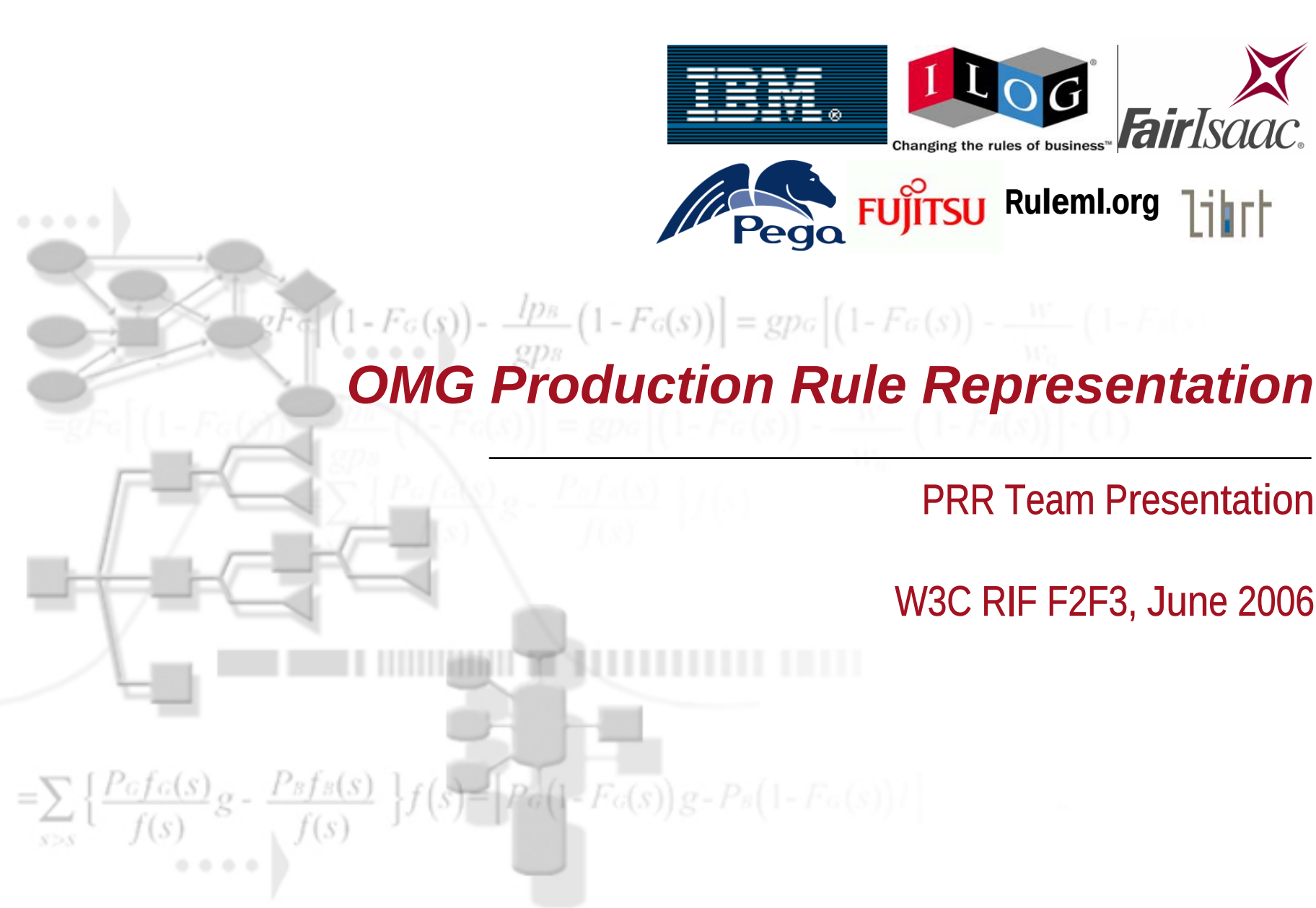


OMG Production Rule Representation

PRR Team Presentation

W3C RIF F2F3, June 2006

Intro: OMG PRR

- Rule standard based on OMG model
  - Mainstream IT focus (looking at near-term)
  - Based on UML mantra
    - Software models
    - Metamodels (including MOF – metamodel standardisation)
  - Large vendor community + specialist organisations
    - Platform, domain focuses
    - Mostly traditional IT types
      - Rules are not considered platform, but business domain
      - Other OMG stds eg QVT
        define declarative rules but in a custom format
  - Standards based on existing implementations (+ inventions)

Intro: PR rule engines

- Approach for implementing flexible business rules in IT systems:
  - **Production rules** executed by a **rule engine**
    - Declarative “If.. Then..” rules
    - Rules defined in sets (rulesets)
    - High performance algorithms (eg Rete)
    - Common execution API (see Java JSR-94)
  - Rules represent behaviour = actions (data updates, invokes…)
  - Powerful deductions / inferences can be made
  - Rules can be changed independent of program logic
  - Users include government, insurance, retail, engineering…

Problem: What is the standard for rule engines?

P R R
rule standard

- Today: non-standard approaches used to represent production rules:
  - Vendors' own representations + associated XML schema
    - Fair Isaac Blaze Advisor (SRL), Ilog JRules (IRL) LibRT (RBML), JESS, DROOLS, etc
  - Vendor-specific standards
    - IBM BRML, Ilog SRML, Haley HRML etc
  - Domain-specific standards
    - ACORD SPX insurance validation rules, XBRL business report rules, etc

This is a problem! Difficult to interchange, little understanding, fear of lock-in…

Solution: Production Rule Representation

- 1st general standards body to address rule standards – OMG – via its Business Rules Working Group
- Production Rule Representation proposed as a cross-vendor (of rule engines) rule modeling representation
- Consortium of developers and supporters
  from the
  rule vendor community (Fair Isaac, Ilog, etc)
  + academic community (RuleML.org)
  + related vendor community (BPM, UML)

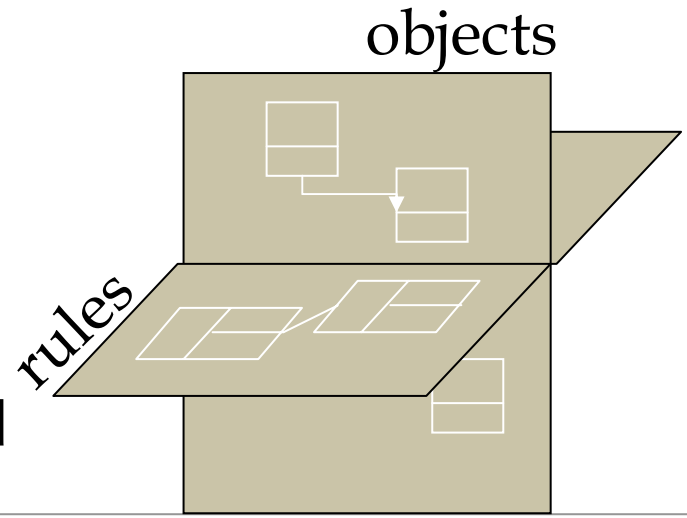
What is PRR?

1. Formal model for production rules
   - Uses OMG MOF, defined in UML
   - Extends UML
   - Provides an XML format (XMI)
2. Vendor-neutral UML-friendly rule representation
   - Intended to be defined via tools, not business users!

OMG PRR is expected to be submitted to OMG later in 2006 as part of the OMG standards process

PRR 1.0 Goals

- Metamodel for production rules:
  - 2 rule “semantics” considered:
    - Forward chaining inference rules (e.g. Rete-model)
      - For commonly-used PR rule engines
    - Sequentially processed procedural rules (e.g. scripts)
      - For tools that separate out simple business logic as non-inference production rules
- Interchange for rule modeling via XMI
  - Import / export rules between UML tools and BRMSs
- UML: production rules to be considered as 1st class citizens alongside objects



PRR 1.0 – Vendor Applicability

- Today:
  - Executable Rule Engine vendors e.g.
    - Fair Isaac Blaze Advisor
    - ILOG JRules
    - IBM Websphere
    - …
  - UML tool vendors
    - IBM Rational
    - …
- Future:
  - Software technologies that need to co-exist with executable rules
    - BPM / workflow
    - EAI / middleware
    - MDA / CASE
    - SBVR-type documented business rules

PRR 1.0 - out of scope

1. Rule Management
   - Business syntax for production rules
   - Metadata, templates etc
2. Rule / Decision metaphors
   - Explicit decision tables and trees
   - Other decision visualizations that can map to production rules
3. Other rule types
   - ECA rules
   - Formal logic
   - Constraint-based reasoning
   - Data relationship (ER model) rules / ontological rules
4. Other production rule semantics
   - Backward chaining inference rules
   - Ruleflows (e.g. activity diagrams for ruleset orchestration)

PRR 1.0 – scope creep

- To implement a PR metamodel, need to define a generic rule metamodel for future extensions to other rule types
- Therefore PRR *includes* a generic rule metamodel (as no such thing exists in OMG / UML today)
- PRR must also be compatible with non-UML representations
  - Expression (condition + action) language used outside of UML
  - Object models mapped from UML class diagrams
- PRR therefore defined at 2 levels
  - PRR core: general Rule + PR model
  - PRR OCL: PRR core + extended OCL expression language

5-minute Guide to OMG: MDA

P R R
rule standard

Model Driven Architecture (MDA)

Computation Independent Models (CIM)

Business Models

Rarely considered by the UML community

MDA is the logical successor to CASE

Model Driven Architecture (MDA)

Computation Independent Models (CIM)

Business Models

Technically: these models in MDA are also “UML” – they are described using UML

Platform Independent Models (PIM)

UML Models

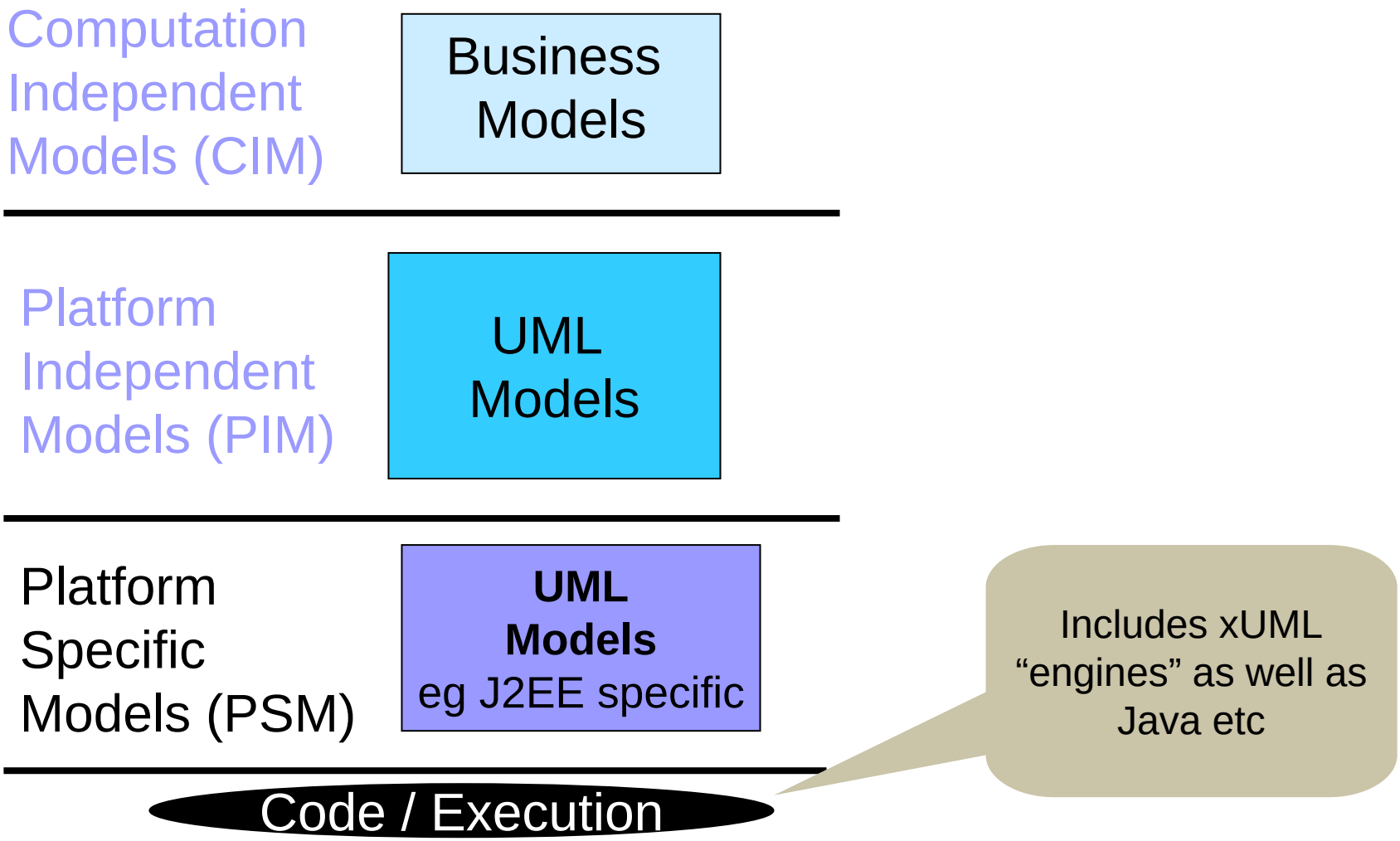
Classic models: class diagrams, activity diagrams, etc

Can include detail like action language for behavior, OCL for constraints…

“Picture to code” in as few steps as possible

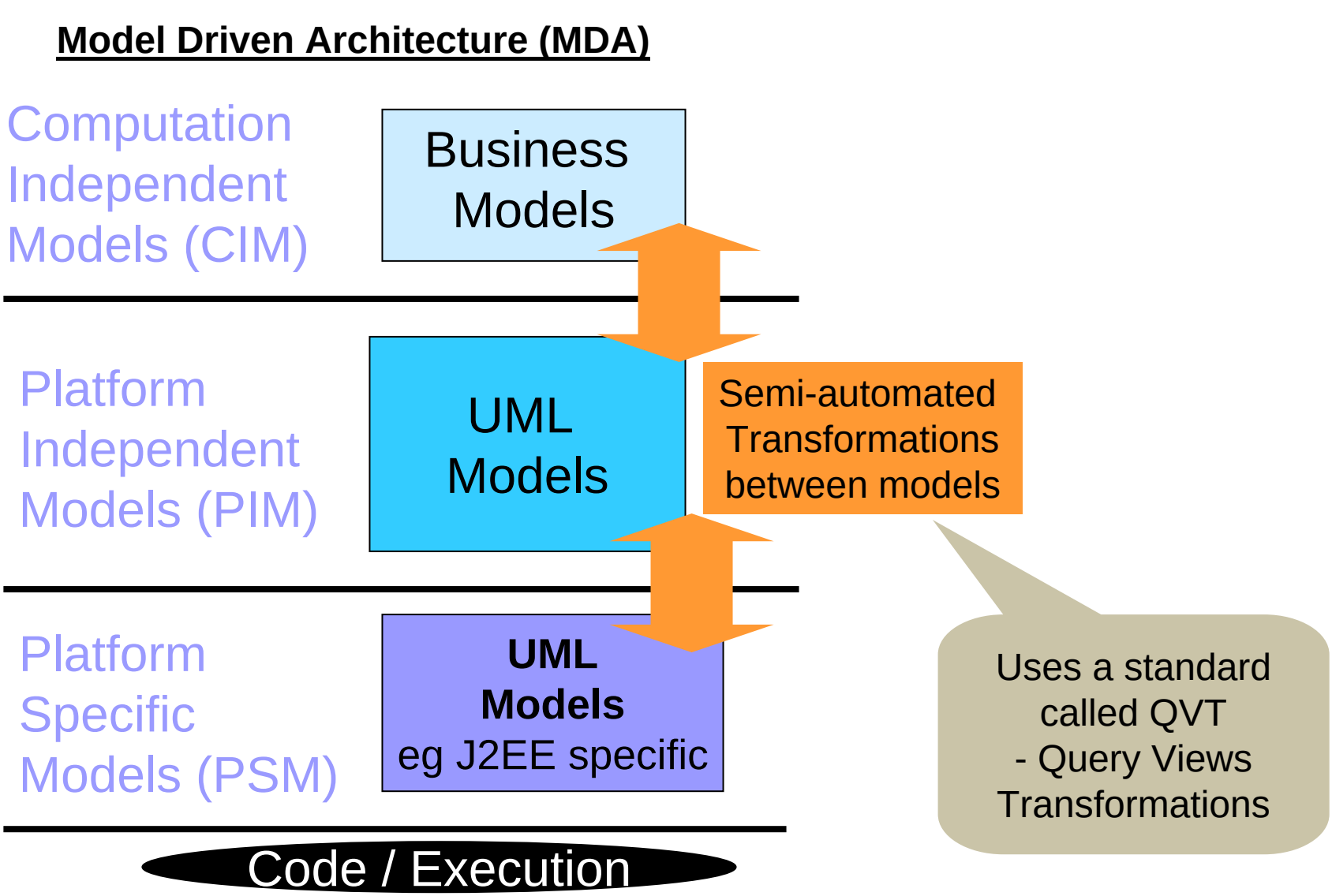
P R R
rule standard

Model Driven Architecture (MDA)

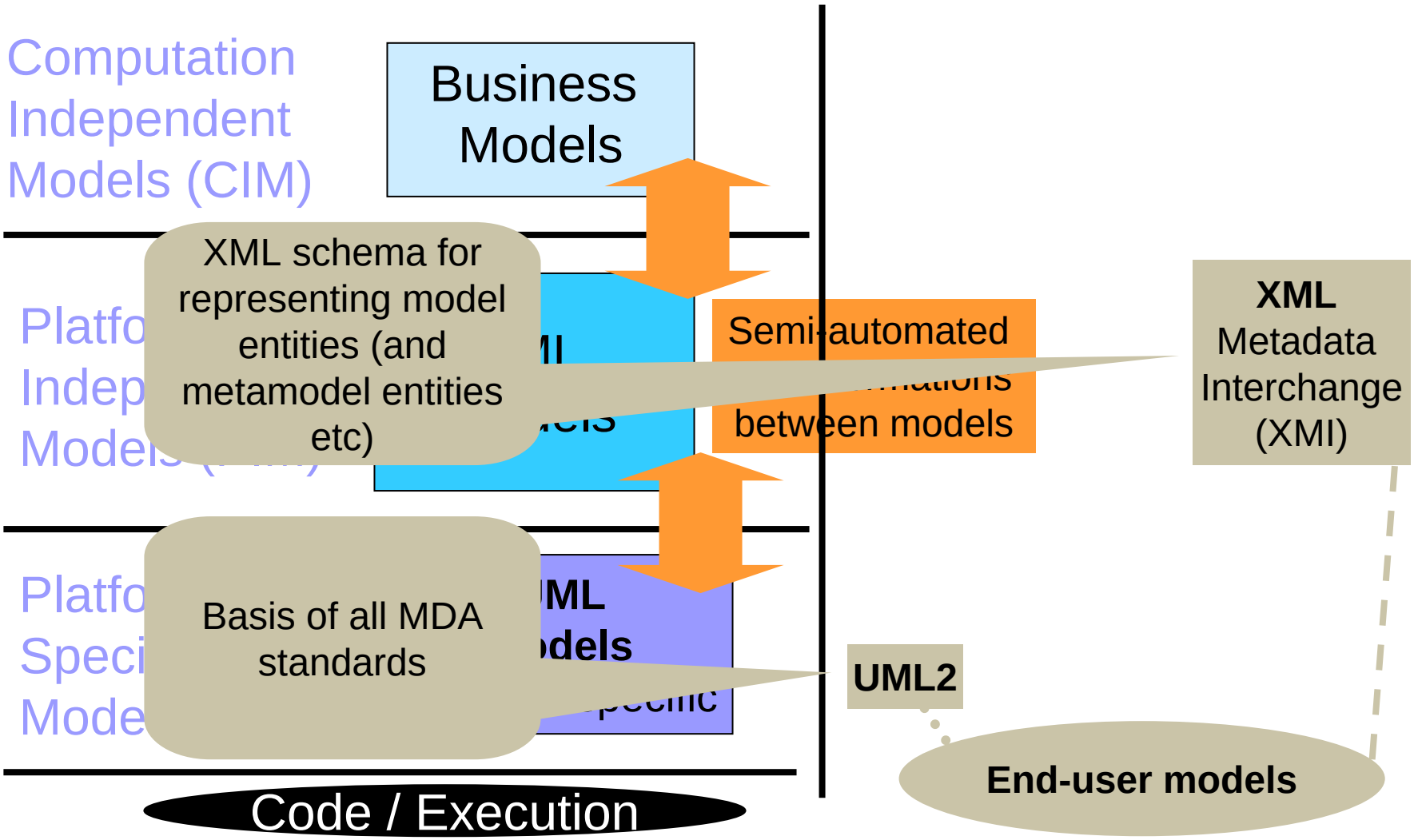


P R R
rule standard

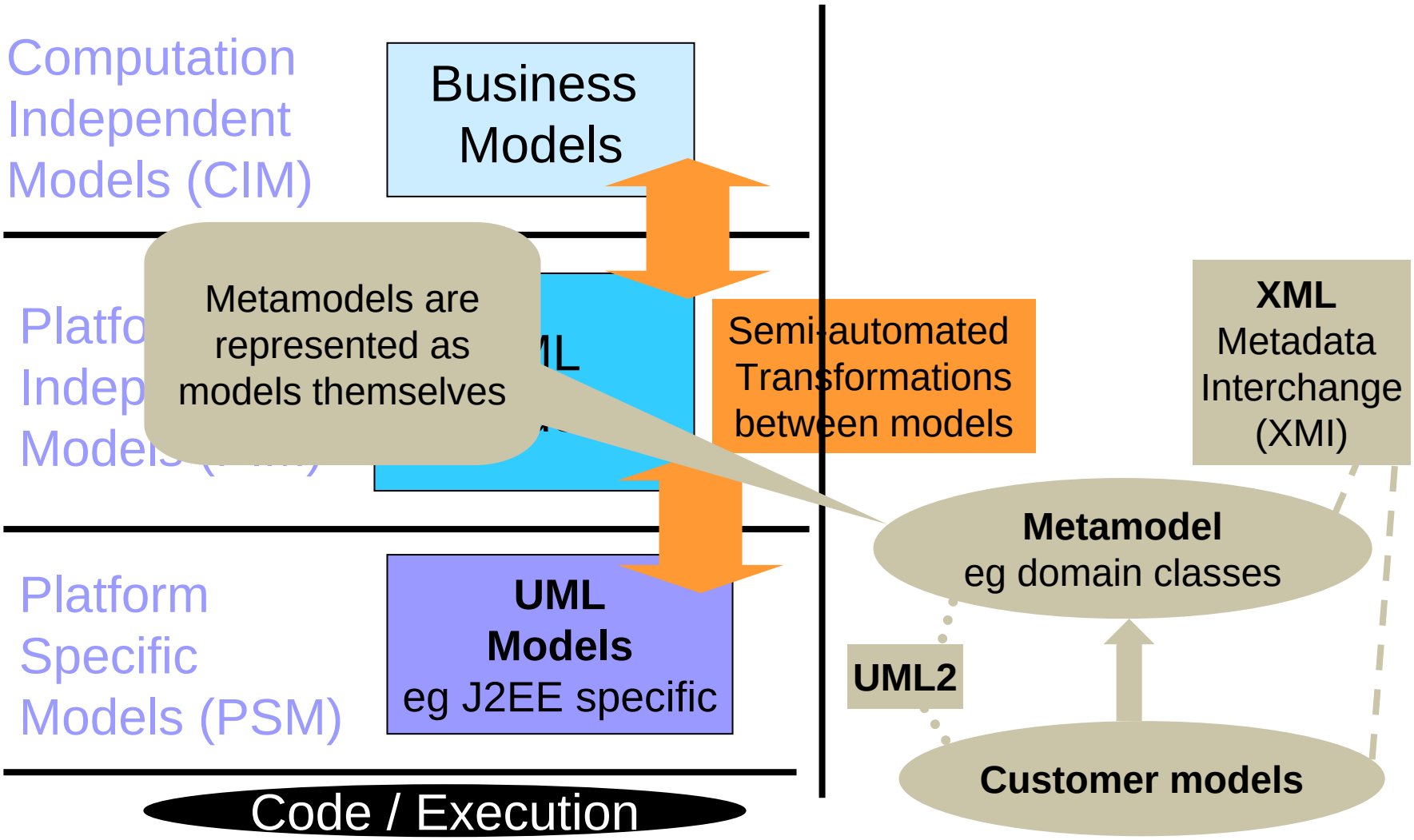
Leading to things like “Executable UML” ☺



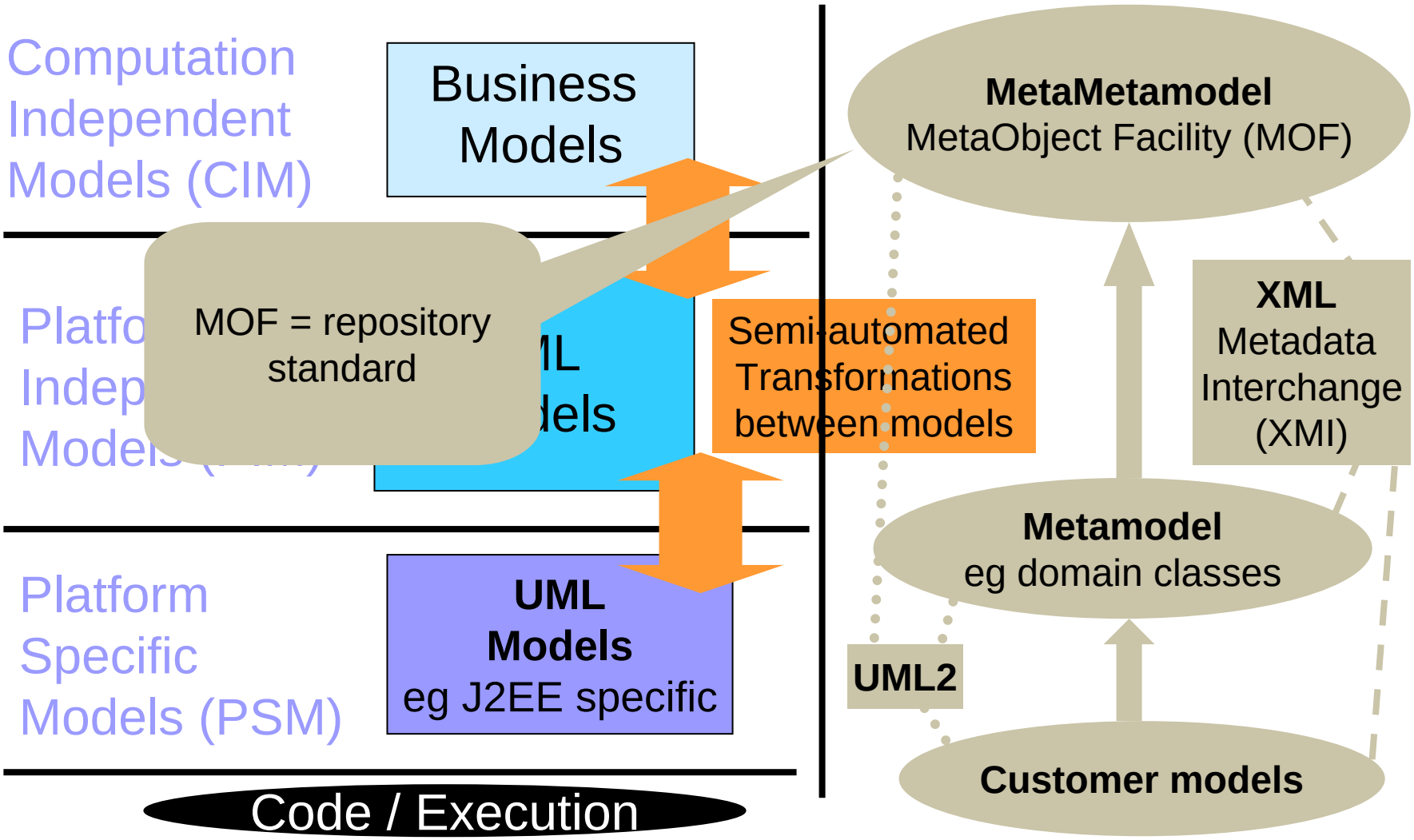
Supported by UML and XMI



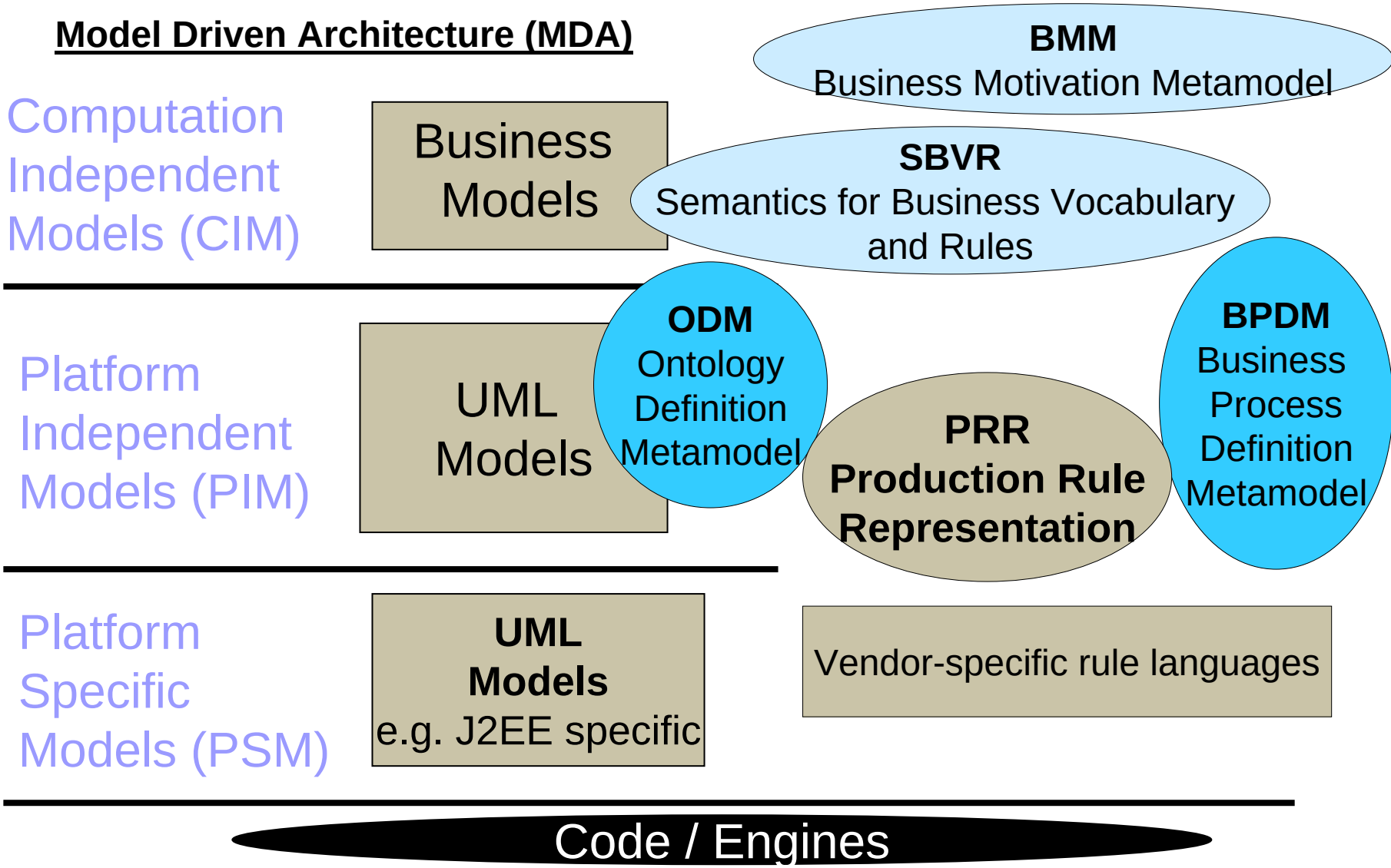
OMG main activity= defining metamodels



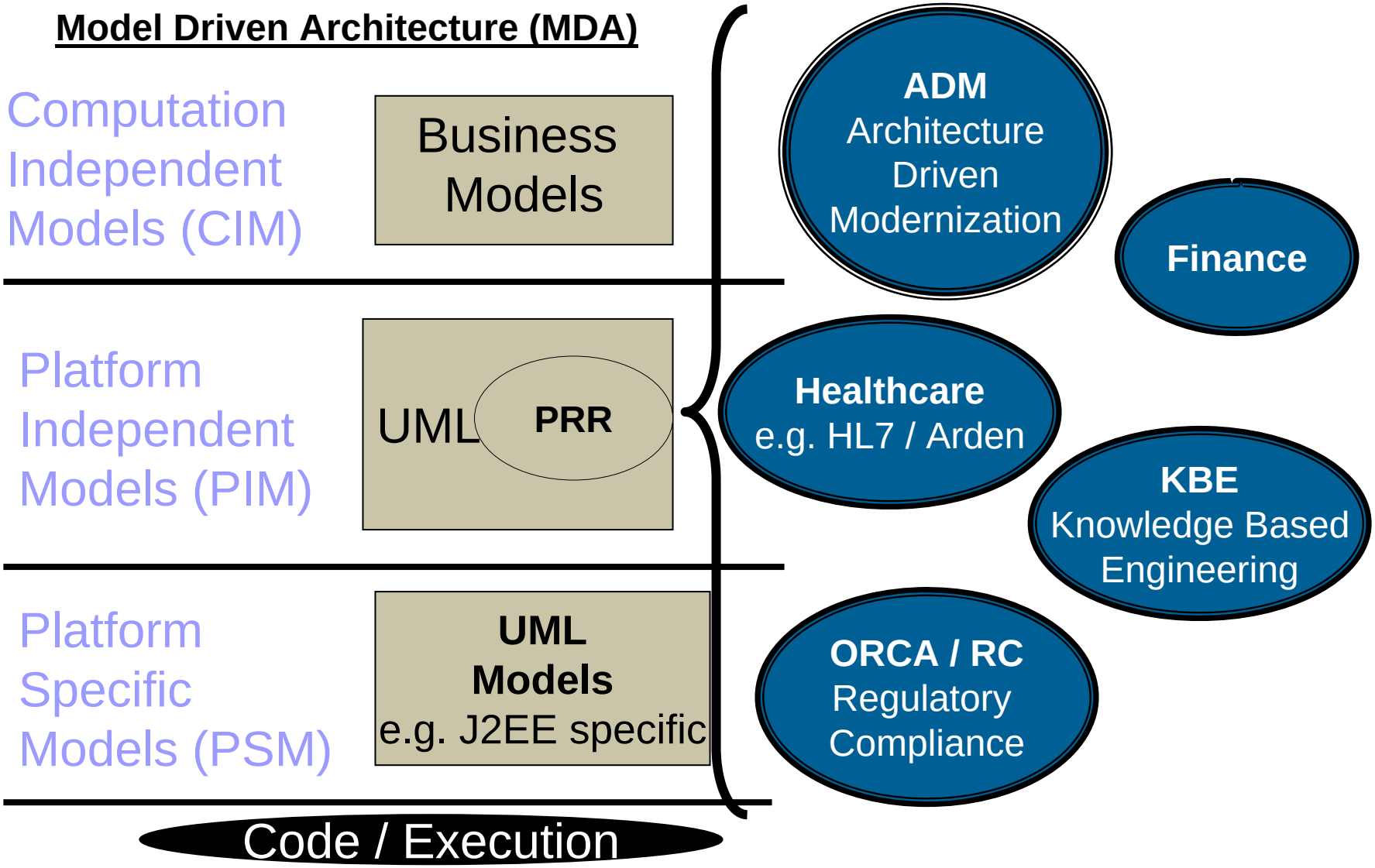
Metametamodels too! All defined in UML



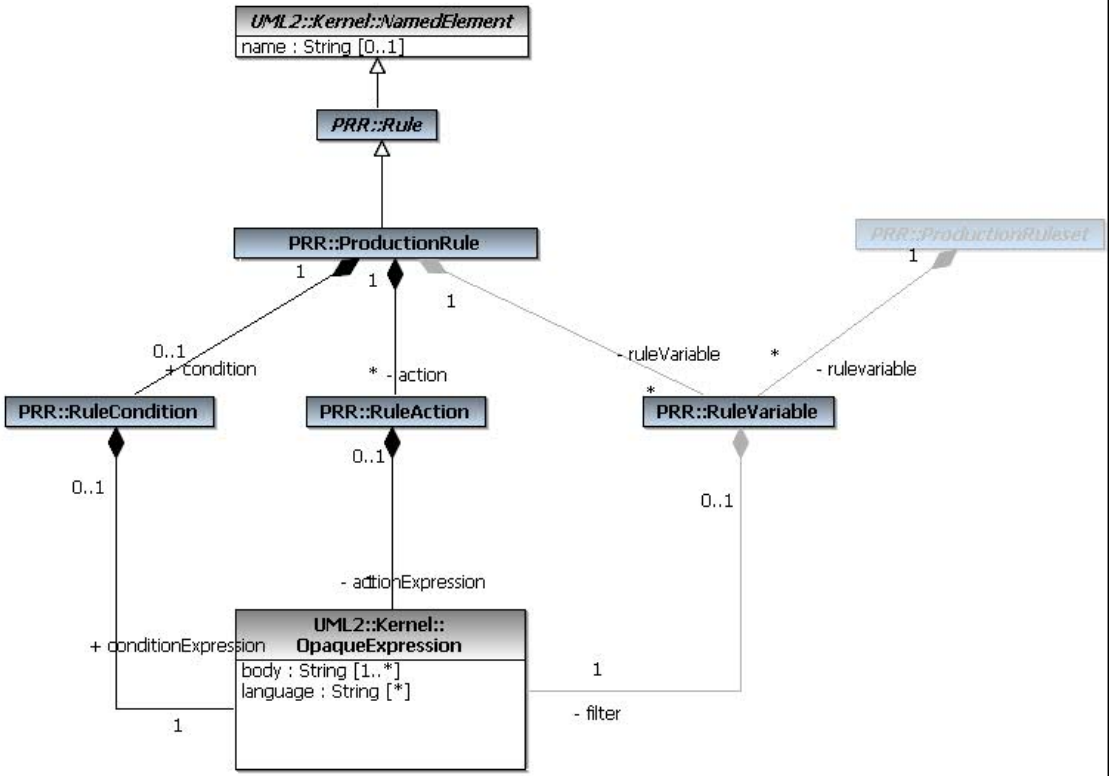
OMG MDA and PRR



OMG MDA: other domains that relate to PRR



Simplified Rule Model



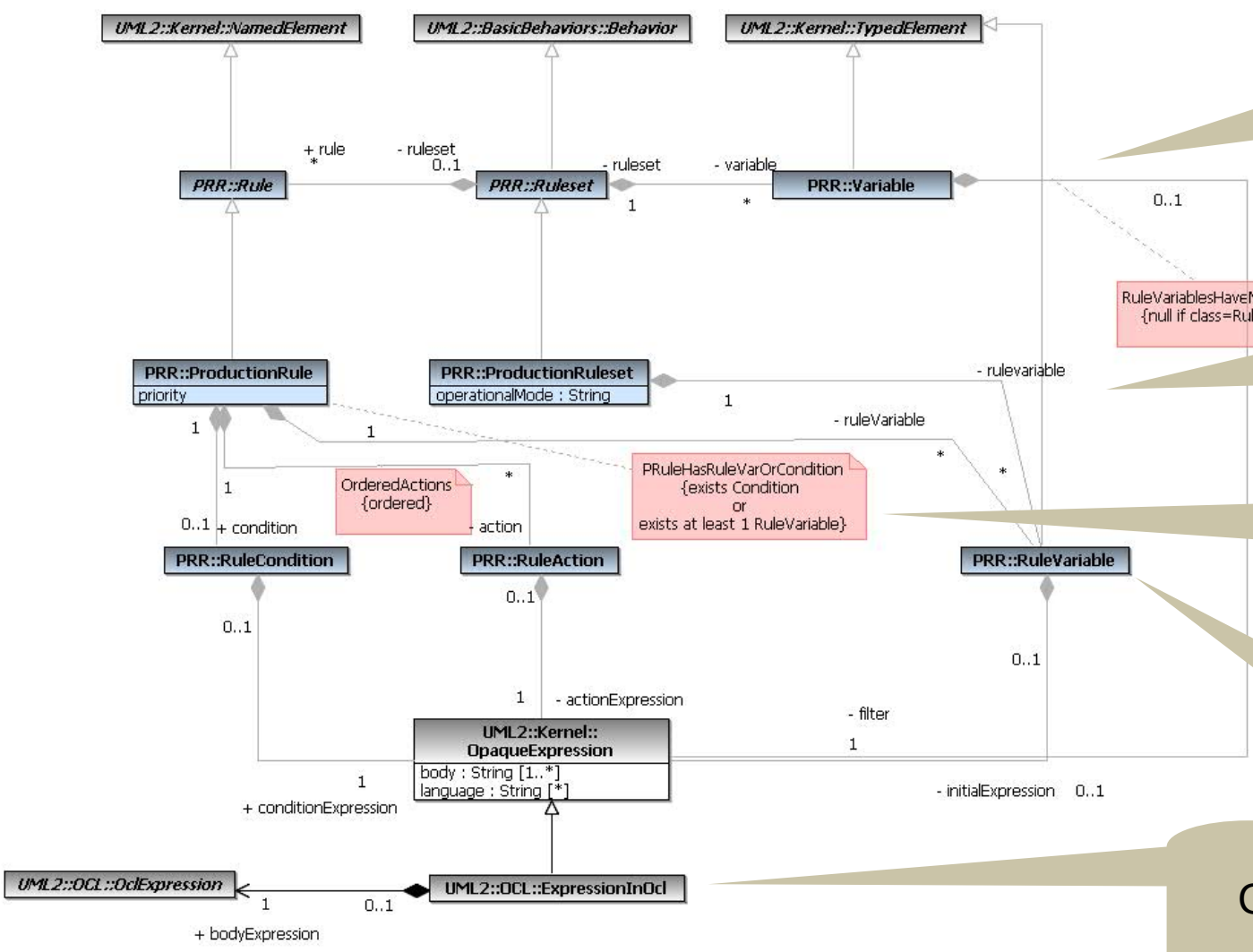
PRR Core	PRR OCL
Generic metamodel	OMG-specific subclass
Expression language a plug-in	OCL2 customized (includes actions)
General production rule model	Per RFP requirements

Production Rule definition

P R R
rule standard

- In Rete world, a rule is MORE than conditions + actions: introducing the `RuleVariable`
  (called a `pattern` in Blaze Advisor, a `variable` in JRules…)
  - Matches to a range of objects in scope / in a collection + an optional filter
  - Defined per rule (JRules) or per ruleset (Blaze Advisor)…
  - Effect:
    - Rule is effectively duplicated at runtime for each tuple
    - Note the RuleVariables can be changed by rule firings!
    - Cf pattern matching in PROLOG
  - Allows for powerful rule statements

Production Rule metamodel overview



Generic rules, rulesets and variables

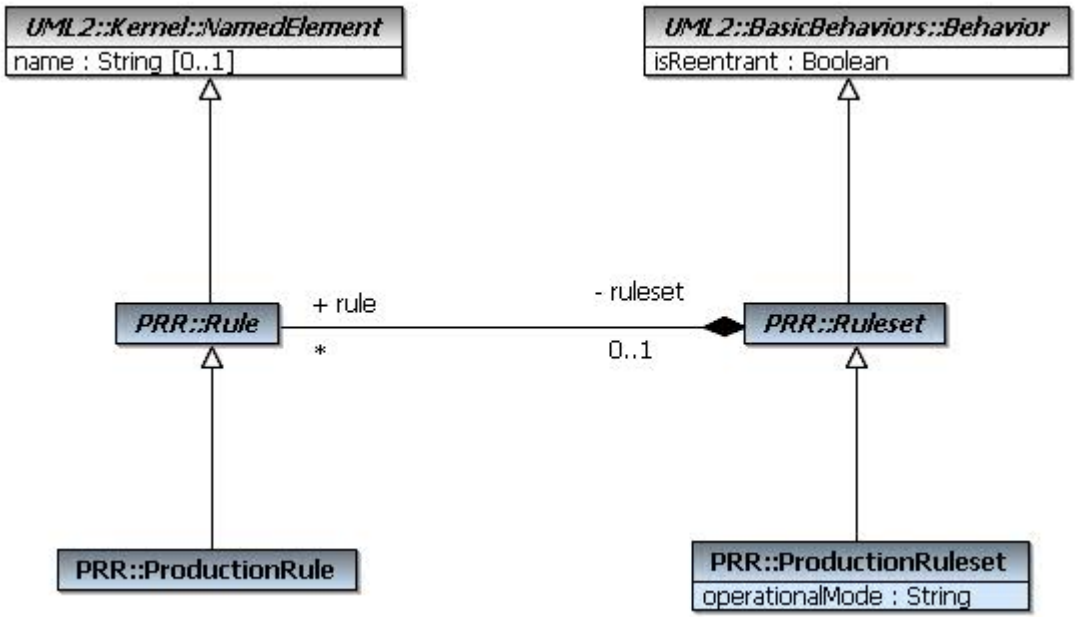
Production rules, rulesets

Rule Condition, Action (lists)

RuleVariables to create Rete bindings

(PRR OCL): OCL = expression language

Ruleset and Rule metamodel

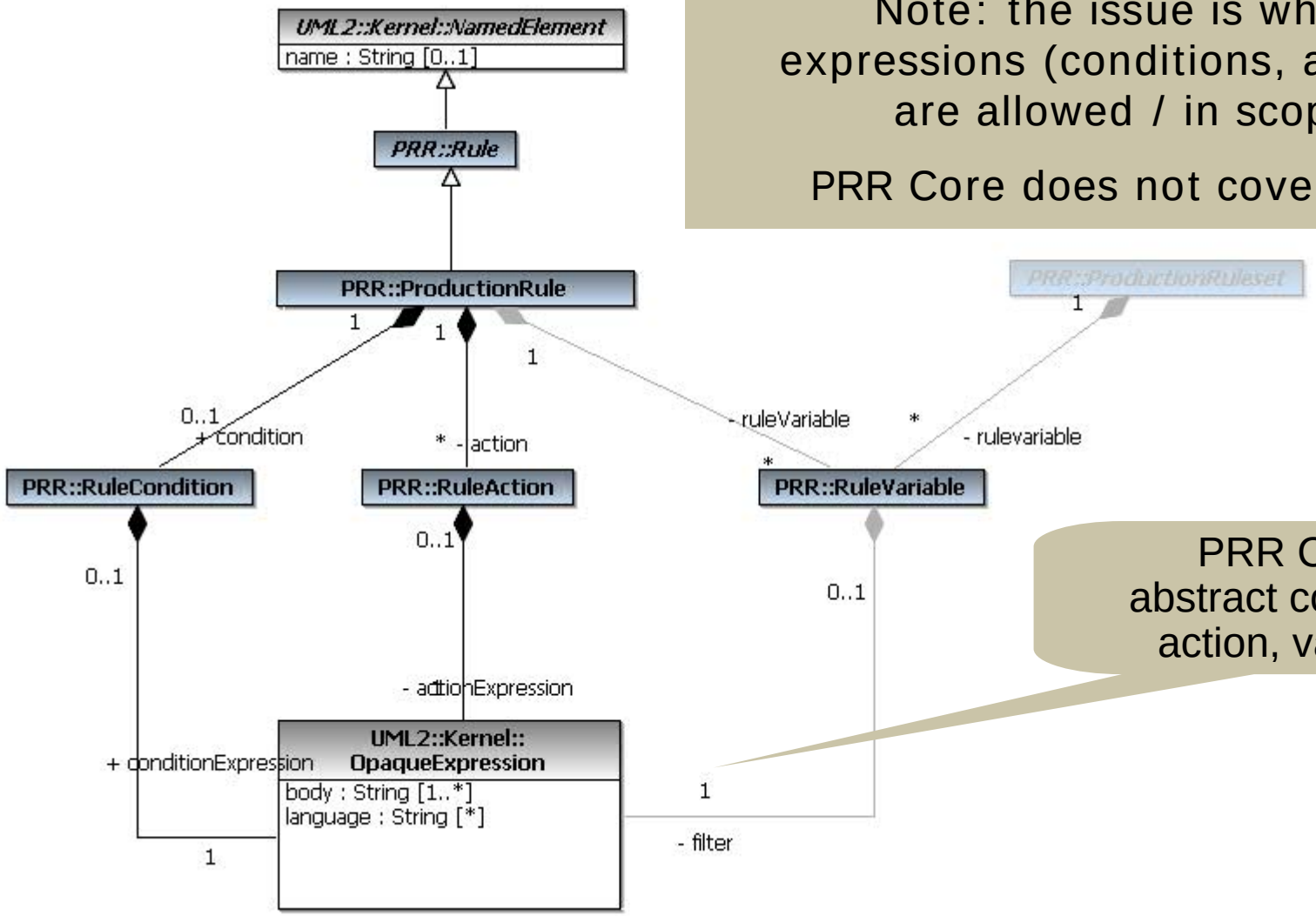


Generic rules, rulesets

Specialization for production rules

Note: clearly the generic model would benefit from being truly generic, with PR just as one subtype of rules

Production Rule Metamodel

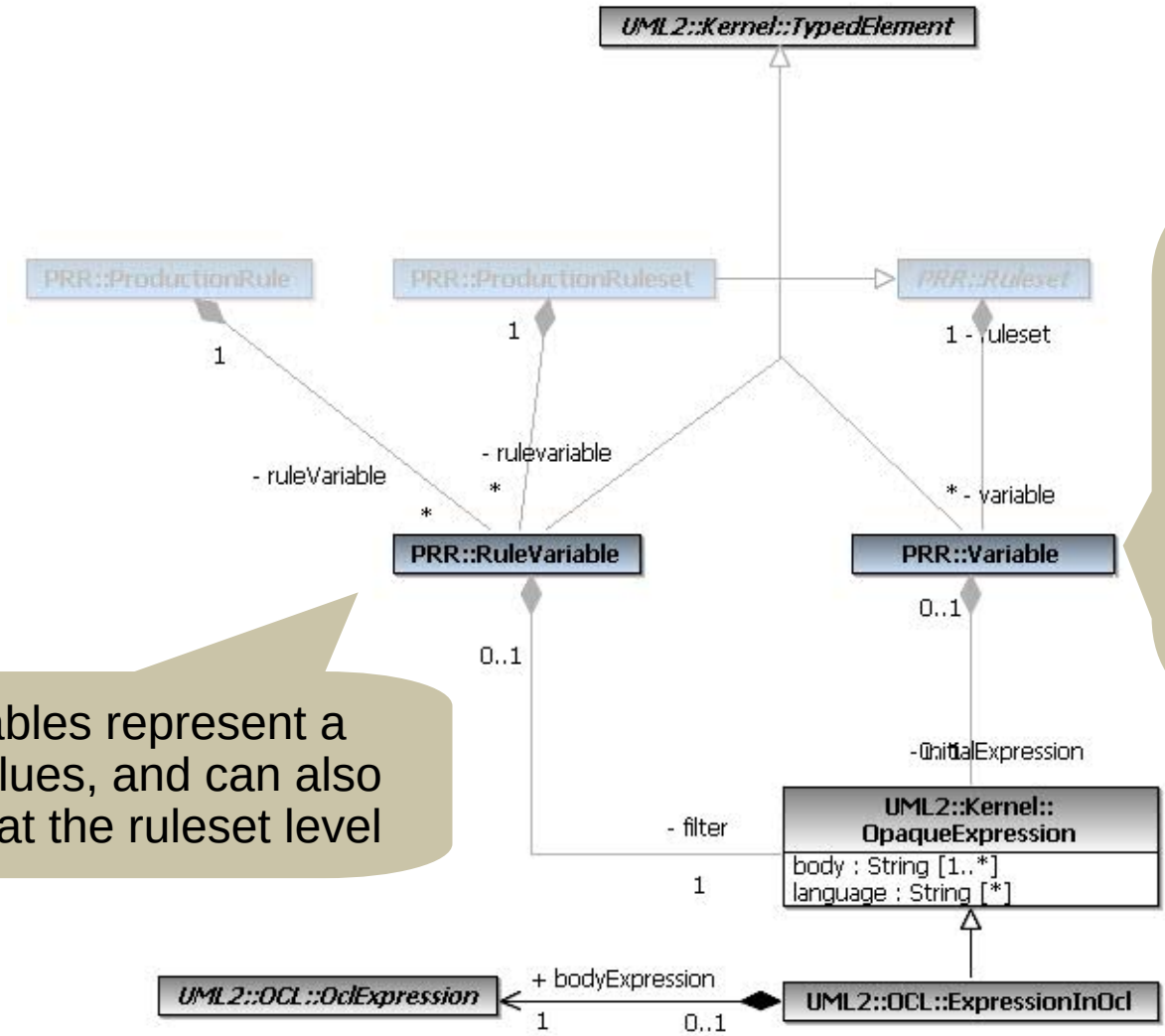


Note: the issue is what expressions (conditions, actions) are allowed / in scope

PRR Core does not cover this.

PRR Core: abstract condition, action, variable

Variable and RuleVariable metamodel



RuleVariables represent a range of values, and can also be defined at the ruleset level

Variables represent a single value used in rule evaluation, although this could be a collection

Comparison with other Rule Standards

- RuleML
  - Contributors to PRR (in RIF: Gerd Wagner, Said Tabet)
  - PRRRuleML (Said Tabet)
    in development as a RuleML implementation of PRR Core
- OMG SBVR
  - MDA CIM standard for documenting business rules
    with no IT context / assumed
  - In “finalisation”

Differences: RIF vs PRR

- NOTE: OMG PRR is for modeling, RIF is for interchange
  - Differentiator: interchange can be runtime
- PRR focused on 1 rule type based on existing use
  - Generic model defined for extensibility
- Data source: defined as model layer by UML (eg UML classes/types)
  - UML can map to MDA PSM level with different data sources eg Java, XML, etc
- PRR Semantics
  - Metamodel (with default XMI specification)
  - Semantic description: textual

RIF Issues vs PRR

- No surprises
  - PRR takes
    - Least common denominator approach for simplicity
    - Superset where easily mapped
      (eg RuleVariables as a ruleset or rule construct, for modeling)
    - Rule actions can invoke external methods for other services etc
- Coverage
  - Limited to 2 main types of rules (Rete inferencing + sequential)
    - No mixing allowed
  - Driven my **current** market pragmatics
- Extensibility
  - Roadmap for unsupported features post v1 – to be based on market requitements / demand
  - Extensibility will be covered by standards body (no metafacility)

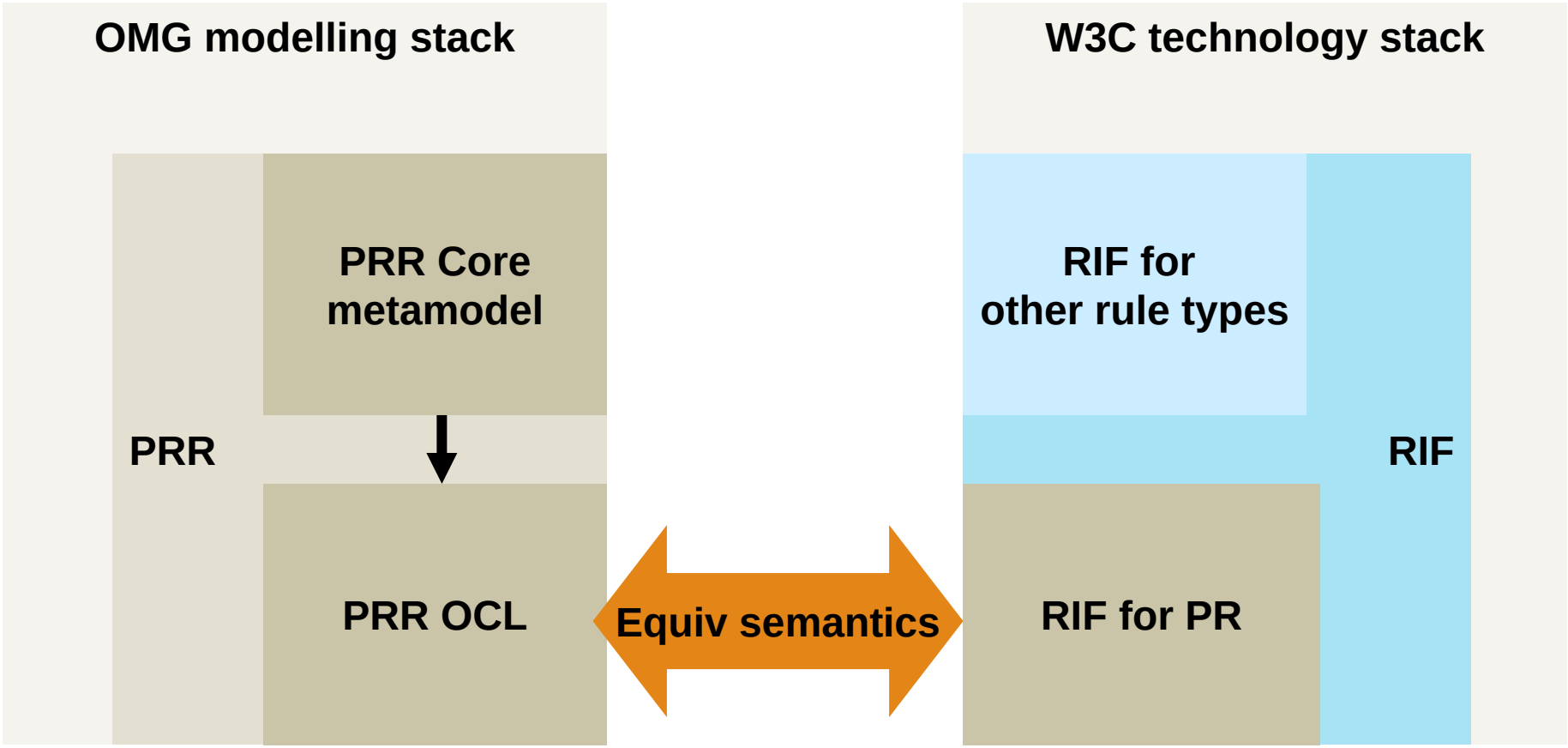
RIF Issues vs PRR (ctd)

- Semantic Web support and/or
  is PRR a candidate Semantic Web Rule Language?
  - PR defined against OWL or RDF?
  - Qus:
    - What rule types make sense for a Semantic Web?
    - Is Semantic Web behaviour-free?
      - PRs usually used to represent behavior
    - Is Semantic Web for business internal use or B2B?
      - Most PR use cases are for these
  - Note: few business requests for Sem Web PR support at this time (but could be a chicken-egg situation)
- Low implementation cost
  - NOT a PRR issue; simplification principle helps here

RIF Issues vs PRR (ctd)

- Support for W3C Specifications
  - PRR is at the MDA PIM level; technology platforms are PSM
    - XPath or RDF could be an expression language for conditions
  - Many PR vendors support XML / write rules against XSD

W3C RIF: potential PRR use



Modelling: design time

Interchange: any time

Summary

- PRR provides a standard metamodel for production rules as used in popular rule engines for business automation
- PRR is much simpler than RIF through scope limitations
- Expected to be completed / go to “finalization” later this year

Thank you

References

- OMG BMI = http://bmi.omg.org/
  includes the Business Rules Working Group, creators of PRR
- OMG PRR details [OMG only] = http://www.omg.org/techprocess/meetings/schedule/Prod._Rule_Representation_RFP.html
- OMG PRR external White Paper = TBA