

Model Driven Design of Distribution Patterns for Web Service Compositions

Presented by: Ronan Barrett

`ronan.barrett@computing.dcu.ie`



School of Computing,
Dublin City University,
Dublin, Ireland

Motivation

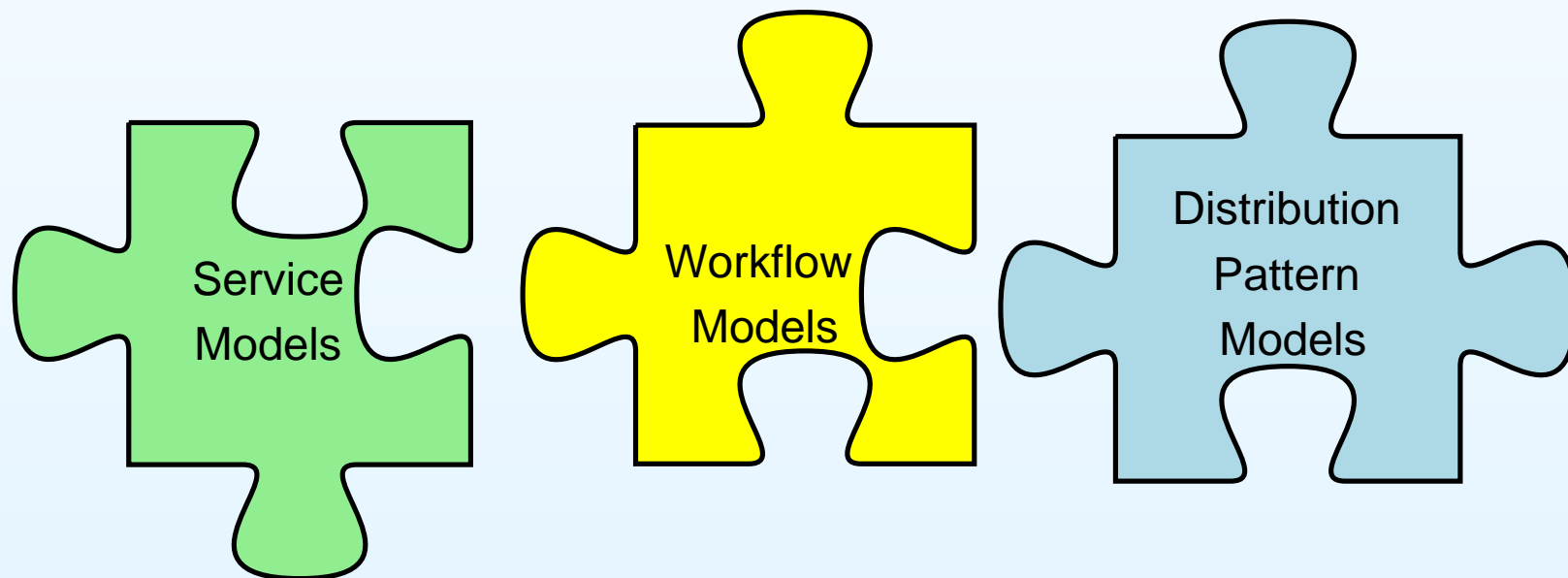
- Web service compositions are inflexible, can this inflexibility be reduced?
 - Coding effort is considerable
 - Architecture is fixed
 - QoS dependent patterns are lost
- Goal: Increase flexibility by explicitly modeling the compositional aspects
 - No coding effort
 - Model based development
 - Flexible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

Outline

- Modeling aspects
- Modeling approach & transformation technique
- Tool implementation
- Future work
- Conclusions/Questions

Modeling Aspects

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



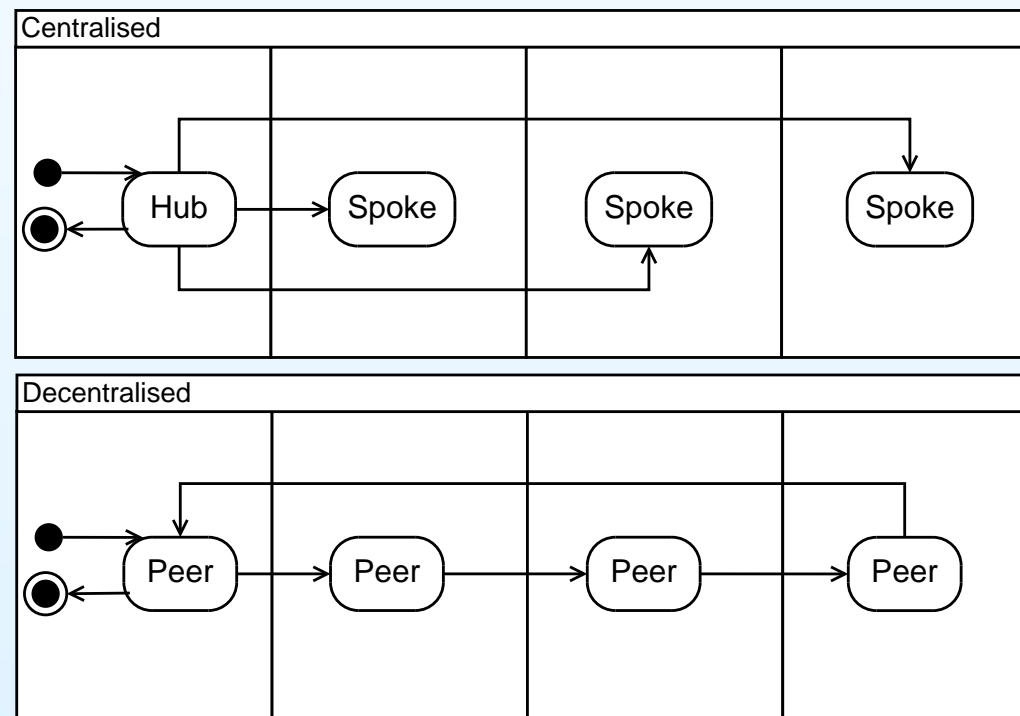
* Armstrong

† Grønmo & Solheim

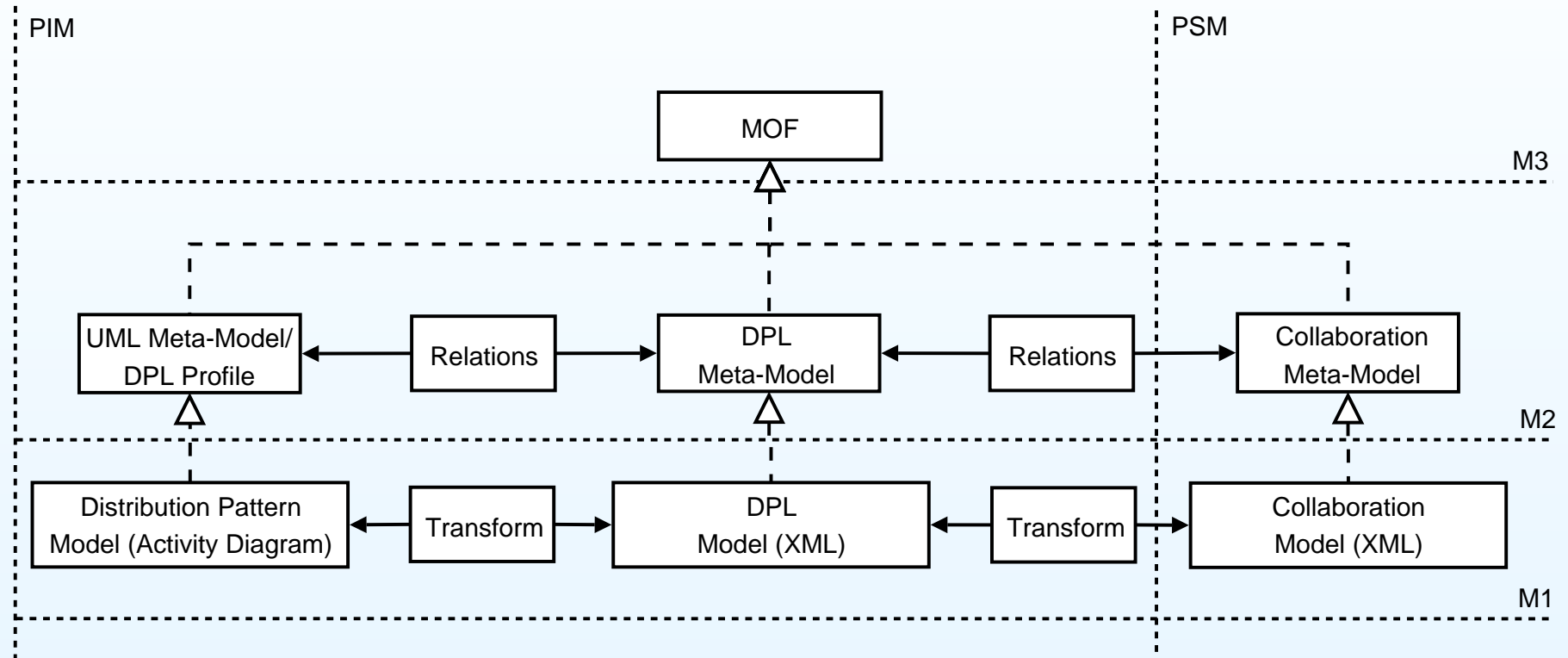
Distribution Patterns

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

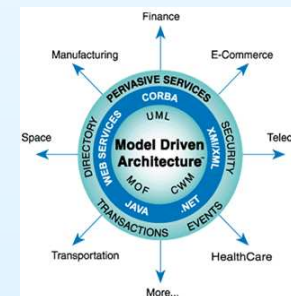
- Patterns express varying QoS metrics
- Patterns independent from workflow
- Architecture is flexible



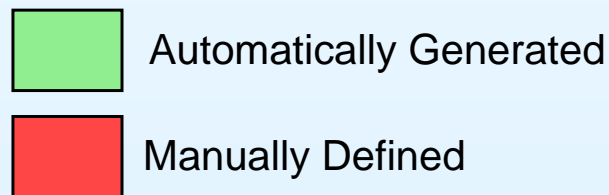
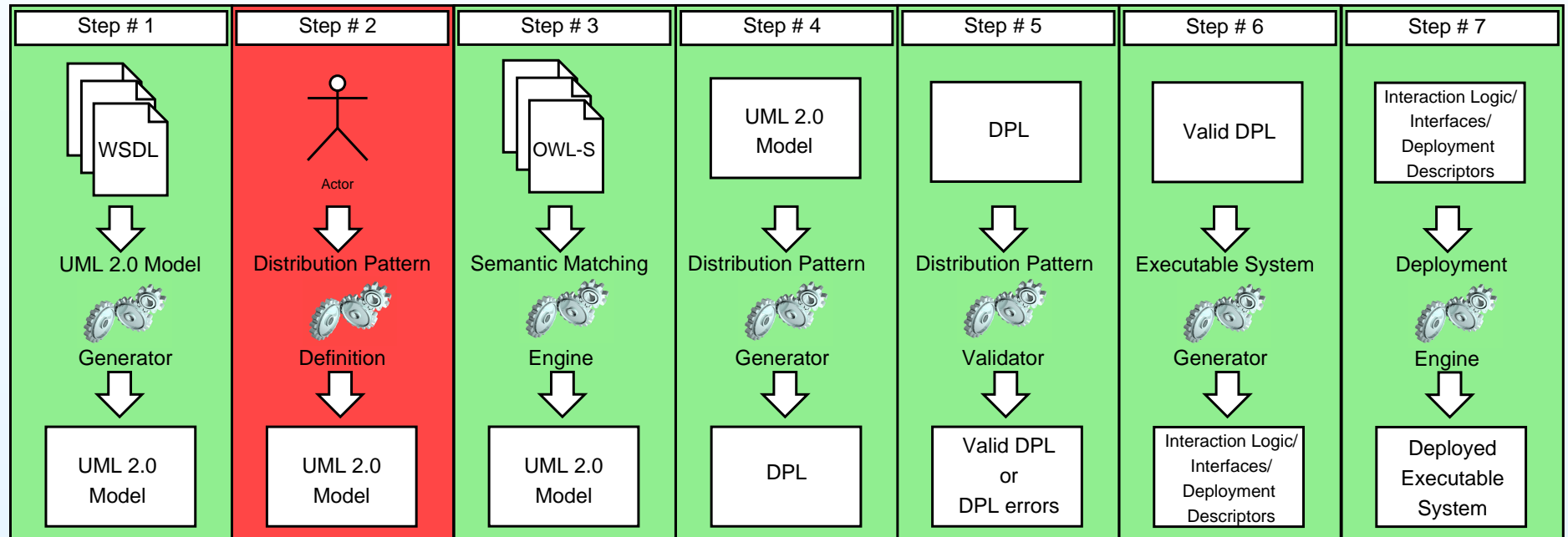
Modeling Approach



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

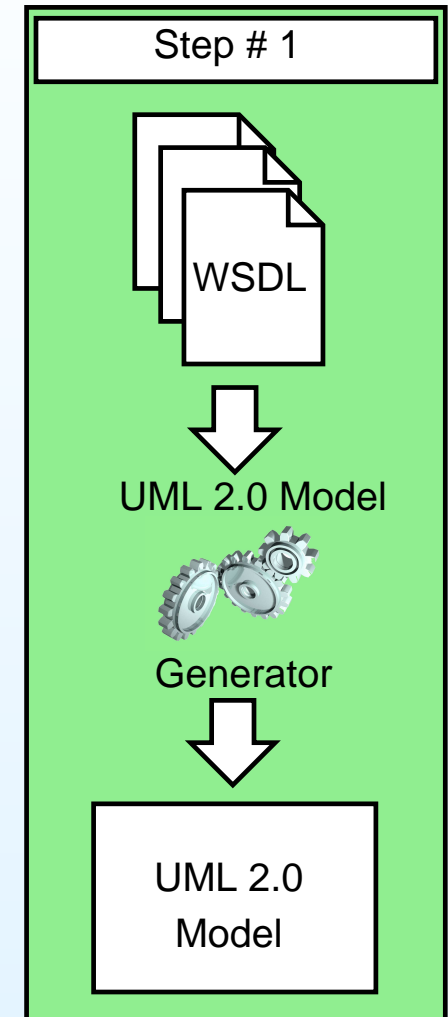
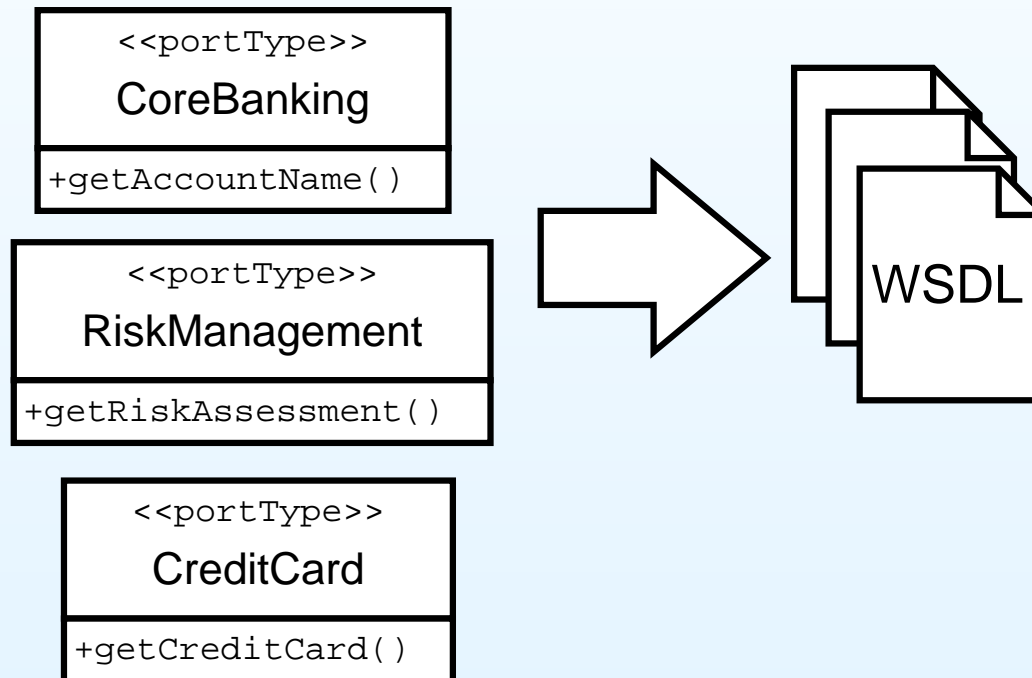


Modeling and Transformation Technique

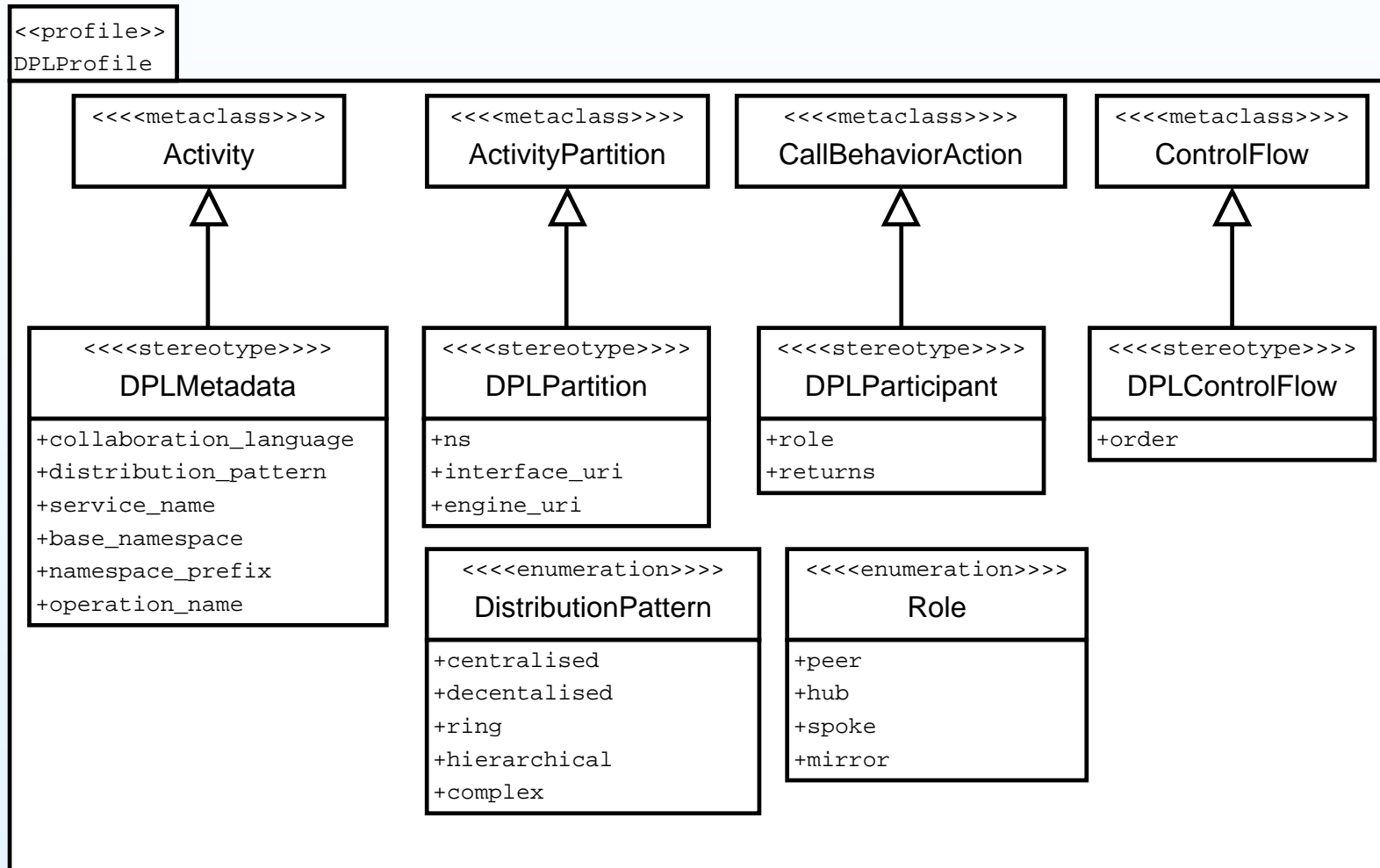


Step 1a - From Interface To UML Model

- Web service interfaces as input
- Generate UML activity diagram

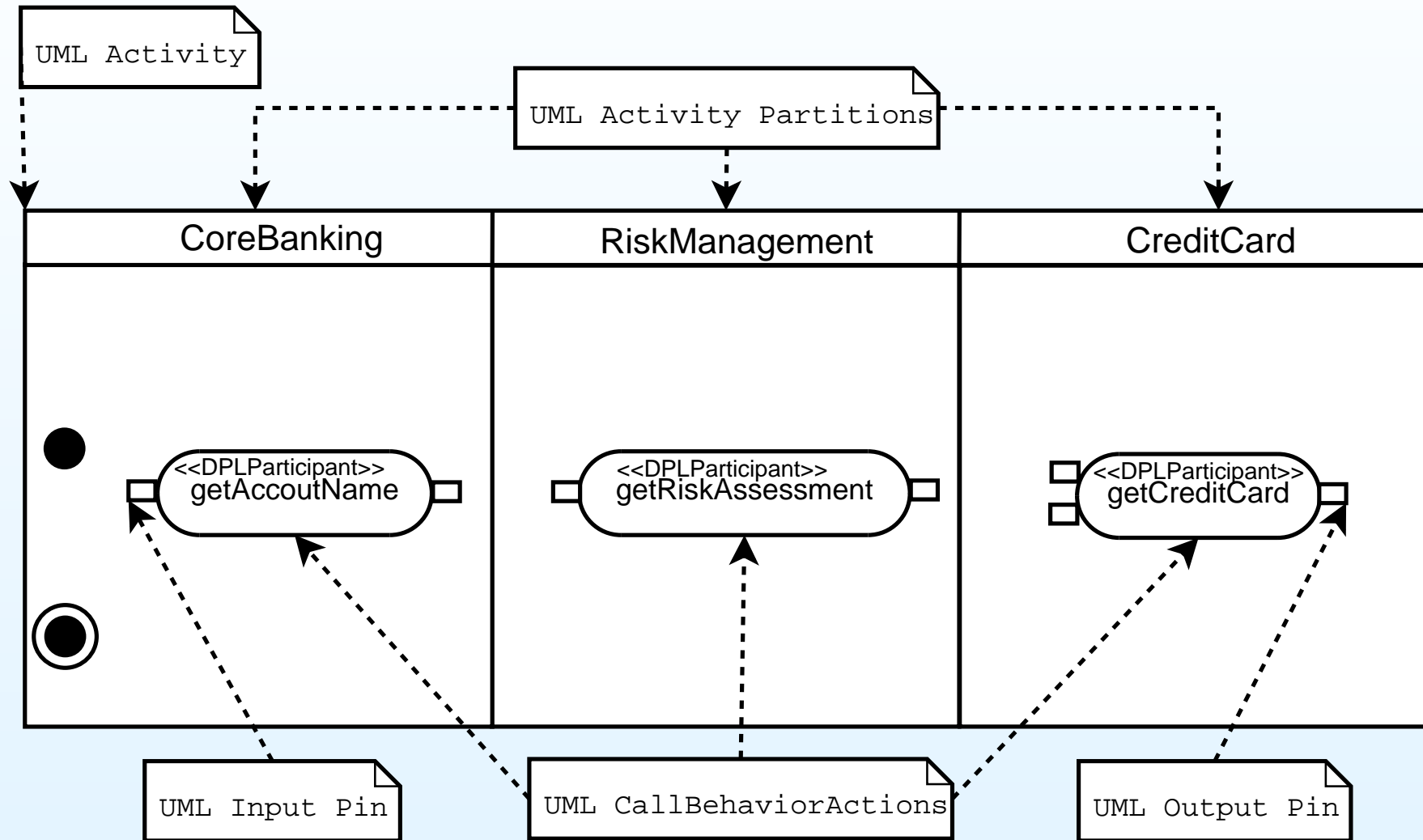


Step 1b - Automatically apply UML Profile



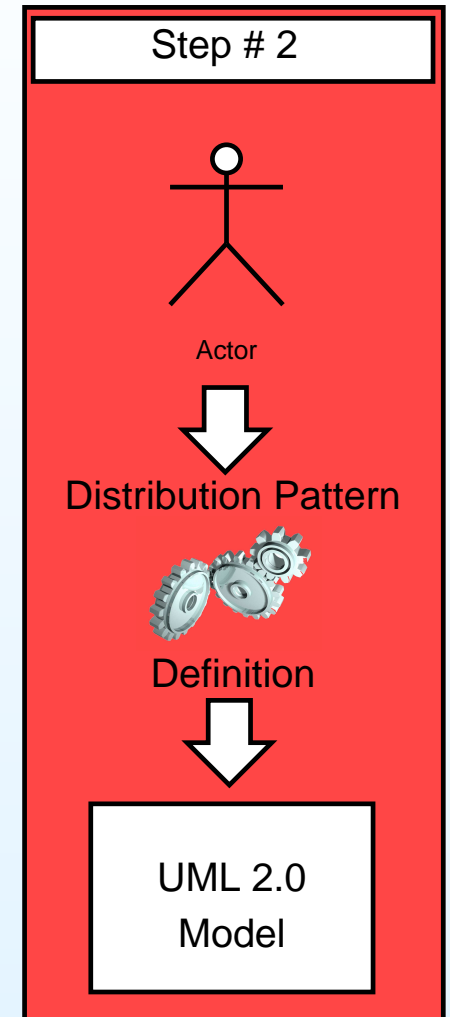
* InputPin and OutputPin elements omitted for space reasons

Step 1c - From Interface To UML Model



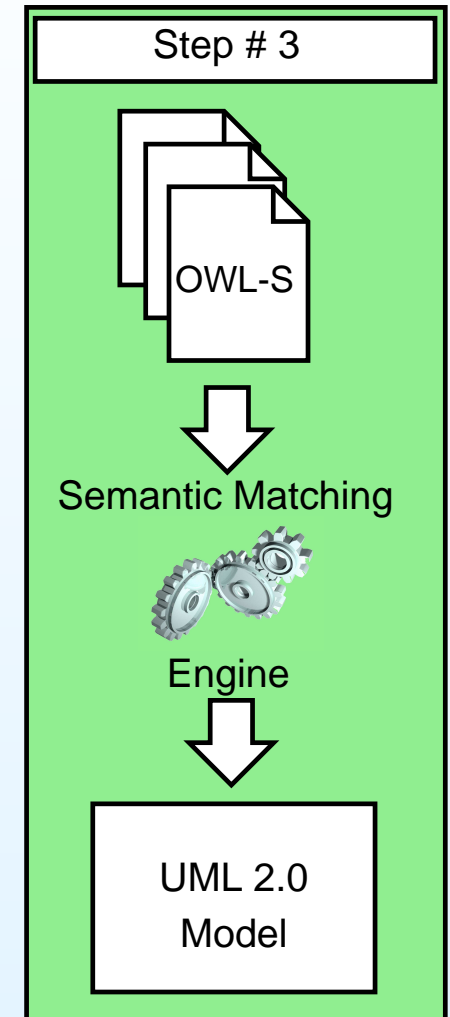
Step 2 - Distribution Pattern Definition

- Manually apply values to UML Profile
 - Select distribution pattern
 - Select collaboration language
 - Select roles for actions
- UML 2.0/XMI 2.0 tool support
 - Rational Software Architect
 - Eclipse UML 2

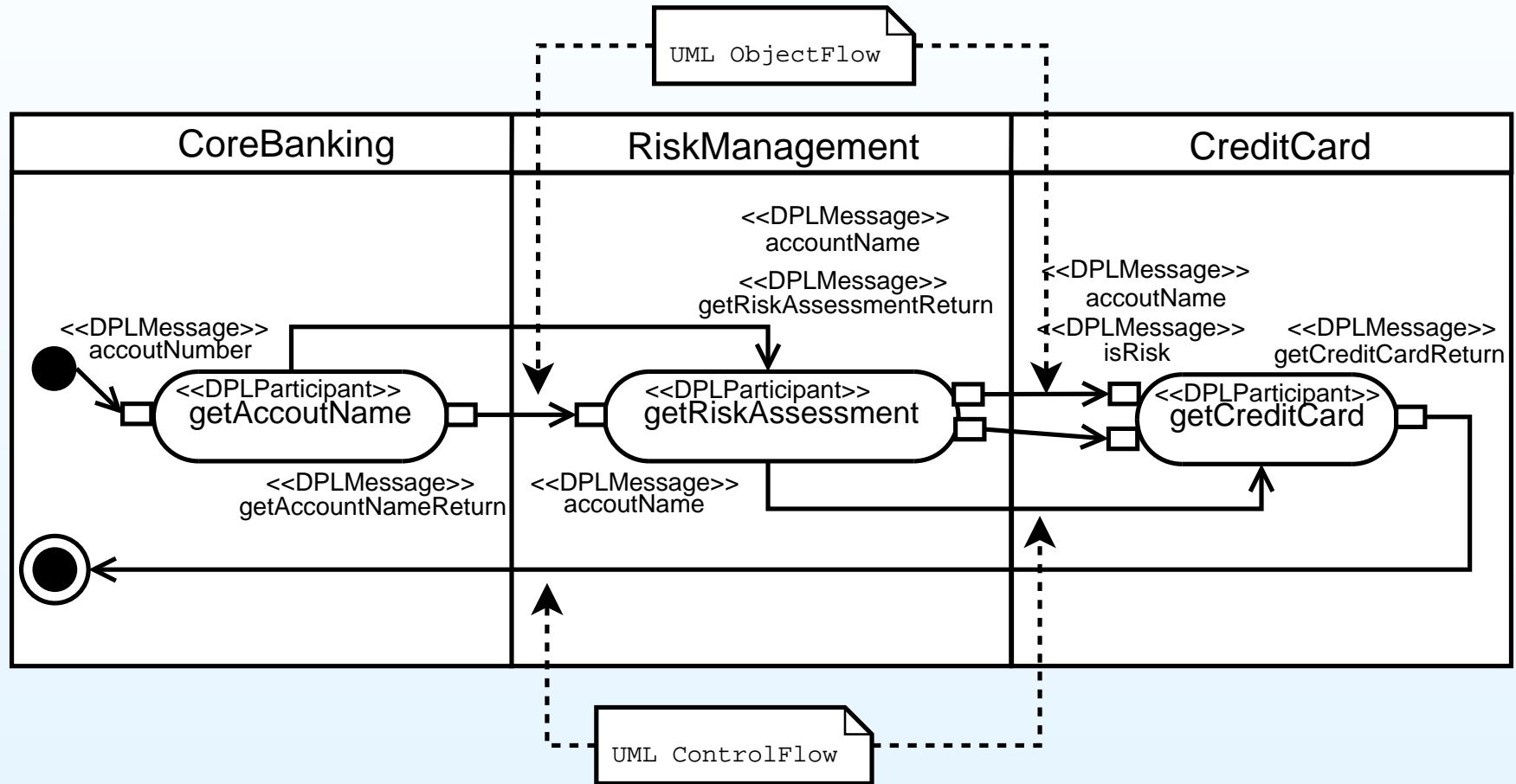


Step 3a - Semantic Matching

- OWL-S descriptions as input
- Process models parsed for matches
- Pre/post conditions satisfied
- Automated connection of ControlFlows
- Automated connection of ObjectFlows

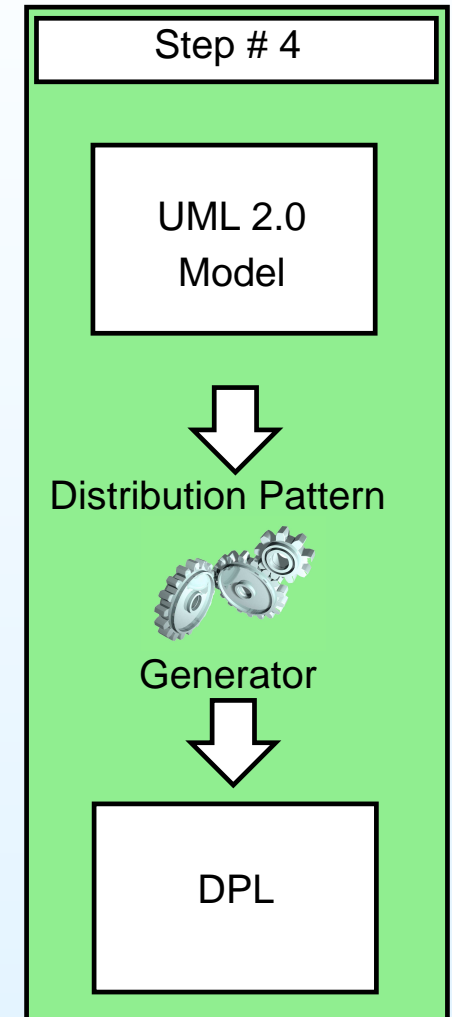


Step 3b - Semantic Matching



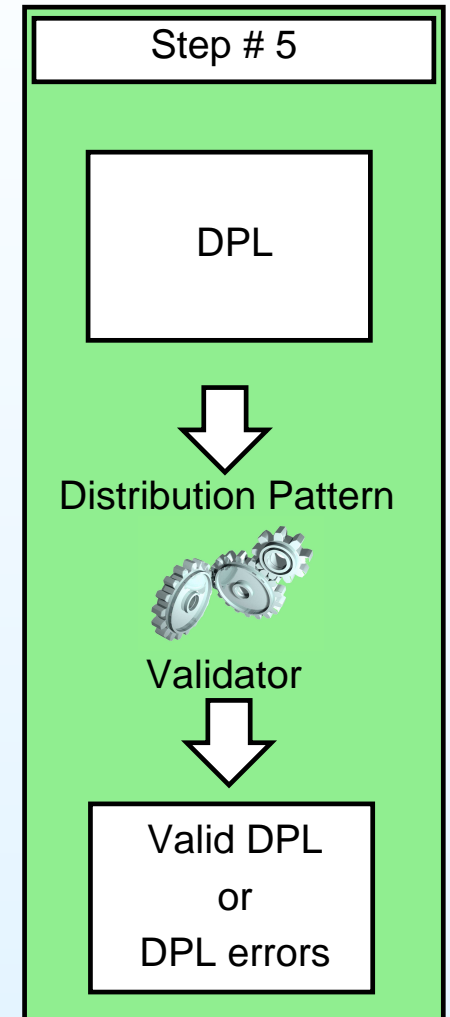
Step 4 - From Model to DPL

- 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



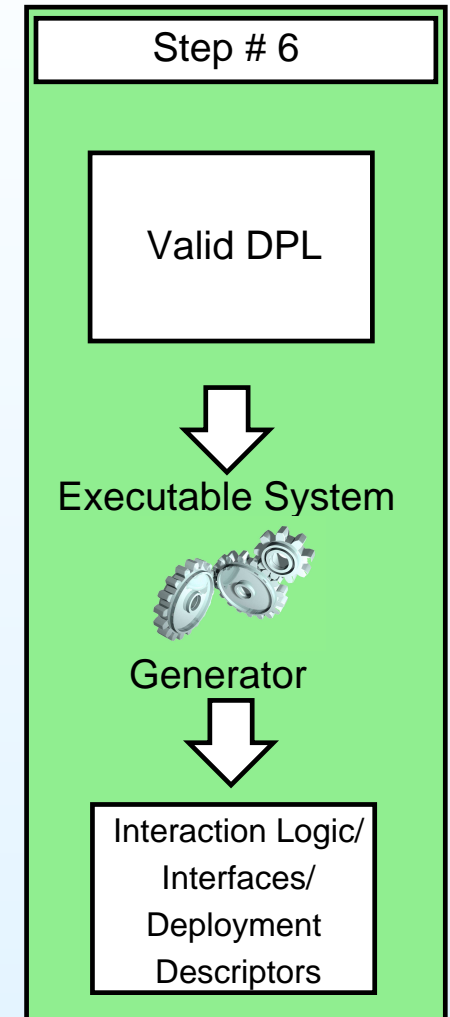
Step 5 - Model Validation

- 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



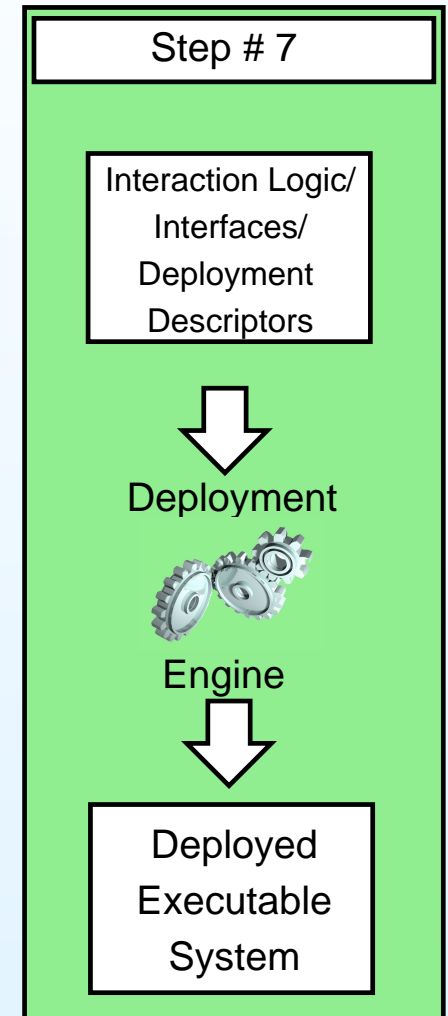
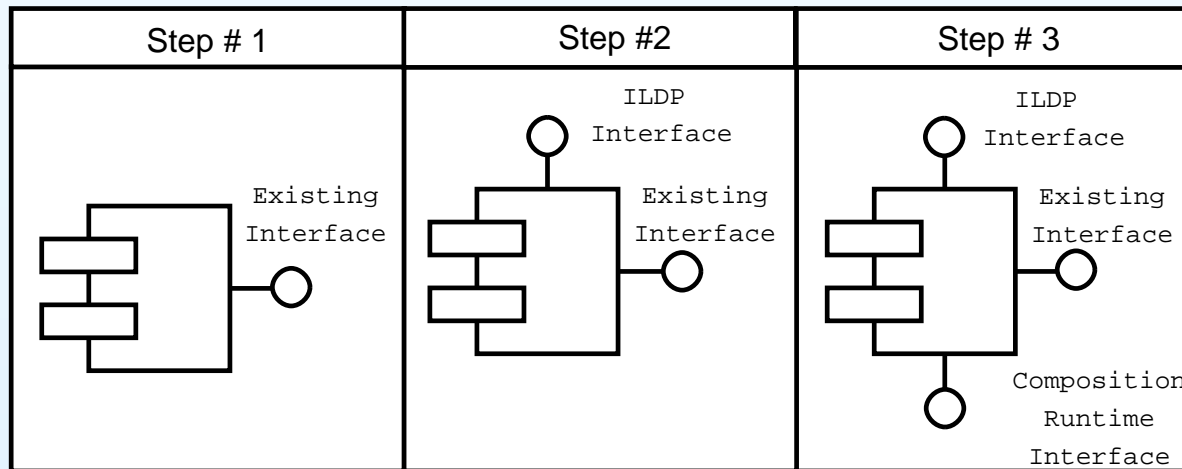
Step 6 - DPL to Executable System

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



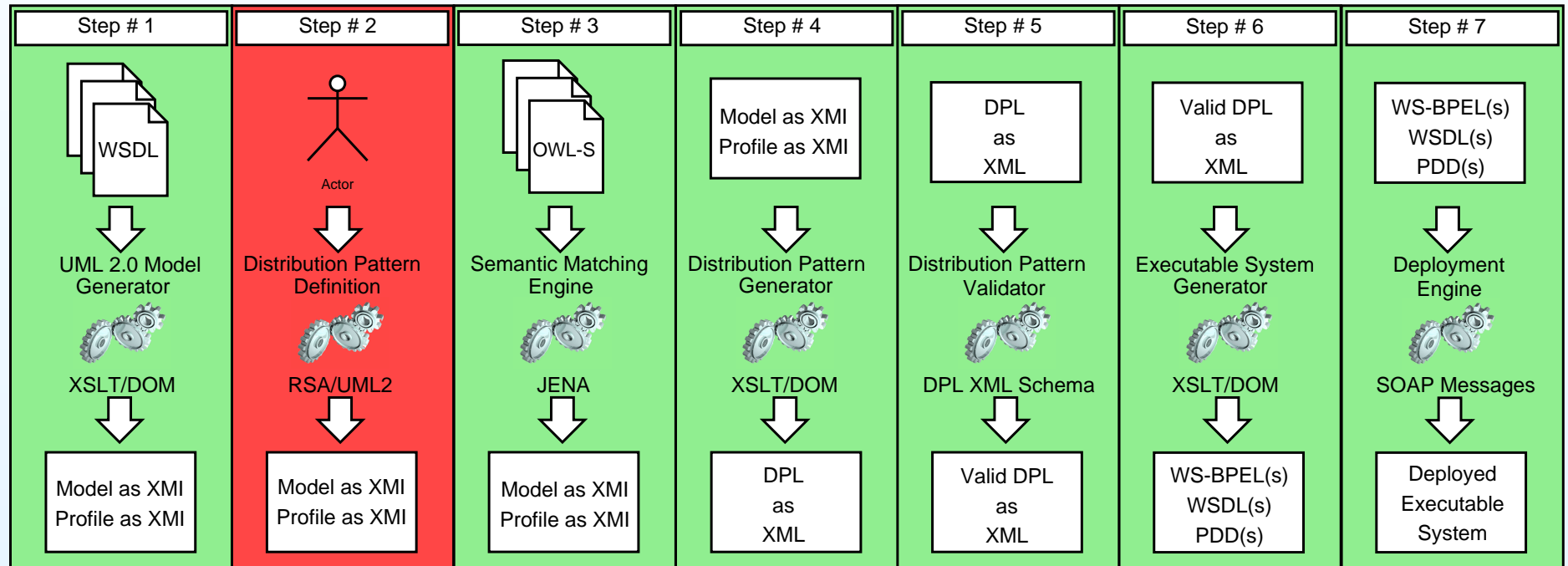
Step 7 - Automated Deployment

- Interaction Logic Document Processor(ILDP) *
 - Enables automated deployment of executable system
 - Participants upgraded
 - Executable artifacts taken as input
 - Artifacts installed & deployed



*ICWE 2006 Presentation - Thursday 11am

Tool Implementation - TOPMAN



Future Work

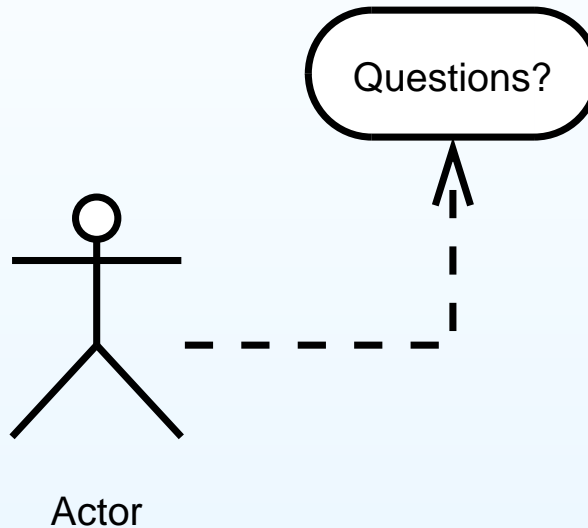
- Complete work on TOPMAN tool
- Develop plugin for AndroMDA
- Extensions to modeling approach
 - π calculus
 - Architecture Description Languages (ADL)
- Unify all three modeling aspects into the tool
- Document distribution pattern metrics

Conclusion

- Motivated new service modeling aspect
 - Distribution Patterns
- Presented modeling approach
 - Standards based
 - Extensible
- Presented seven step modeling technique
 - UML profile extension
 - Utilising semantics for semi-automation
 - Introduced dynamic deployment facility
- 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.

