Model-Driven Engineering of Rules for Web Services

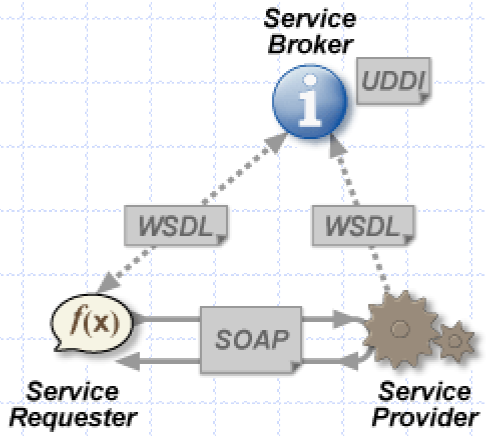
In collaboration with:

Vladan Devedžić, Adrian Giurca, Sergey Lukichev, Milan Milanović, Marko Ribarić, & Gerd Wagner

Dragan Gašević
School of Computing and Information Systems
Athabasca University, Canada
dgasevic@acm.org

Web Services

- What are Web services?
  - software systems designed to support interoperable Machine to Machine interaction over a network
  - frequently just Web APIs to be
    - accessed over a network, such as the Internet, and
    - executed on a remote system hosting the requested services



http://en.wikipedia.org/wiki/Web_service

Web Services

- Characteristics
  - By the nature a simple technology, but the development is not so easy
  - Developers are focus on very low-level details
    - No high-level approach (modeling)
  - Shared by different parties
    - A shared understanding is needed (e.g., ontology)

Web Services

- Characteristics
  - Many aspects to be integrate that were originally not be planned
    - Non-functional requirements (e.g., policies, security, QoS)
  - Dynamic category
    - Should reflect fast business process changes
    - Evolution of Web services and applications
    - Deployed and implemented for different platforms

Developing Web Services

- Our approach
  - **MDE and rules** for Web services
- Advantages of MDE
  - Web services and rules for different platforms
    - Metamodeling of rule languages (DSML)
    - Well-known software modeling languages (UML)
    - Model management –
      complexity of different languages and technologies
    - Model transformations

Rules for Web Services

- Advantages of using rules
  - Business requirements
    - Captured in the form of rules in a natural language
    - Formulated by (non-technical) domain experts
  - Describe behavior of Web services by means of reaction rules
    - Declarative programming (dynamic BP changes)
  - Reaction rules
    - Event-Condition-Action (ECA)
    - Flexible way to specify control
    - Integrate events/actions from the real life

Rule Languages

- Potential technologies around (W3C)
  - Rule Interchange Format (RIF)
    - Identified ten use-cases
    - Examples

A buyer must provide credit card information together with delivery information (address, postal code, city, and country).

A wireless device can transmit on a 5 GHz band if no priority user is currently using that band.

Model-Driven Rule Engineering

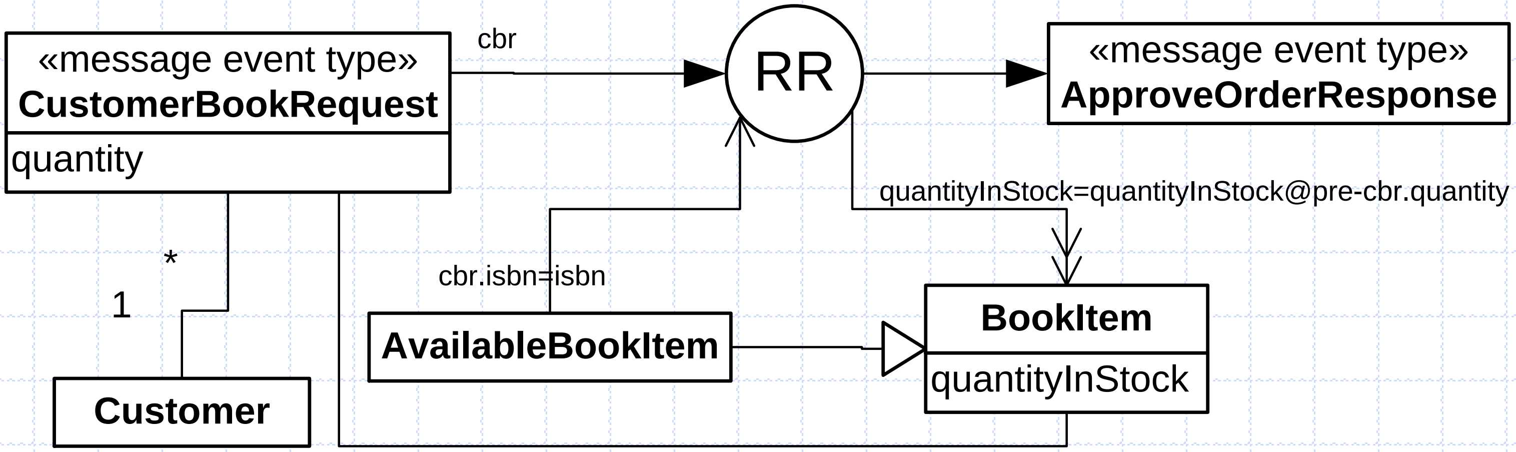
- REWERSE Rule Markup Language
  - http://rewerse.net/I1/
  - Current version 0.5
  - Addresses RIF requirements
  - Organization
    - R2ML MOF-based metamodel for rules
      - An abstract syntax
    - R2ML XML Schema
      - A concrete syntax
    - UML-based Rule Modeling Language (URML)
      - A concrete syntax
    - Transformations

REWERSE Rule Markup Language

- R2ML reaction rules
  - Event-Condition-Action (ECA) rules
  - Statements of programming logic that
    - Specify the execution of one or more actions
    - In the case of a triggering event occurrence and
    - If rule conditions are satisfied
  - Example:
    On customer book request,
    if the book is available,
    then approve order and
    decrease amount of books in stock

URML Reaction (ECA) Rules

Example

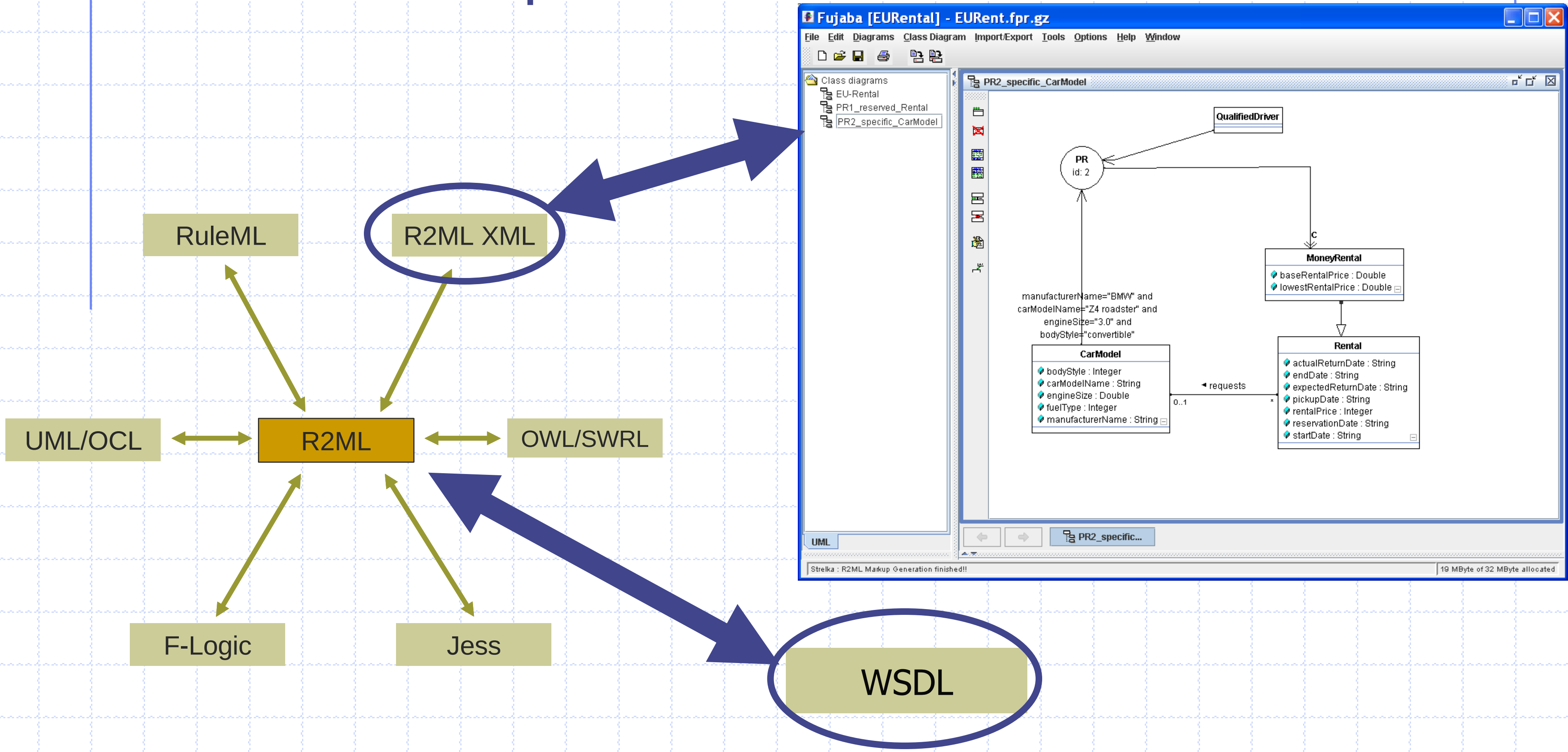


- On customer book request,
  if the book is available,
  then approve order and
  decrease amount of books in stock

Transformations

- R2ML as a pivotal metamodel

Strelka



Strelka : R2ML Markup Generation finished!!

Conclusion

- Summarizing
  - MDE principles to model Web services
    - Modeling services - metamodeling
    - Model transformations
    - Generating to several platforms
  - Rules
    - Close to domain experts and end-users
    - Flexible way to update and integrate business processes
    - Naturally built on top of vocabulary languages

Athabasca University
Canada's Open University

LORNET

Model-Driven Engineering of Rules for Web Services

In collaboration with:
Vladan Devedžić, Adrian Giurca, Sergey Lukichev, Milan Milanović, Marko Ribarić, & Gerd Wagner

Dragan Gašević
School of Computing and Information Systems
Athabasca University, Canada
dgasevic@acm.org