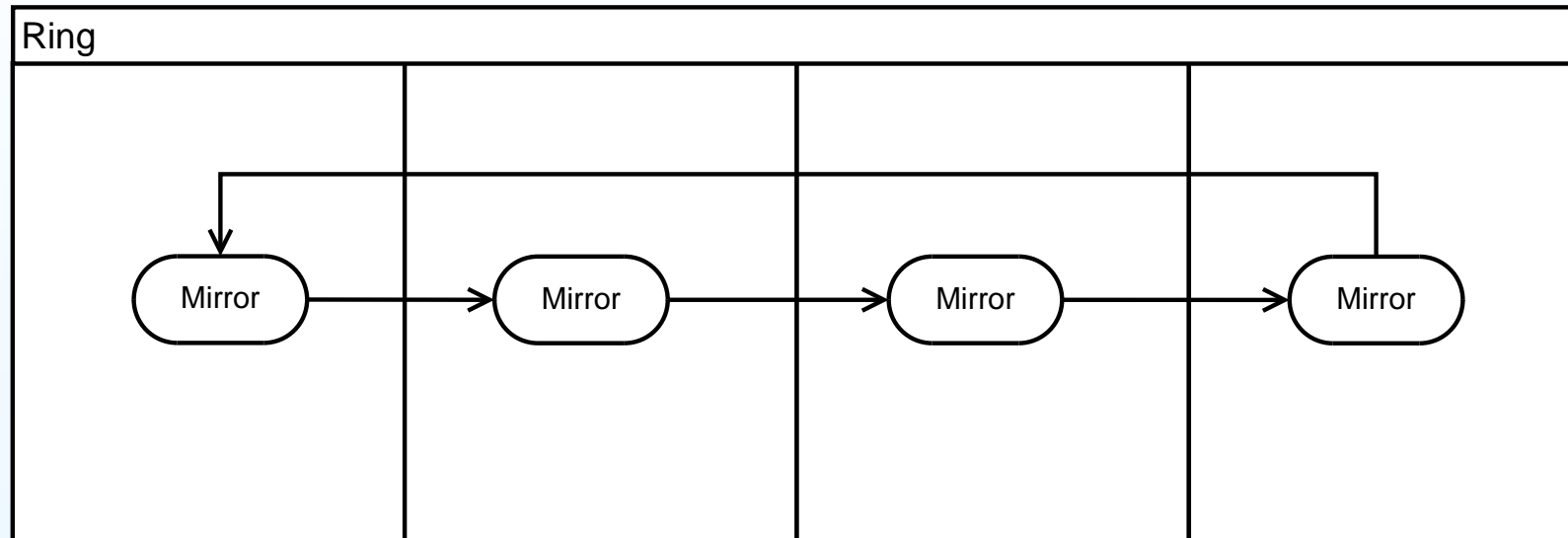
“Expression of how a composed system is to be deployed”

- Patterns express varying QoS metrics
- Patterns independent from workflow
- Architecture is flexible
- Pattern categories
 - Core patterns
 - Auxiliary patterns
 - Complex patterns

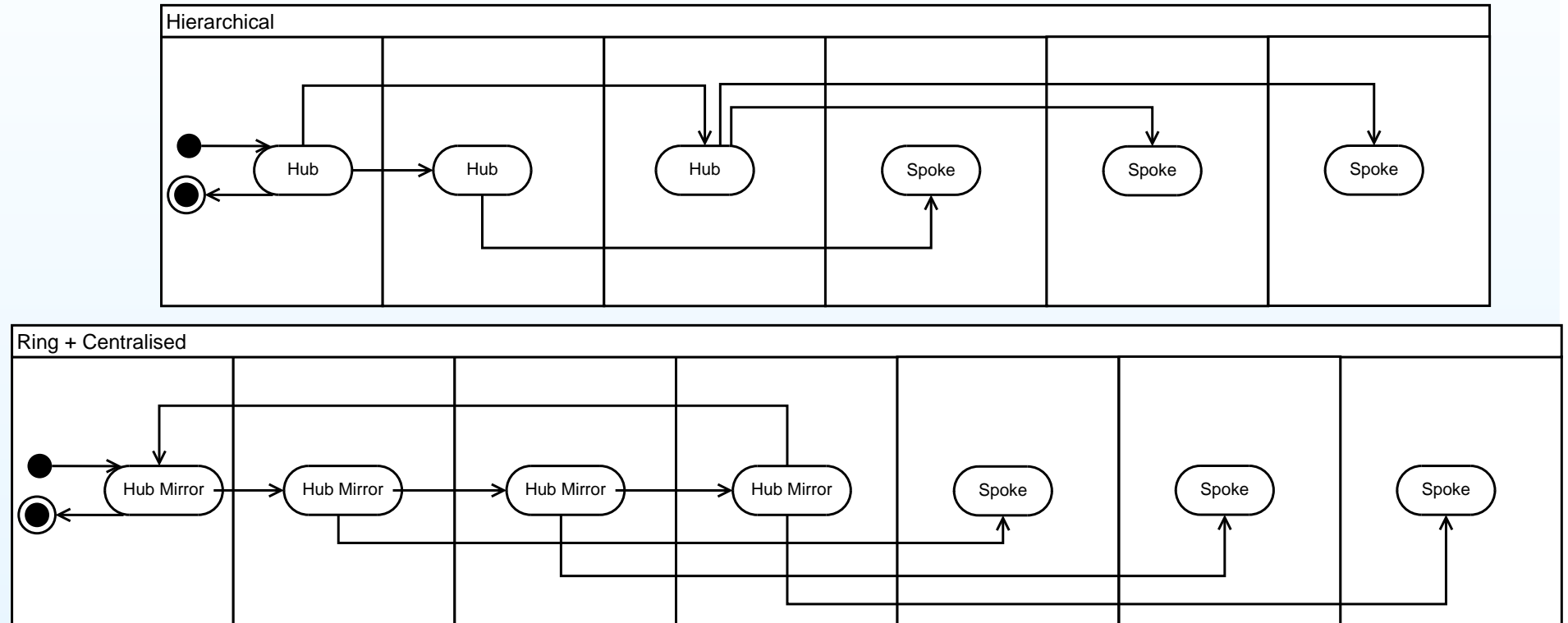
Distribution Patterns - Core Patterns



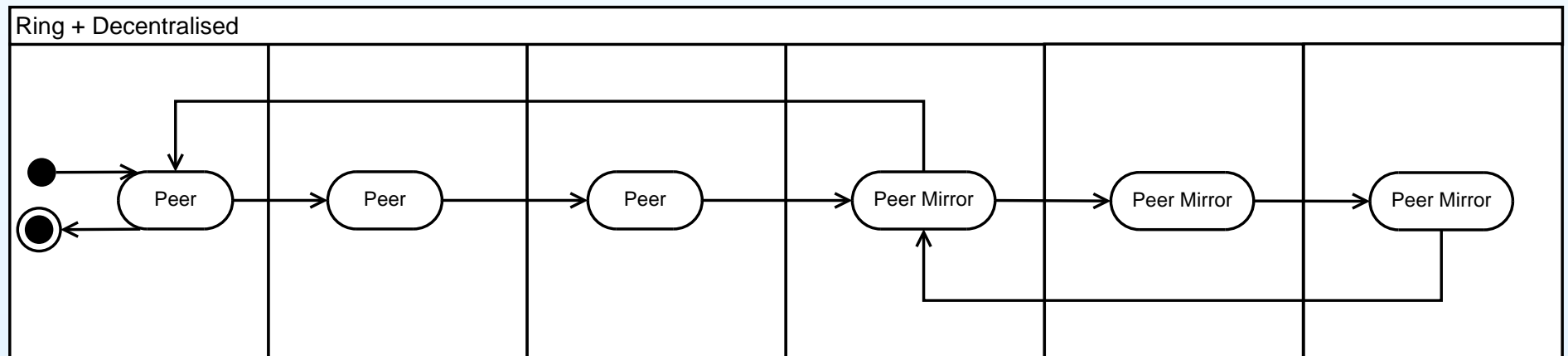
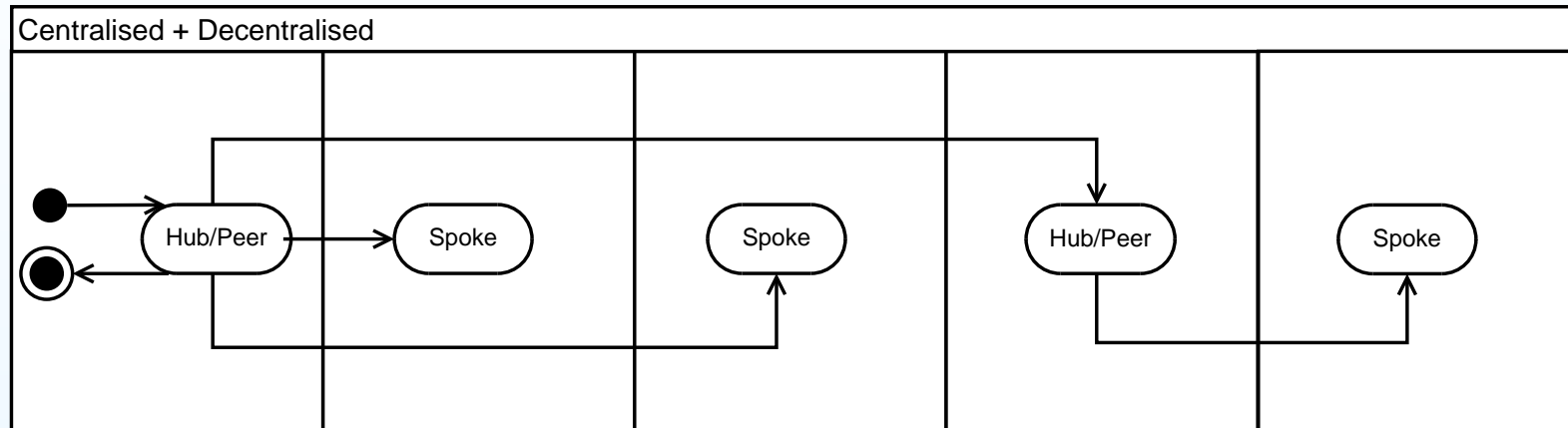
Distribution Patterns - Auxiliary Patterns



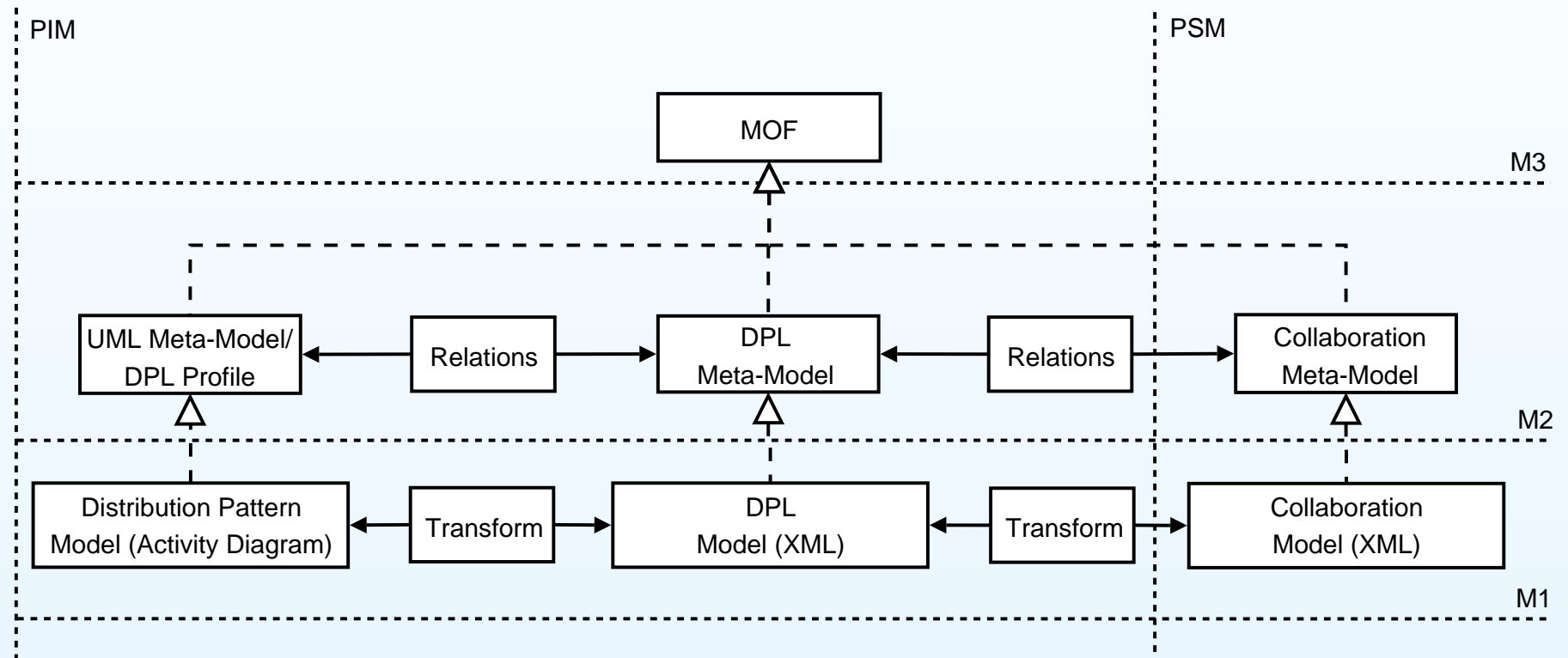
Distribution Patterns - Complex Patterns (a)



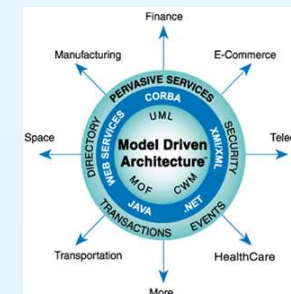
Distribution Patterns - Complex Patterns (b)



Modeling Approach



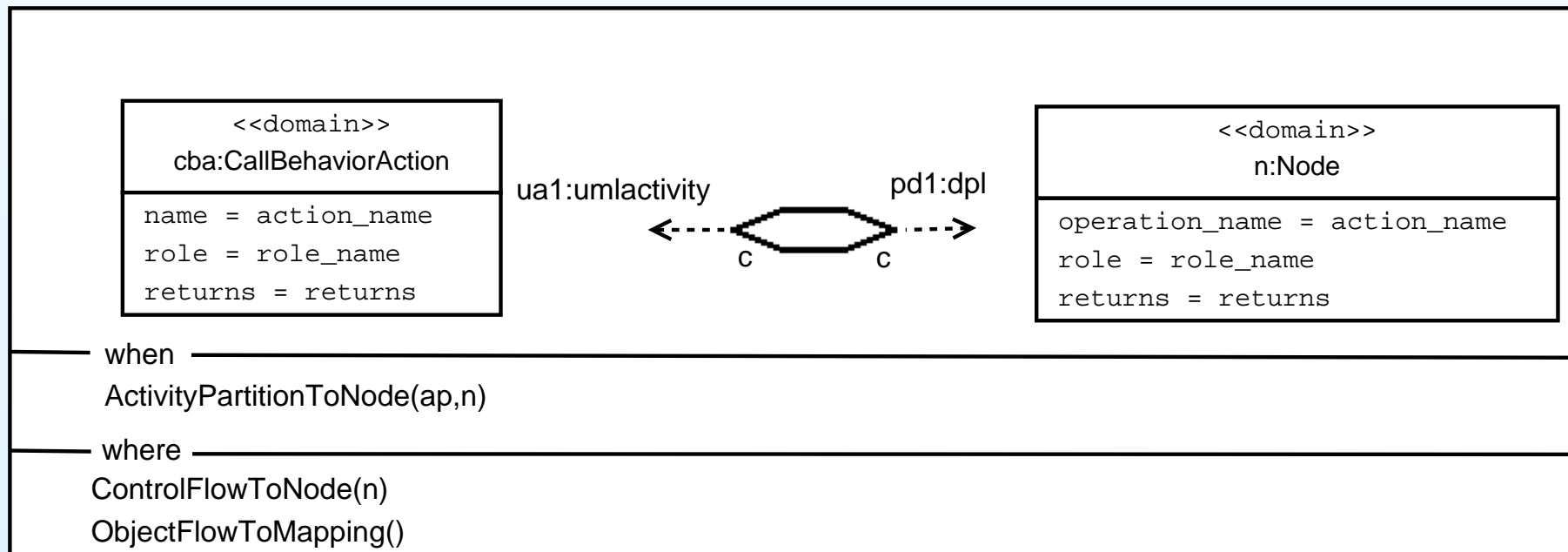
- Based on OMG standards
- Relations defined using QVT
- Transformations written in XSLT



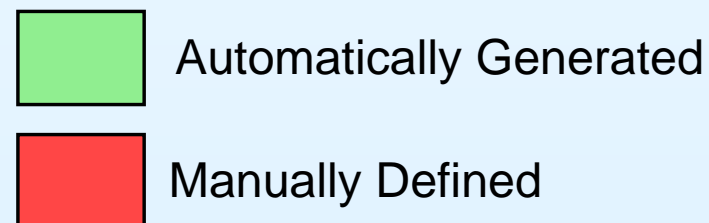
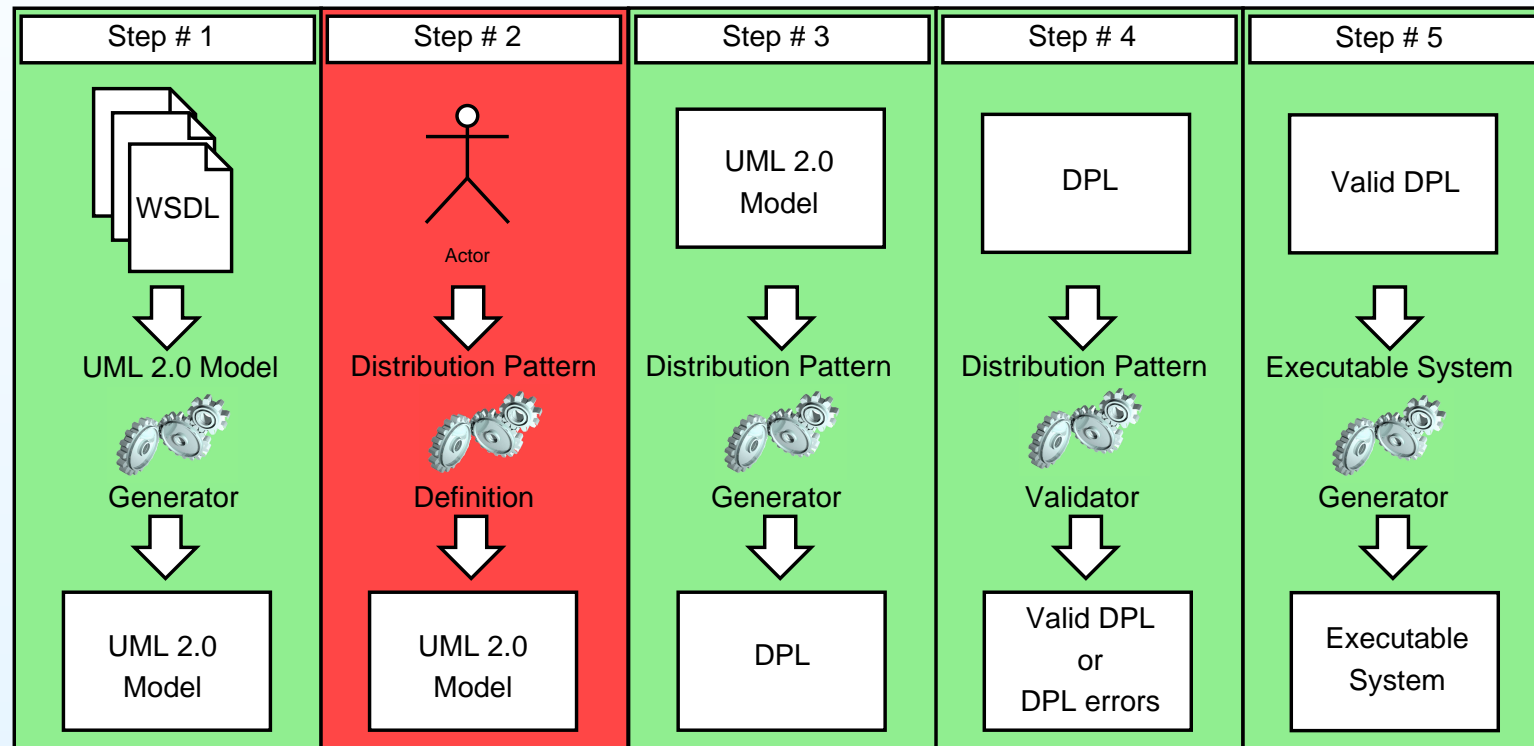
Modeling and Transformation Basis

- Relation defined at meta-model level (M2)
- Relations provide traceability between meta-models
- Each relation mapped to a XSLT transformation

CallBehaviorActionToNode

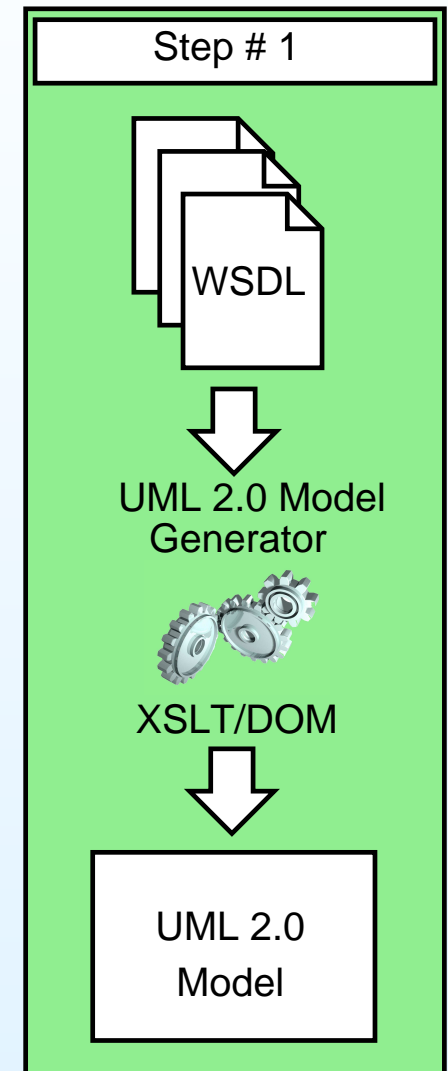
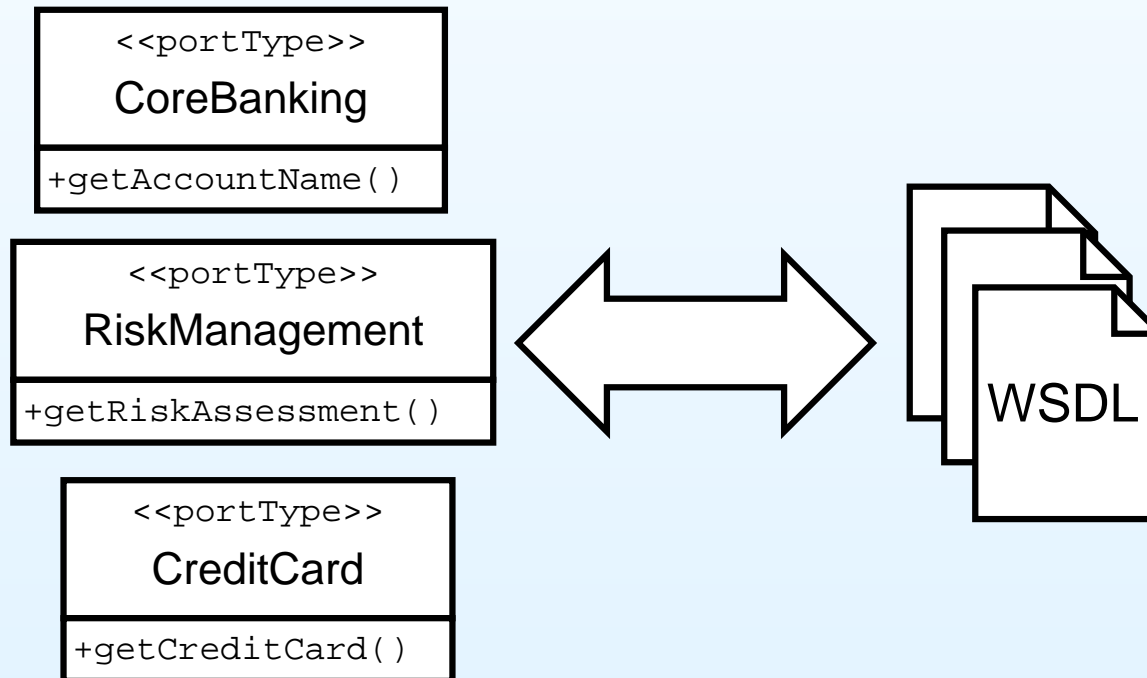


Modeling and Transformation Technique

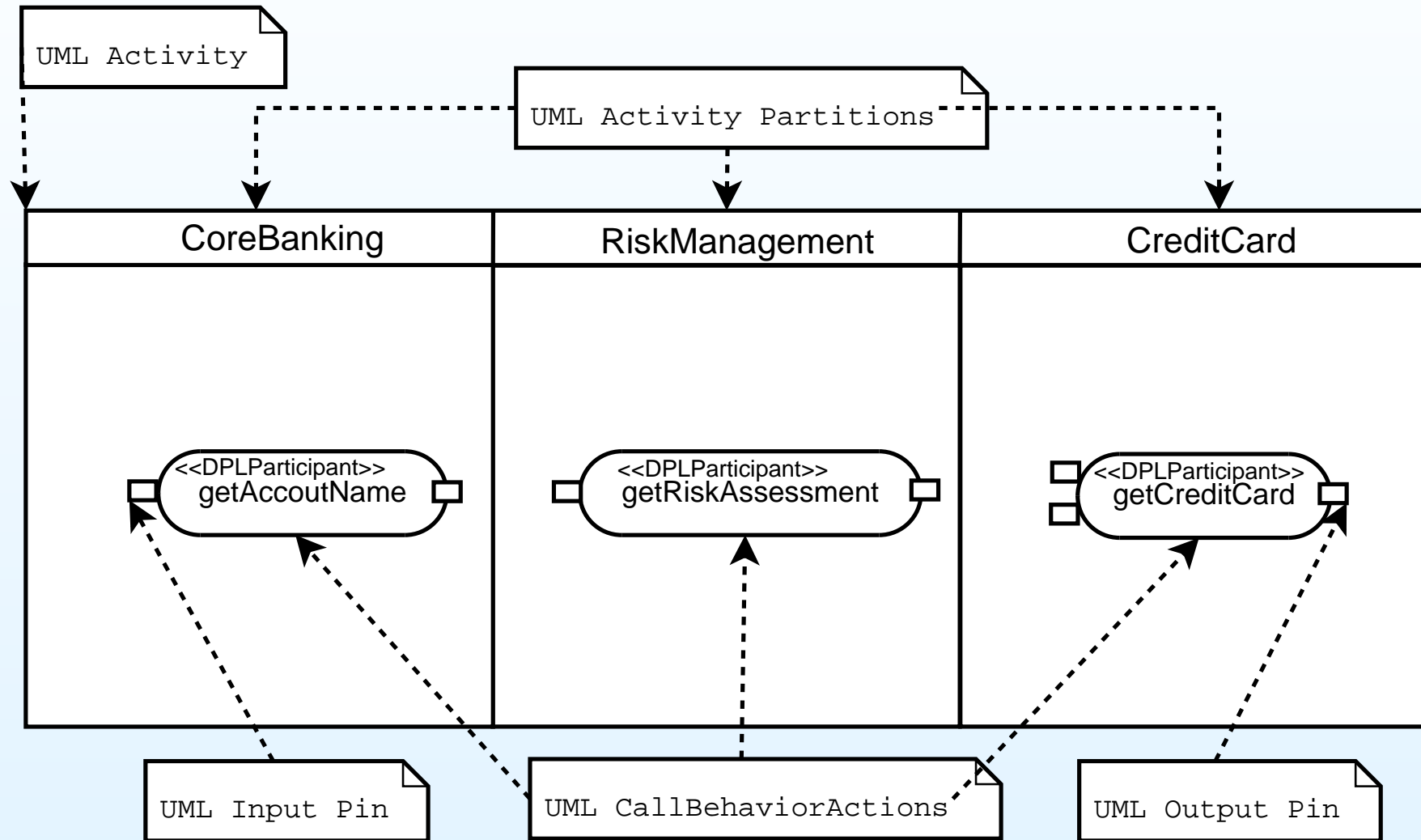


Step 1a - From Interface To UML Model

- Web service interfaces as input
- Generate UML activity diagram
- Automatically apply UML Profile

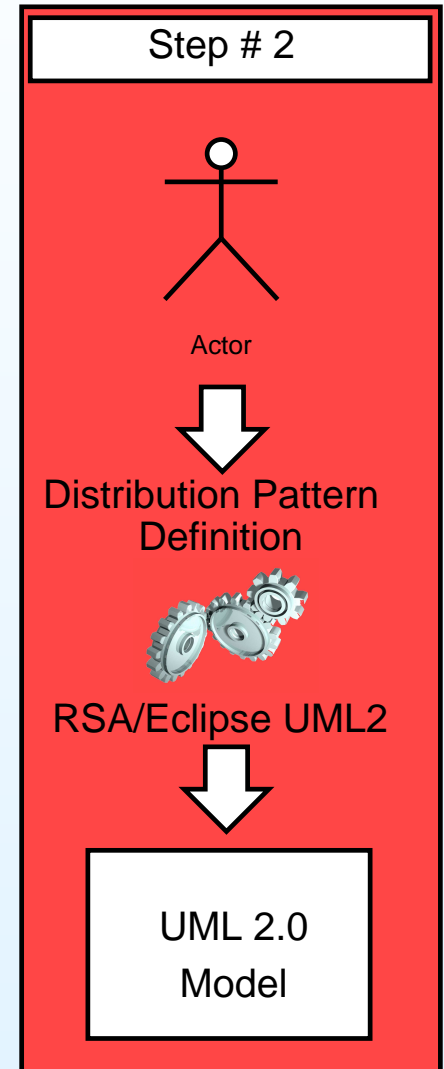


Step 1b - From Interface To UML Model

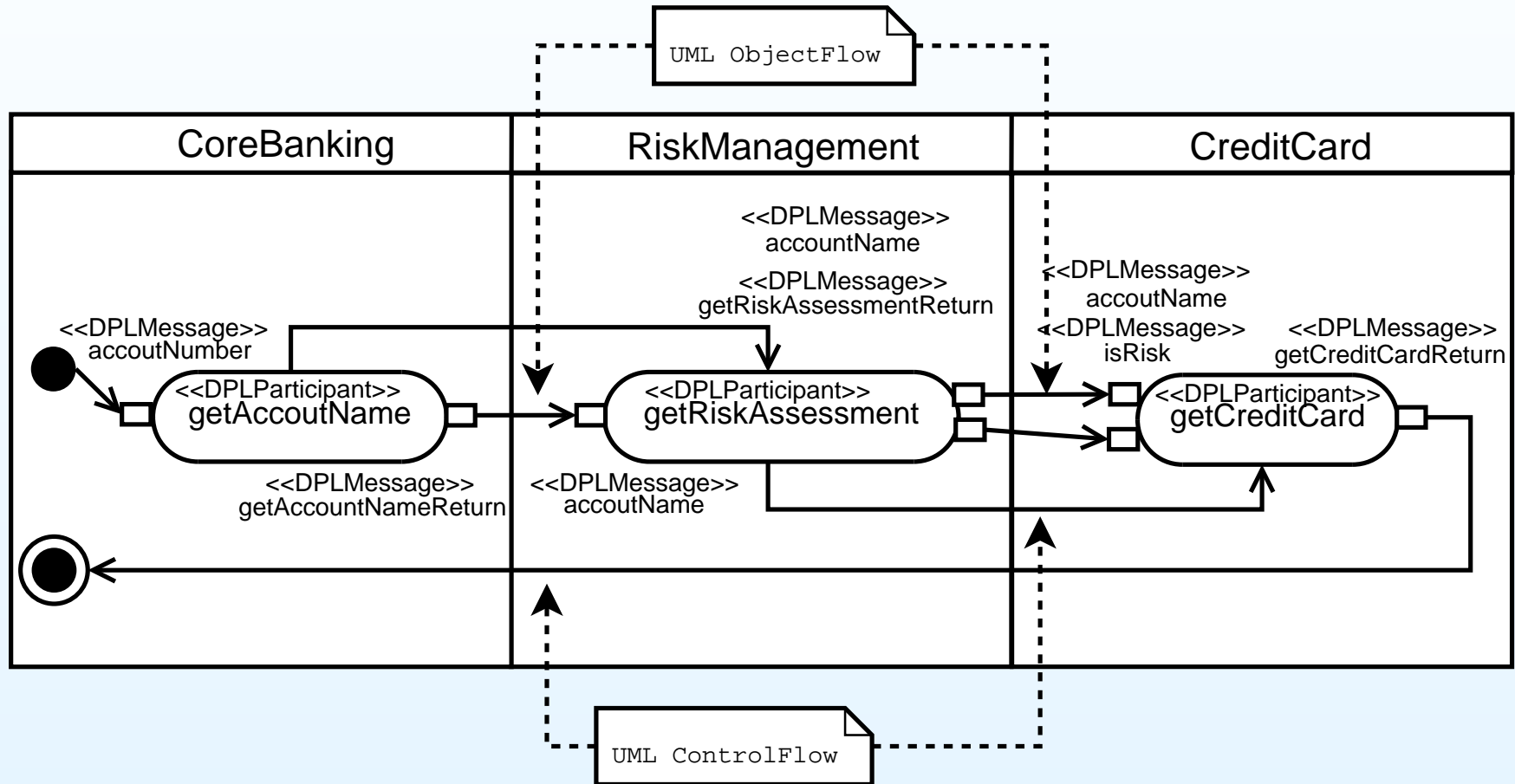


Step 2 - Distribution Pattern Definition

- Manually apply values to UML Profile
 - Select distribution pattern
 - Select collaboration language
 - Select roles for actions
- Manually apply connections
 - Apply ControlFlows
 - Apply ObjectFlows
 - Place initial/final nodes
- UML 2.0/XMI 2.0 tool support
 - Rational Software Architect
 - Eclipse UML 2

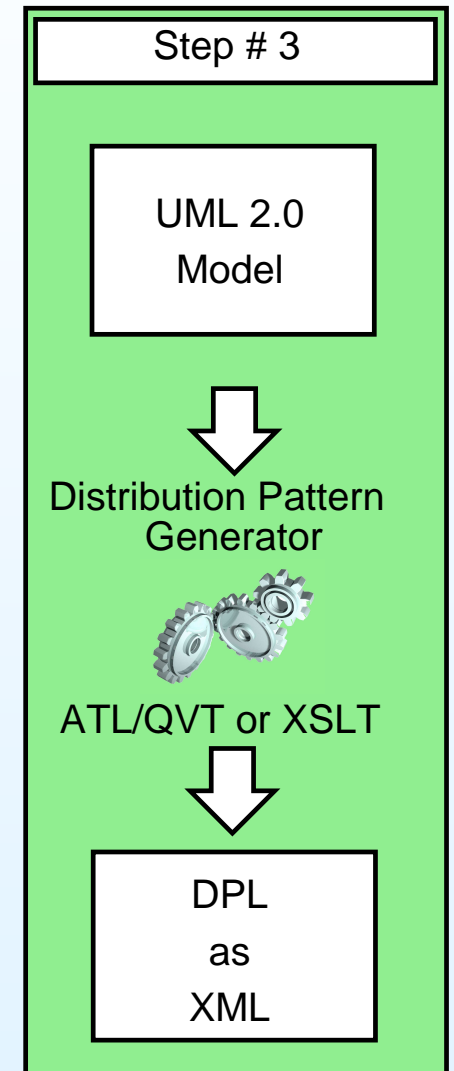


Step 2b - Distribution Pattern Definition



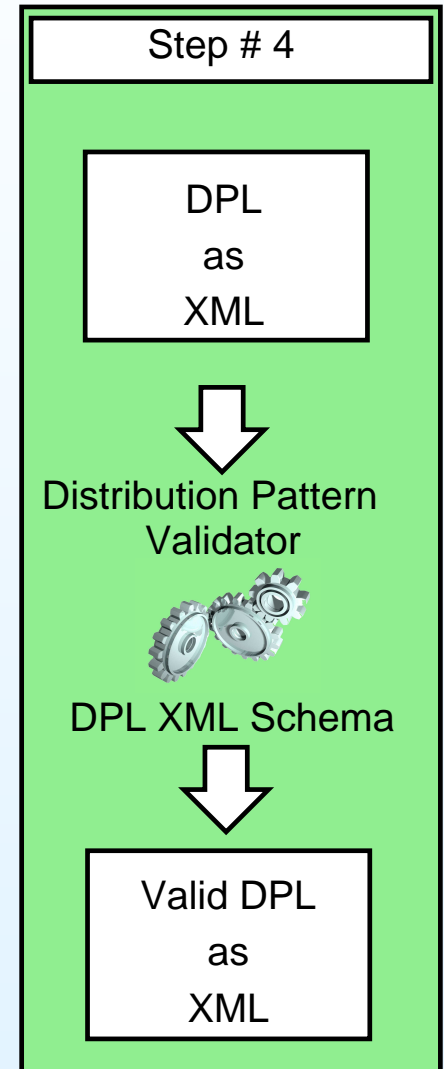
Step 3 - From Model to DPL

- Novel Distribution Pattern Language (DPL)
- $DPL \equiv (UML \text{ Model} + UML \text{ Profile})$
- No reliance on UML
- Distribution pattern as XML
- Restricted by XML Schema
- Can be validated at build time



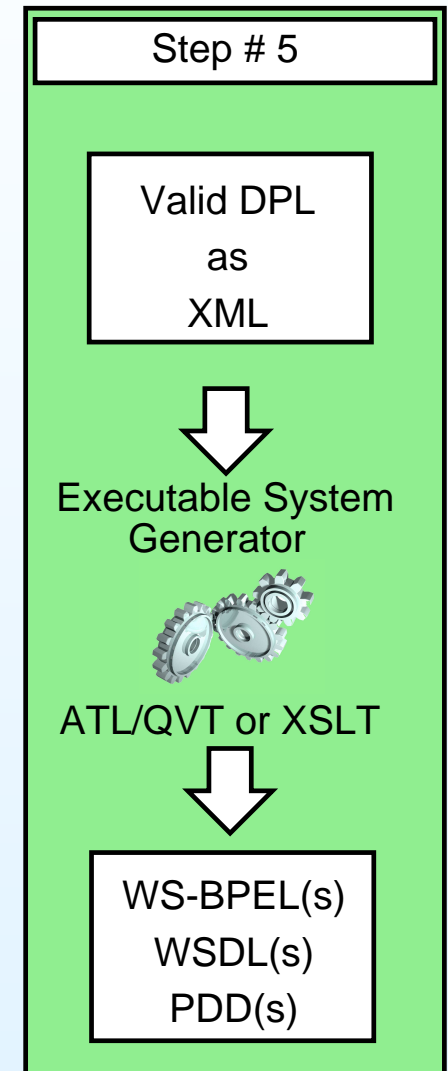
Step 4 - Model Validation

- Validate pattern instance against XML Schema
- Validation ensures
 - Connections are compatible with pattern
 - Profile values are compatible with pattern
- If errors are found return to Step 2

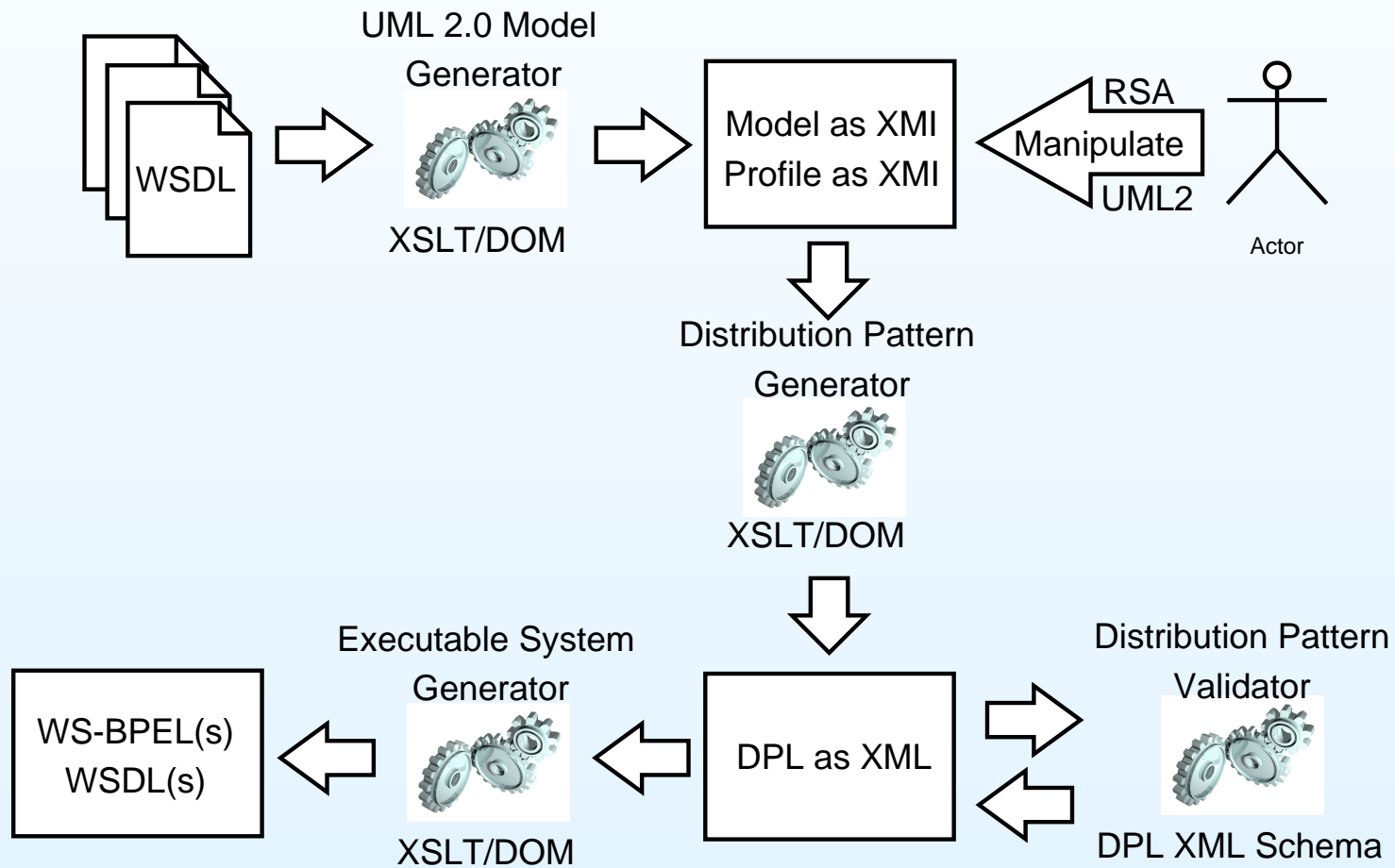


Step 5 - DPL to Executable System

- Valid DPL taken as input
- Generates
 - Interaction logic document(s)
 - Interface document(s)
 - Deployment descriptor(s)
- System is ready for deployment



Tool Implementation - TOPMAN



Enhancements & Future Work

- Use ATL to define relations instead of QVT/XSLT
- Use semantics to semi-automate modeling effort *
- Automatic deployment of generated executable system †
- Extensions to modeling approach
 - π calculus
 - Architecture Description Languages (ADL)
- Unify all three modeling aspects into the tool
- Document distribution pattern metrics

*To be presented on Wednesday at EDOC'06

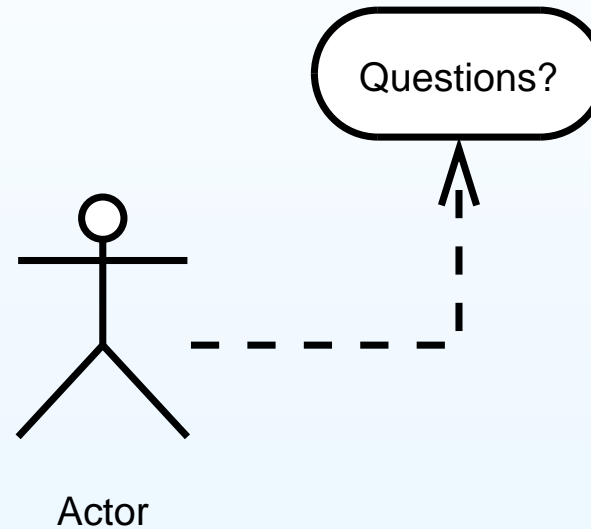
† Presented at ICWE'06

Conclusion

- Motivated new service modeling aspect
 - Distribution patterns
- Presented modeling approach
 - Standards based
 - Extensible
- Presented five step modeling technique
- Implementation
 - Compatible with existing modeling tools
 - Generates executable system

Questions

Go raibh maith agat!



Open Source tools used to make this presentation.....



Funded by the Irish Research Council for Science, Engineering and Technology: funded by the National Development Plan.

