# An Ontology- and Resources-Based Approach to Evolution and Reactivity in the Semantic Web

**Wolfgang May**

**José Júlio Alferes** **Ricardo Amador**

Institut für Informatik, Universität Göttingen, Germany

CENTRIA, Universidade Nova de Lisboa, Portugal

Supported by the EU Network of Excellence


ODBASE 2005, Agia Napa, Cyprus, Nov. 3., 2005

# Motivation and Goals

(Semantic) Web:

- XML: bridge the heterogeneity of data models and languages
- RDF, OWL provide a computer-understandable semantics

... same goals for describing behavior:

- description of behavior *in* the Semantic Web
- semantic description *of* behavior

# Cooperative Evolution of the Semantic Web

There are not only *queries*, but there are *activities* going on in the Semantic Web:

- Semantic Web as a base for processes
  - Business processes, designed and implemented in participating nodes: banking, . . .
  - Predefined cooperation between nodes: travel agencies, . . .
  - Ad-hoc rules designed by users
- The less standardized the processes (e.g. human travel organization), the higher the requirements on the Web assistance and flexibility

# Goal

- Ontology for describing behavior on the Semantic Web
  - correlate actions, state, and events
  - high-level actions are translated to lower levels
  - high-level events are derived from
    - low-level events
    - high-level actions
- combine application-dependent semantics with generic concepts of behavior
- [Markup]
- Operational Semantics

$\Rightarrow$ use Event-Condition-Action Rules as a well-known paradigm.

# ECA Rules

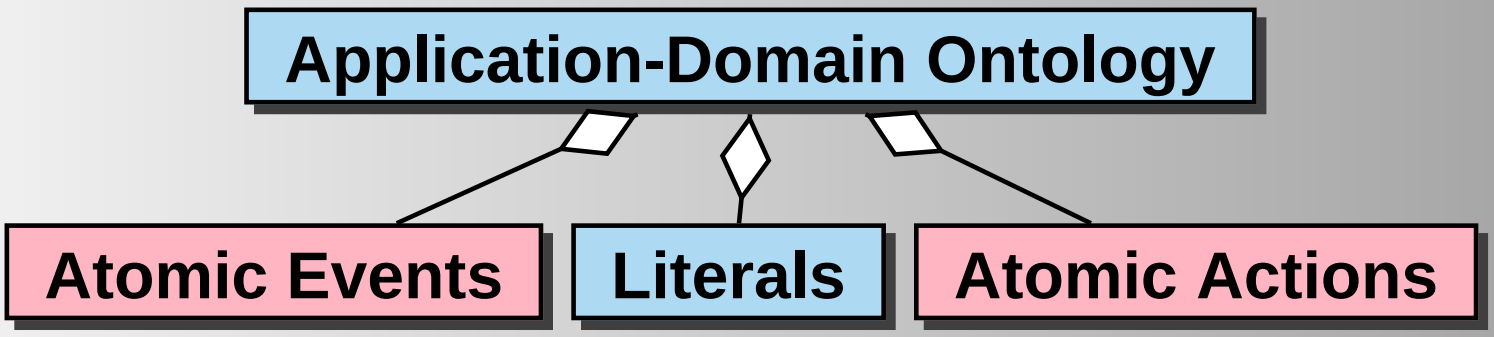
"On Event check Condition and then do Action"

- Active Databases
- modular, declarative specification
- sublanguages for specifying *Events*, *Conditions*, *Actions*
- simple kind (database level): triggers

# Events and Actions in the Semantic Web

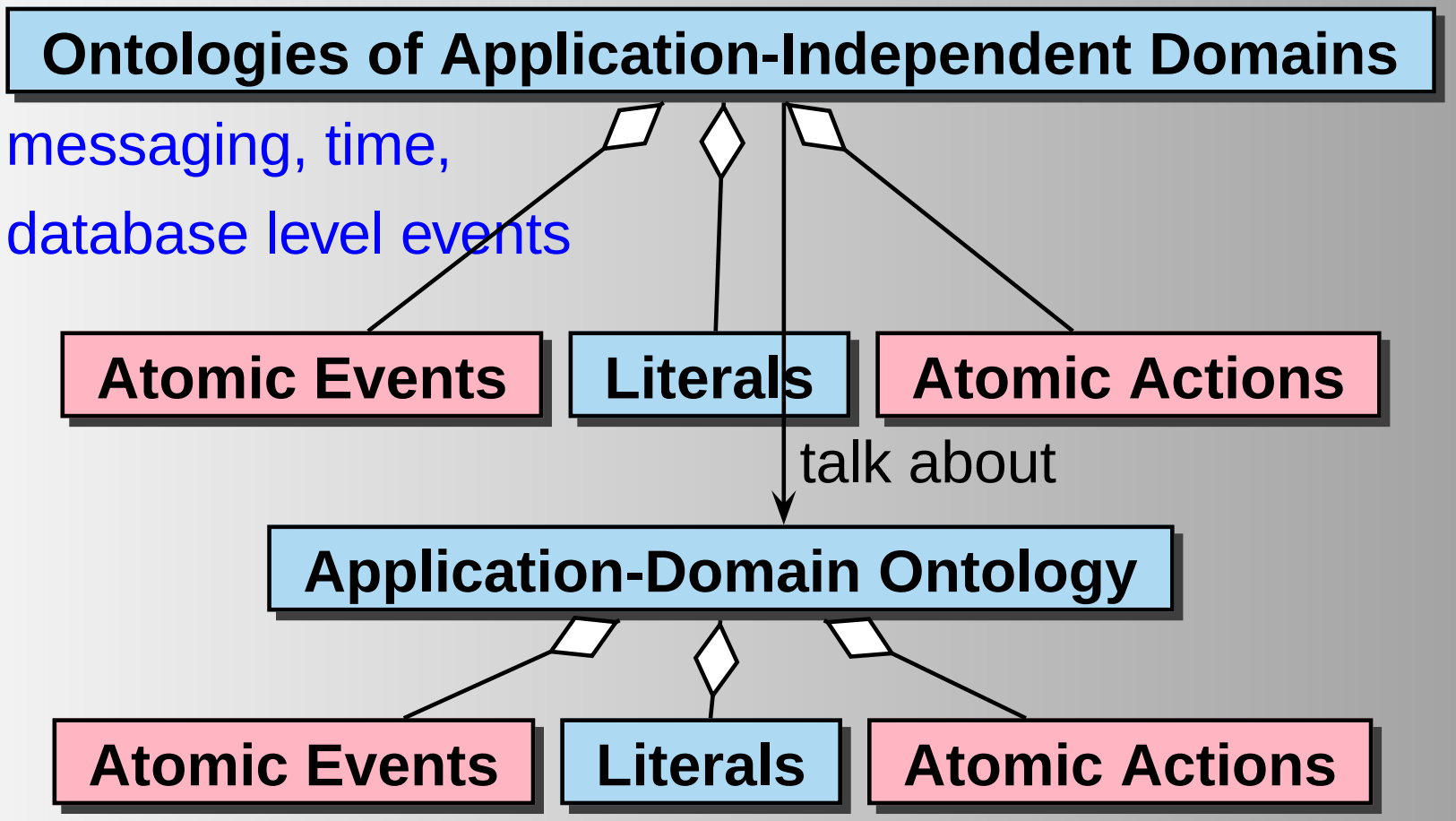
- applications do not only have an ontology that describes static notions
  - cities, airlines, flights, hotels, etc., relations between them ...
- but also an ontology of events and actions
  - cancelling a flight, cancelling a (hotel, flight) booking,
- allows for correlating actions, events, and derivation of facts
  - intensional/derived events are described in terms of actual events
    e.g., “economy class of flight X is now 50% booked”
    (e.g., as “if *(composite) event* and *condition* then (raise) *derived event*”)

# Ontologies including Dynamic Aspects



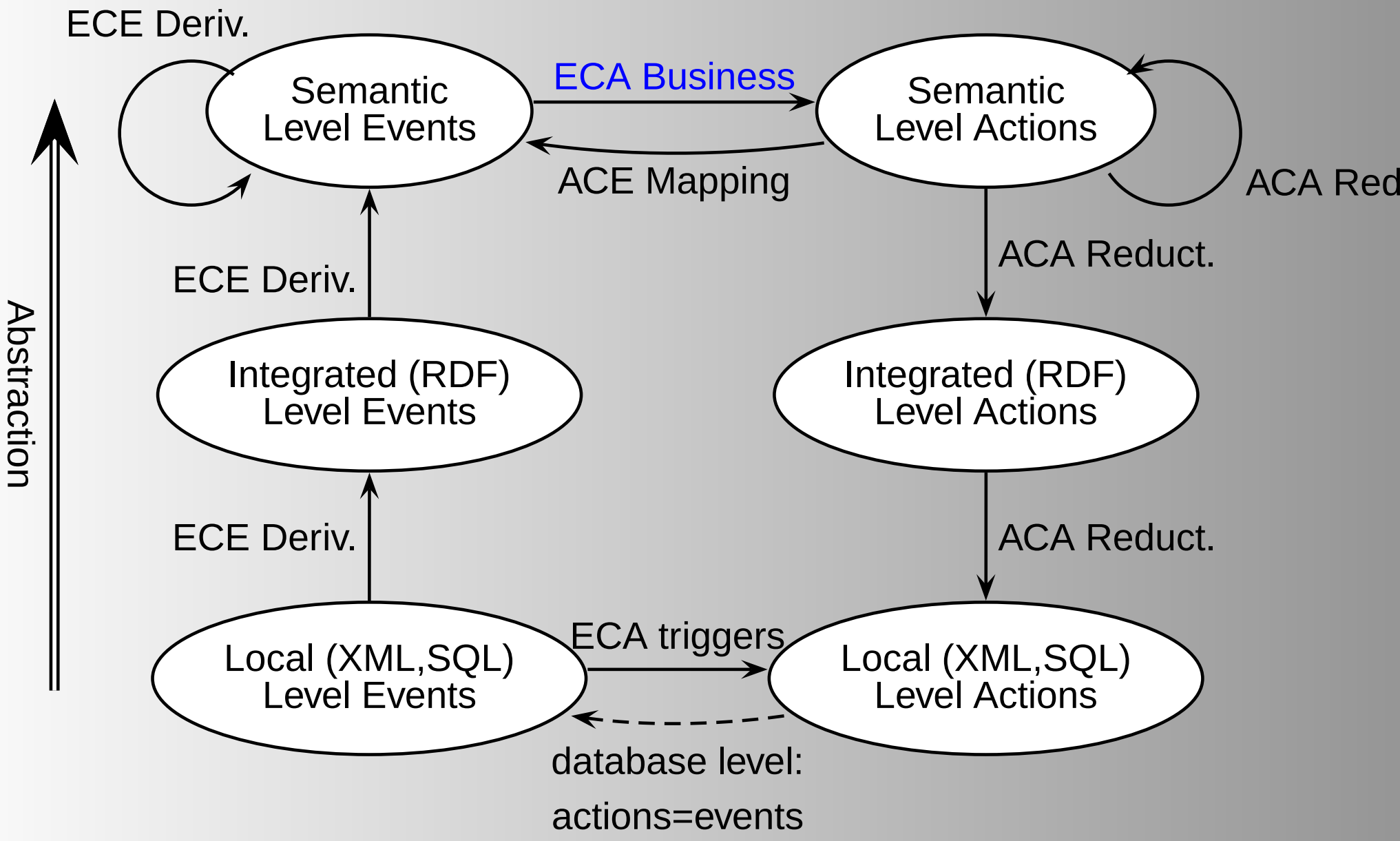

- correlate actions, state, and events

# Ontologies including Dynamic Aspects




- correlate actions, state, and events

# Abstraction Levels and Types of Rules



# Sample Rule on the XML Level

- reacts on an event in the XML database
- here: maps it to an event on the RDF level
- actually an *ECE derivation rule*

```
ON INSERT OF department/professor
let $prof:= :NEW/@rdf-uri,
      $dept:= :NEW/parent::department/@rdf-uri
RAISE RDF_EVENT(INSERT OF has_professor OF department)
   with $subject:= $dept, $property:=has_professor, $object:=$prof;
```

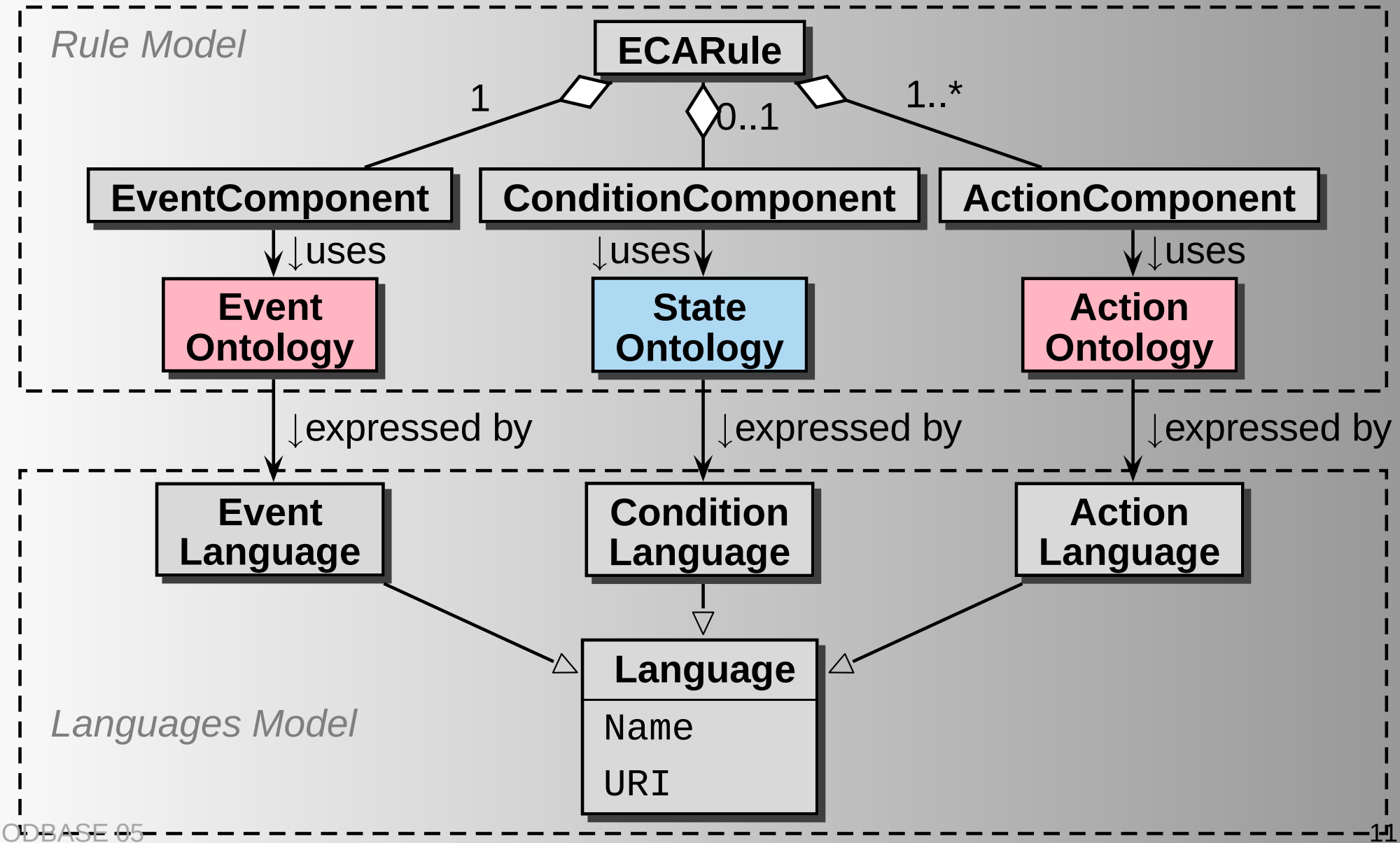
# Sample Rule on the RDF Level

- reacts on an event on the RDF view level
- here: maps it to an event on the OWL level
- again an *ECE derivation rule*

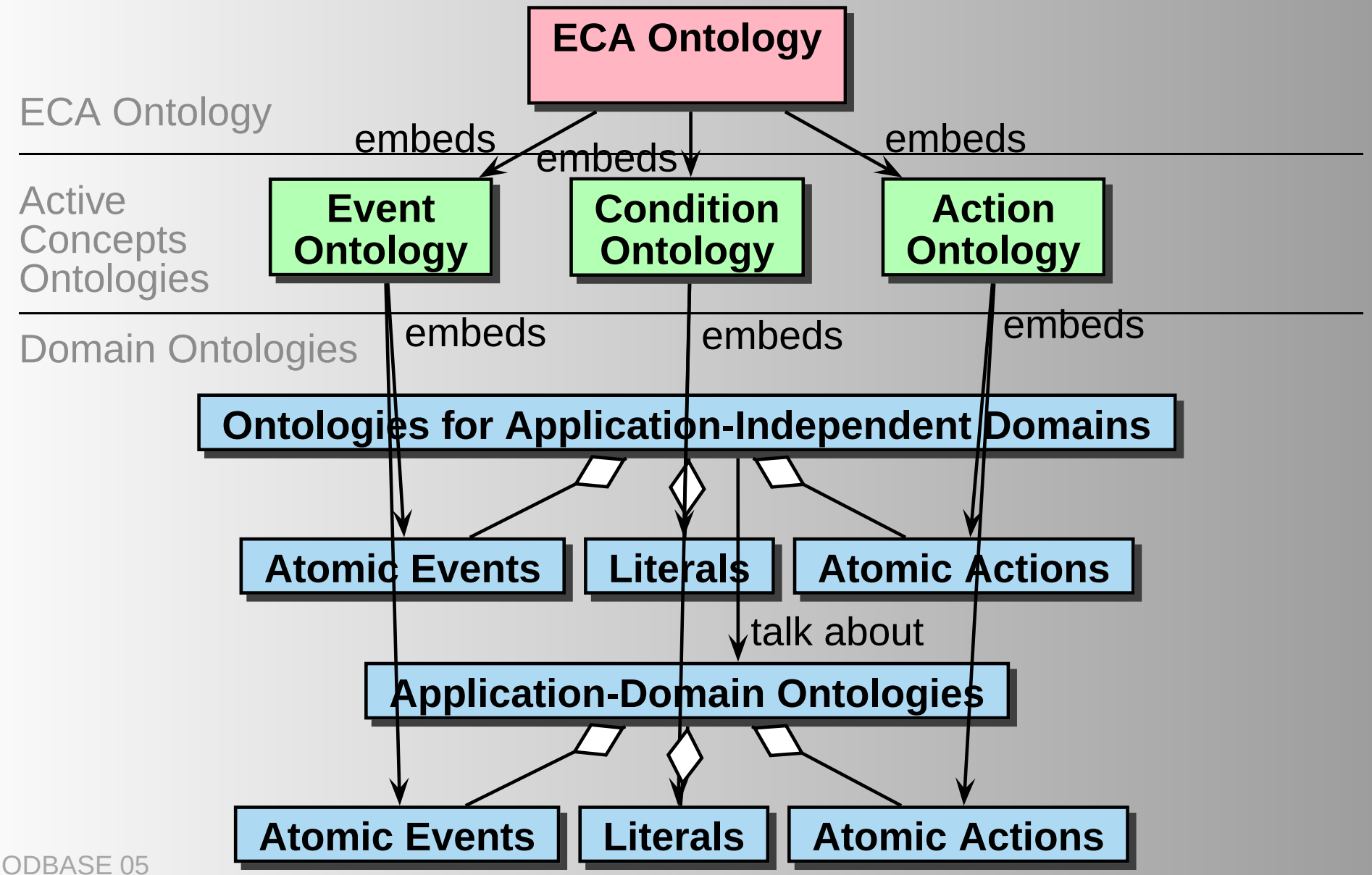
```
ON INSERT OF has_professor OF department
  % (comes with parameters $subject=dept,
  %    $property:=has_professor and $object=prof )
  % $university is a constant defined in the (local) database
RAISE EVENT
  (professor_hired($object, $subject, $university))
```

... which is then an event of the domain ontology.

# Modular ECA Concept: Rule Ontology



# Combination of Ontologies



# Active Concepts Ontologies

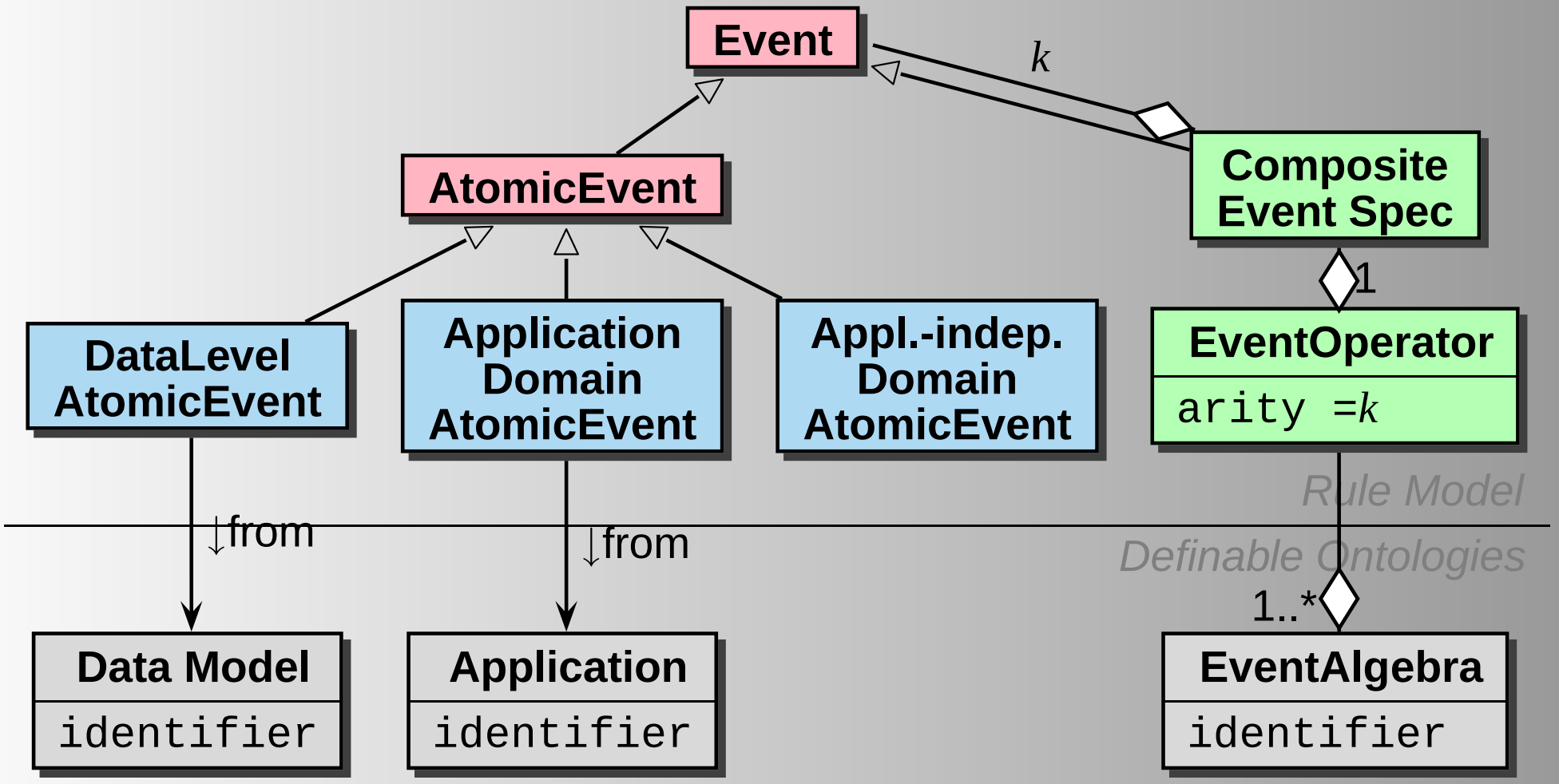
- Domains specify atomic events, actions and static concepts

## Composite [Algebraic] Active Concepts

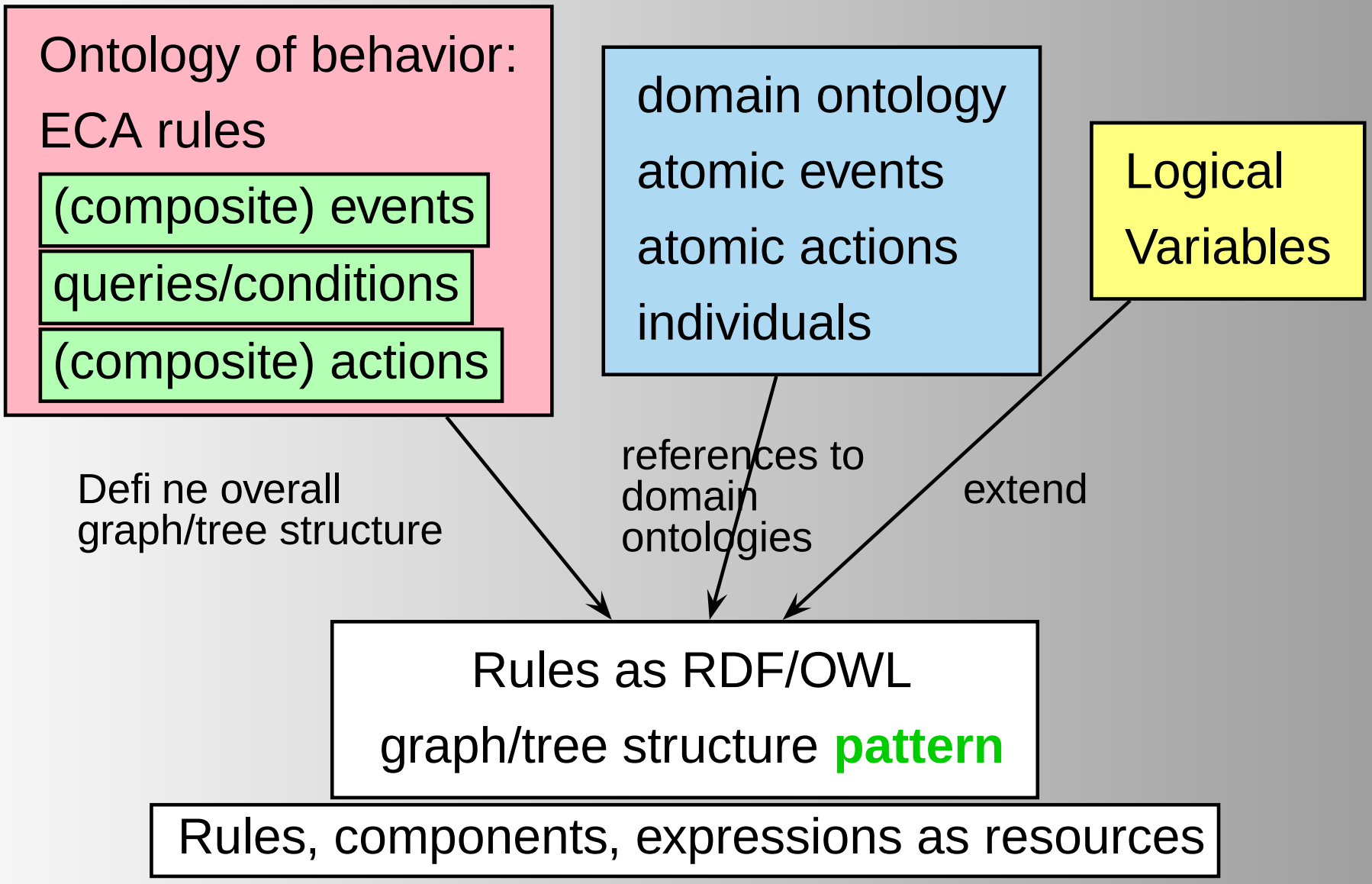
- Event algebras: composite events
  - (when) $E_1$ and some time afterwards $E_2$ (then do $A$)
  - (when) $E_1$ happened and then $E_2$, but not $E_3$ after at least 10 minutes (then do $A$)
  - well-investigated in Active Databases (e.g. SNOOP).
- Process algebras (e.g. CCS)

$\Rightarrow$ See concepts defined by these *formal methods* as defining ontologies

# Events Subontology



# ECA Rules in RDF/OWL



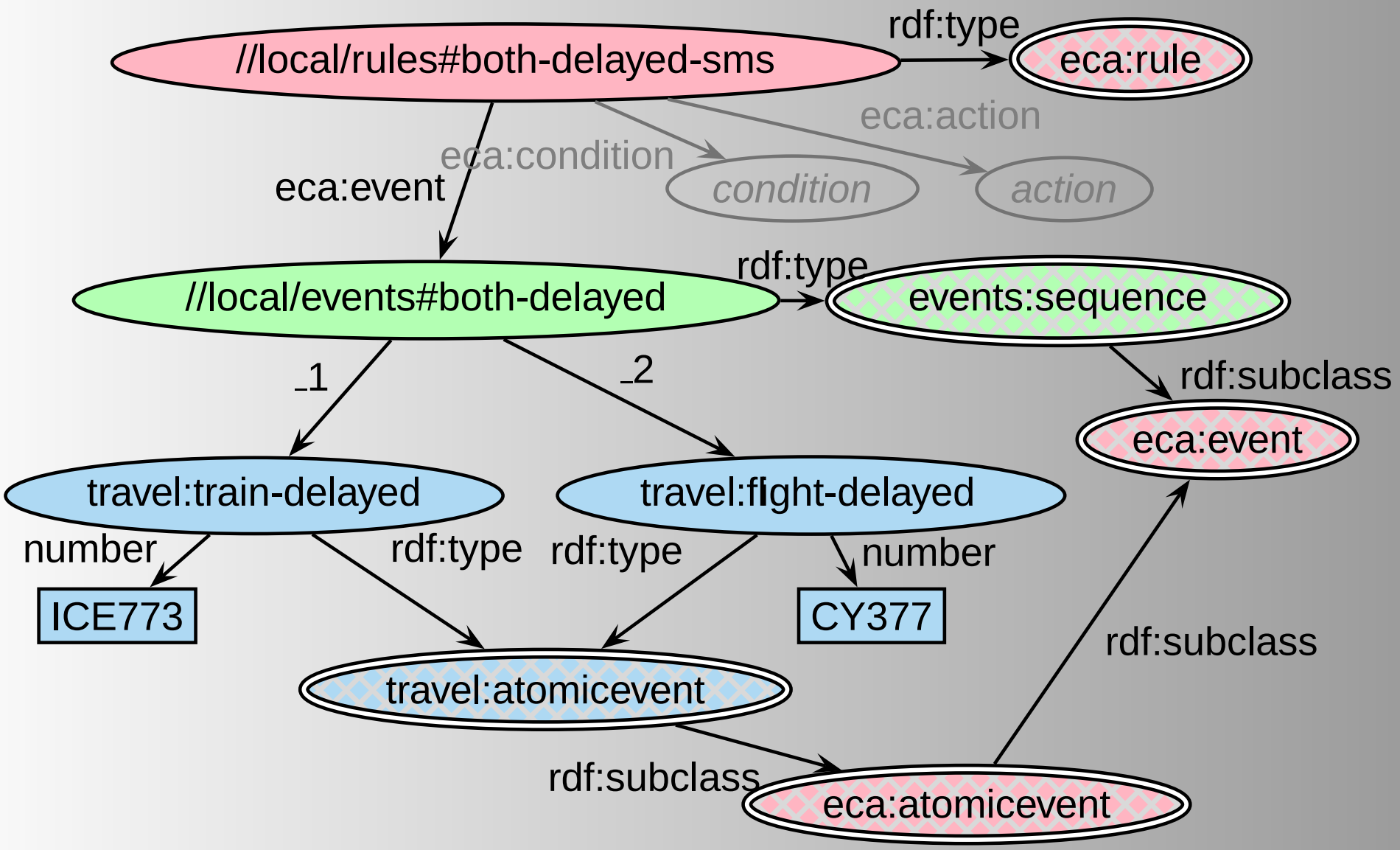
# Example: Travel Domain

Portal functionality:

- static: access to train/flight schedules, hotels etc.
- dynamic: communication of events and announcements like delayed or cancelled flights
- users: can register rules for their personal needs

1. “if my train is delayed, send me an SMS”.
   Simple, often not necessary – no danger.
2. “if my flight is delayed ... don’t send me a mail”
   (same as above).
3. But: “if my train towards the airport is delayed, and the flight is also delayed, then send me an SMS (about the latter)”
   ⇒ composite sequential event.

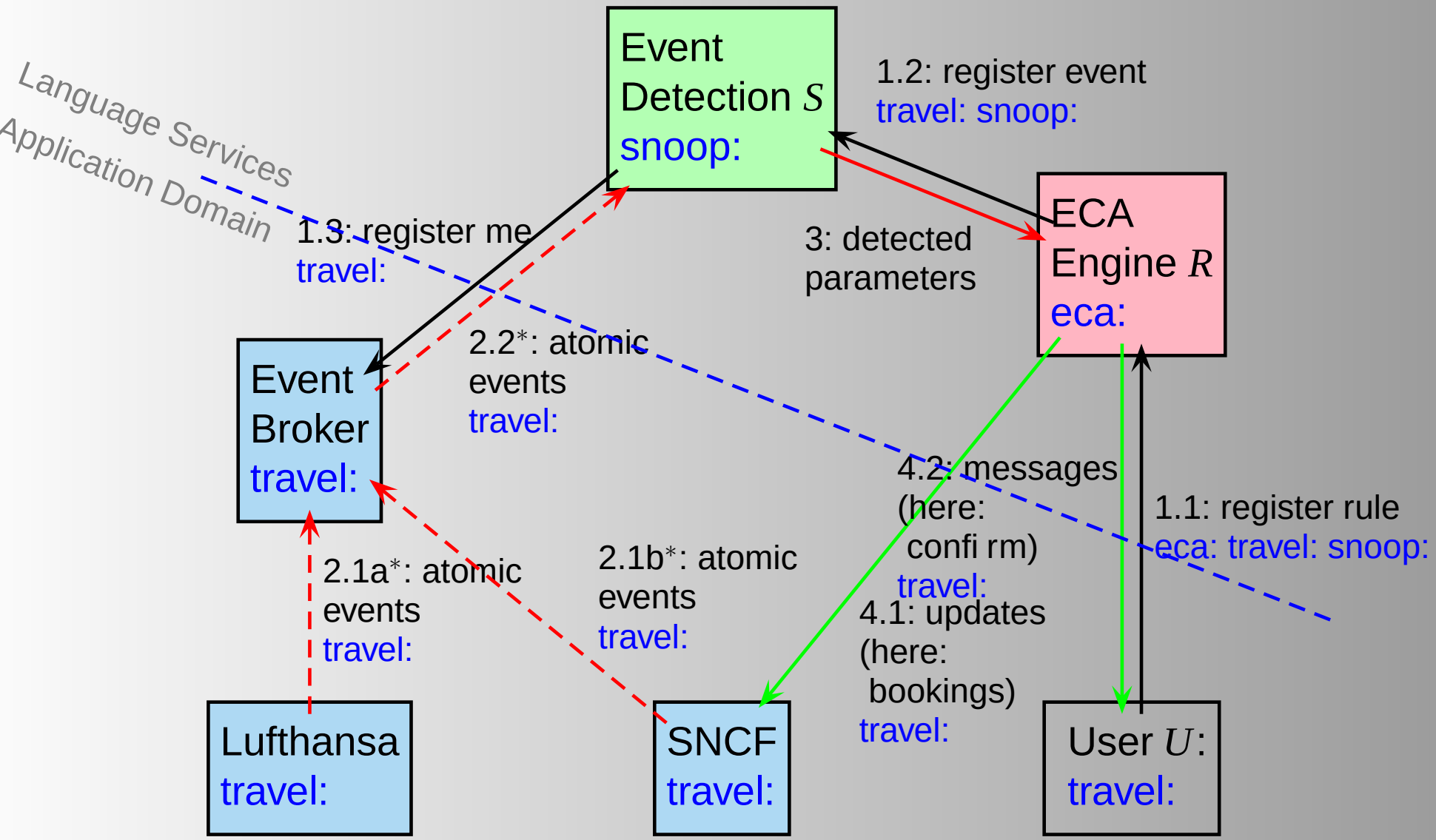
# Example


//local/rules#both-delayed-sms
rdf:type
eca:rule
eca:action
eca:condition
eca:event
condition
action
//local/events#both-delayed
rdf:type
events:sequence
_1
_2
rdf:subclass
eca:event
travel:train-delayed
travel:flight-delayed
number
rdf:type
rdf:type
number
ICE773
CY377
rdf:subclass
travel:atomicevent
rdf:subclass
eca:atomicevent

# Ontologies, Languages and Resources

- Rule components, subexpressions etc. are resources
- associated with languages corresponding to the ontologies (event languages, action languages, domain languages)
- each language is a resource, identified by a URI.
- DTD/XML Schema/RDF description of the language
- Algebraic languages:
  processing engine
- Domain Languages:
  Event Broker Services (subscribe) and
  processors for actions

# Architecture



# Publications & Details

- REWERSE I5-D4: “Models and Languages for Evolution and Reactivity”: Everything + examples
- ODBASE05: ontology of rules, rule components and languages, and the service-oriented architecture
- RuleML05: languages and their markup, communication and rule execution model