REWERSE©
reasoning on the web

A Generic Module System for Web Rule Languages: Divide and Rule

Uwe Aßmann, Sacha Berger, François Bry,
Tim Furche, Jakob Henriksson,
Paula-Lavinia Pătrânjan

RuleML 2007

October 2007

Modules

- Software units that
  - Group together parts of programs and knowledge
  - Usually serve a specific purpose
- An essential feature for programming languages
  - Make large scale projects tractable by humans
  - Facilitate the integration of existing applications
  - Offer a flexible solution to application development

… and Rule Languages

- The rule-based programming paradigm
  - High-level means for developing applications of various domains
  - Flexible and adaptive approach towards application development
- Rule languages
  - Have been developed within different communities
    - Database systems, business rules, (Semantic) Web
  - There is still room for further exploiting their potential
    - Reuse and integrate knowledge
    - Specified in different rule languages

Modules for Web Rule Languages

Besides the (general) advantages of modules in programming languages:

- Scoped inference
  - Infer new knowledge within an explicitly given scope
- Data integration
  - Issue also for the W3C RIF WG

… most of (Web) rule languages still lack a module system!

Modules for Web Rule Languages
Our Proposal

- A conceptual framework for modules in rule languages
  - Provides abstract language constructs
  - Abstracts away from a particular data model
  - Preserves the languages' semantics
- Main idea
  - Rewriting modular programs into semantically equivalent non-modular programs
  - Based on so-called *reduction semantics*
- Focus on genericity
  - Applicable to many Web rule languages

Modules for Web Rule Languages Framework

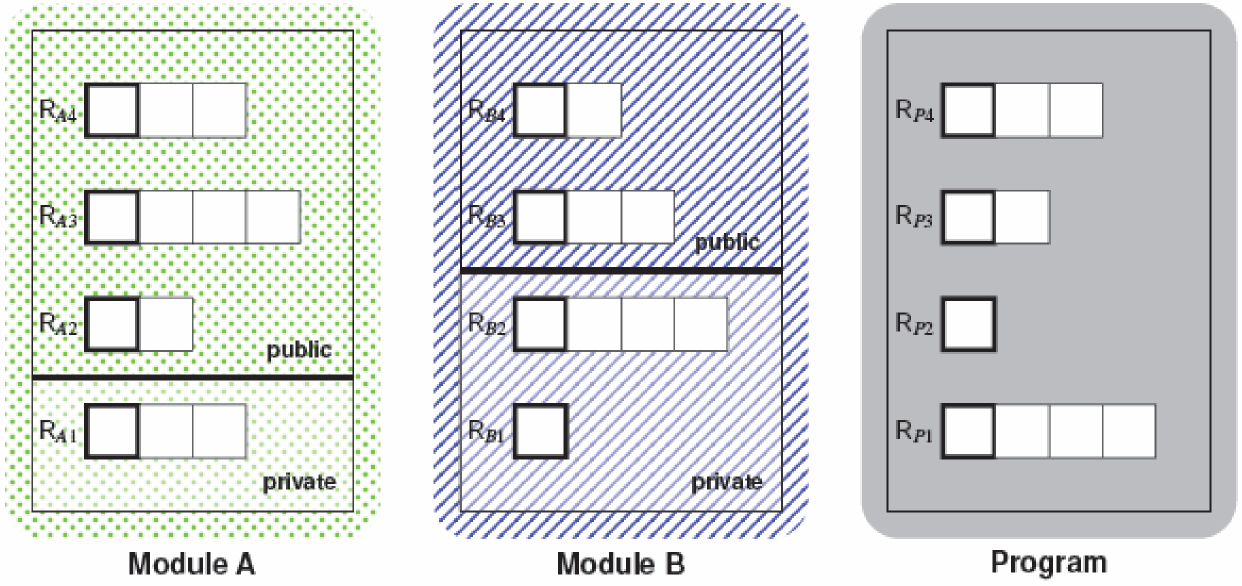
- Languages based on deductive rules
  - Of the form *head <- body*
  - Where head and body are finite sequences of atomic formulas
- Requirement for rule languages
  - Rule chaining (or rule dependency)
  - Dependency relations between body and head parts of a program (or module)
- Example rule languages
  - SWRL, RIF BLD, Datalog, Triple, Prolog
  - but not CSS

Module System Algebra
Module Definition

A module M is a triple (R_{PRIV}, R_{PUB}, $\triangle$)

- R_{PRIV}, R_{PUB} are the private and public, respectively, rules of M
- $\triangle$ is the dependency relation between body parts and heads of rules

Program and two defined modules without imports

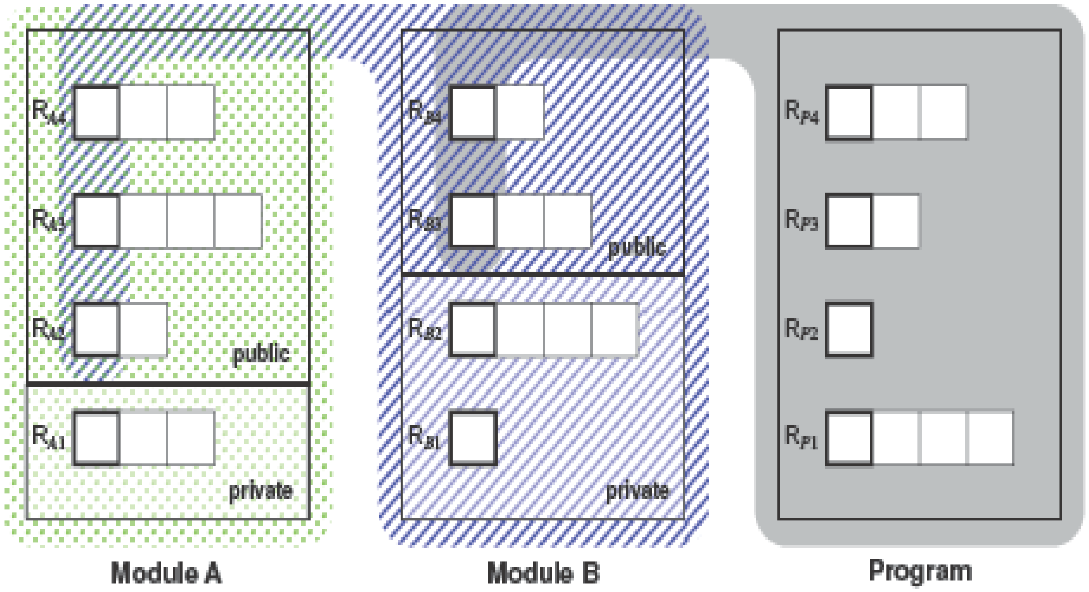


Module System Algebra
Module Composition Operators

- Import, denoted M1 × M2
  - Public rules of M2 are visible in M1
  - … and in all other modules importing M1
- Qualified import
  - Import public, denoted M1 × M2
  - Import private, denoted M1 ⋈ M2
    - M2 not visible in modules importing M1
- Scoped import, denoted M1 $\rtimes_{S}$ M2
  - M2’s rules become visible only to specifically marked body parts of M1
  - Extends the dependency relation of M1
  - Pairs of body parts from M1 and heads of rules from M2
- …

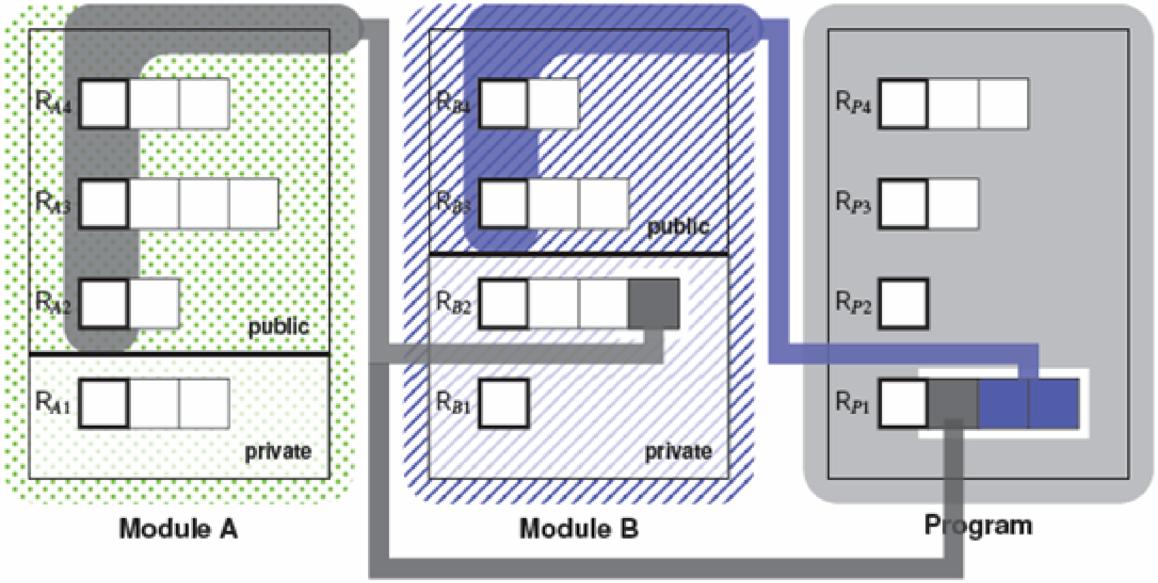
Module System Algebra
Public and Private Import

Private import of A into B and B into the main program



Module System Algebra
Module Composition

Scoped import of (1) module A into body part 3 of rule R_{B2} and into body part 1 of rule R_{P1} and (2) of (the expanded) module B into body part 2 and 3 of rule R_{P1}. into the main program



Module System Algebra
Module Composition

- A single fundamental module composition operator
  - Scoped import as base of our module algebra
  - Public and private import
    - Can be reduced to scoped import in presence of views
    - Details in the paper
- By composing modules a new set of rules is formed
  - A new dependency relation is attached to this set
  - An operation is needed
    - For adjusting a given dependency relation
    - The **slide operation** suffices
    - Formal definitions in the paper!

Concluding Remarks

- We do not
  - Propose a concrete module system for a given rule language!
- Instead, our work
  - Abstracts from similar module systems realizable for different rule languages
  - Proposes a framework for a generic module system
  - That can be implemented e.g.
  - By means of a composition framework such as Reuseware

    http://www.reuseware.org/
  - As done e.g. for the Web rule language Xcerpt

Thank you for your attention!

For more information on our research work:

http://rewerse.net/