

# r<sup>3</sup>: Towards a Foundational Ontology for Reactive Rules

José Júlio Alferes and Ricardo Amador

Centro de Inteligência Artificial - CENTRIA, Univesidade Nova de Lisboa

{jja, ra}@di.fct.unl.pt

## Introduction

The **Semantic Web** is based on peer-to-peer communication between autonomous, and **autonomously developing, nodes** and it should support (not only querying, but also) propagation of knowledge and changes in a semantic way. Such **evolution** depends on the cooperation of nodes, and the heterogeneity of concepts for expressing **behavior** requires an appropriate handling on the semantic level. It is really unlikely that there will be a unique language for describing behavior throughout the entire Web.

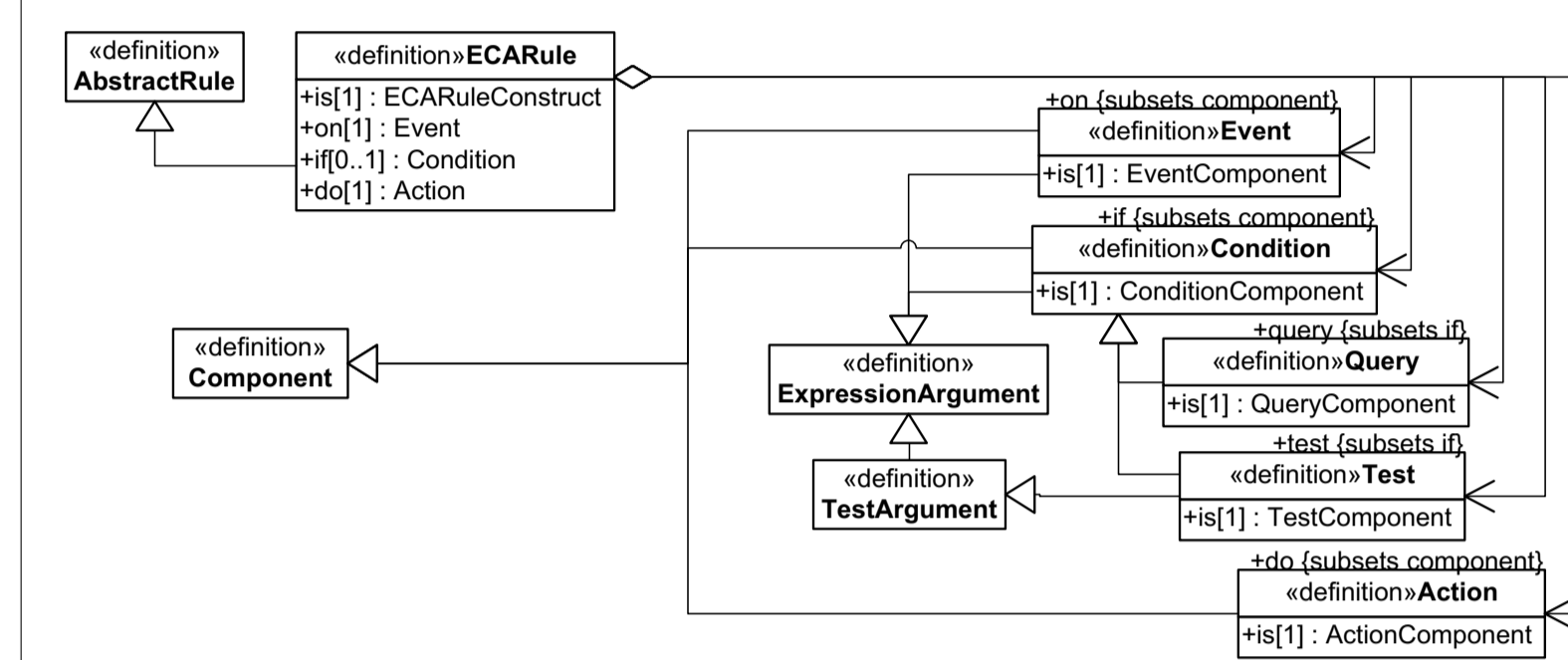
*Heterogeneous Reactivity.* A suitable common model (described in [4, 5, 2]) for dealing with such heterogeneity is provided by **reactivity** together with its formalization as *Event-Condition-Action (ECA) rules*, bearing: a clean separation between the *content* of a rule and the *generic semantics* of the rules themselves; an inherent loosely coupled nature, allowing the declarative combination of functionality from different Web sites (providing events and services).

*Resourceful Reactive Rules.* Much research effort is currently being targeted upon rule interchange formats, and upon defining rules for/about ontologies. In what concerns the former, most of proposals are XML-markup based and rely on specific abstract syntax, ignoring the fact that rules do not only operate on the Semantic Web, but are themselves also part of it. **Rules must be first class citizens of the Semantic Web**, especially if one wants to express and reason about evolution of behavior (towards **dynamic behavior** and **behavior policies**).

*The r<sup>3</sup> Ontology.* This calls for a **foundational ontology** for (ECA) rules that, according to the heterogeneity requirement previously identified, must allow also the description of different languages to be used and composed at the rule component level. The **OWL-DL r<sup>3</sup> ontology** [2] (here presented using UML diagrams) is an intermediate step towards that goal reflecting work in progress in the r<sup>3</sup> project [9].

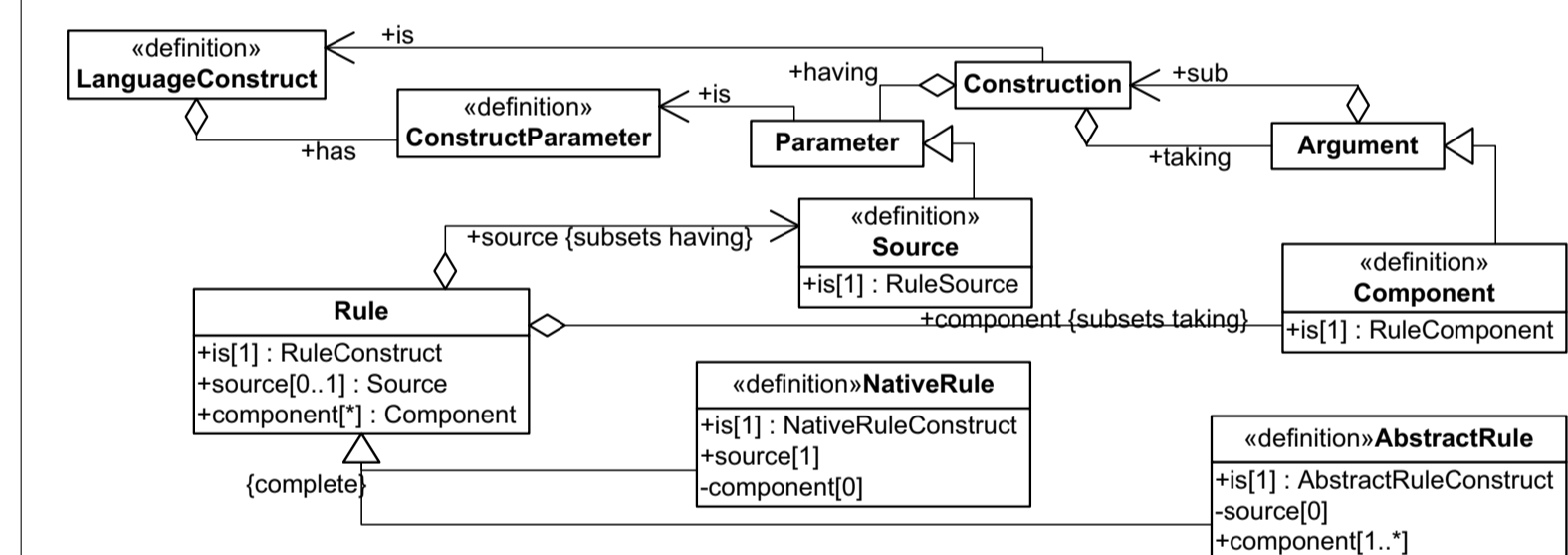
## Describing Reactive Rules

The **component structure** of an ECA rule (which is shared by other kinds of reactive rules) is core to the r<sup>3</sup> ontology.



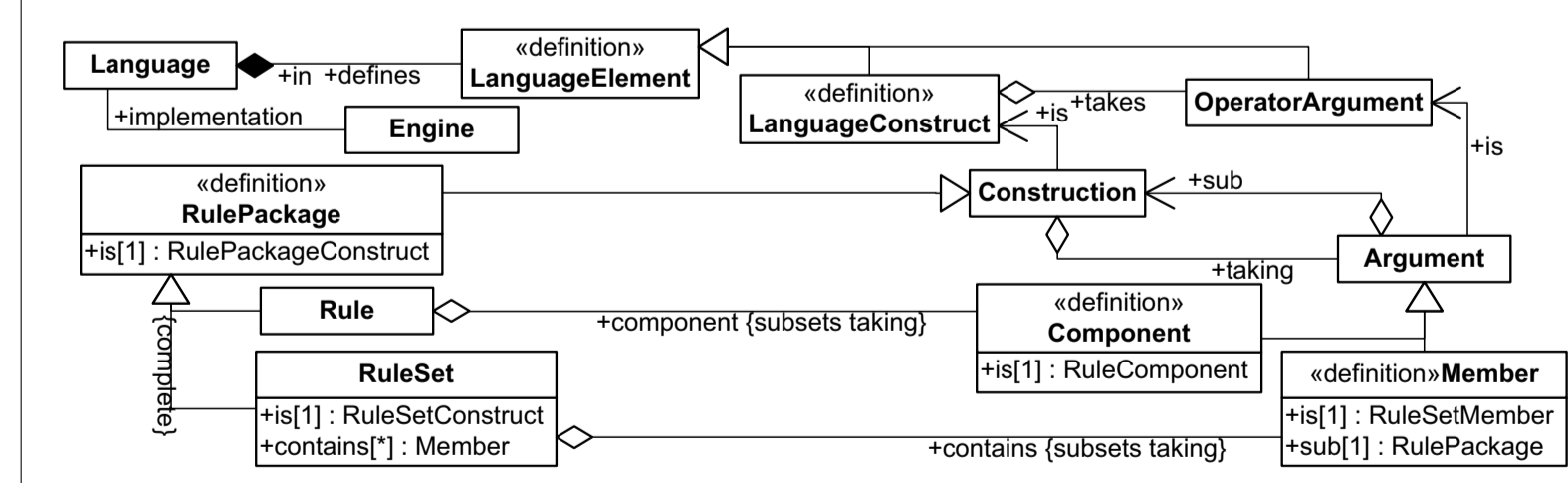
```
_:quotation-request-rule :is eca:rule;
:on [:is eca:event; :sub _:request-for-quotation];
:if [:is eca:condition; :sub _:available-quotation];
:do [:is eca:action; :sub _:send-quotation].
```

Such a modular structure may be generalized to a **recursive term-structure** of constructions (using **parametric operators**).



```
_:available-quotation :is eca:collect-and-test;
:taking [:is eca:collect; :sub _:quotation-data];
:taking [:is eca:test; :sub _:check-price]].
_:quotation-data :is fol:and;
:taking [:is fol:some; :sub _:flight-info];
:taking [:is fol:more; :sub [:is fol:and;
:taking [:is fol:some; :sub _:get-client];
:taking [:is fol:more; :sub _:get-available-cars]]].
```

This term-structure is applied not only to rules but also to rule components and **rule sets**.

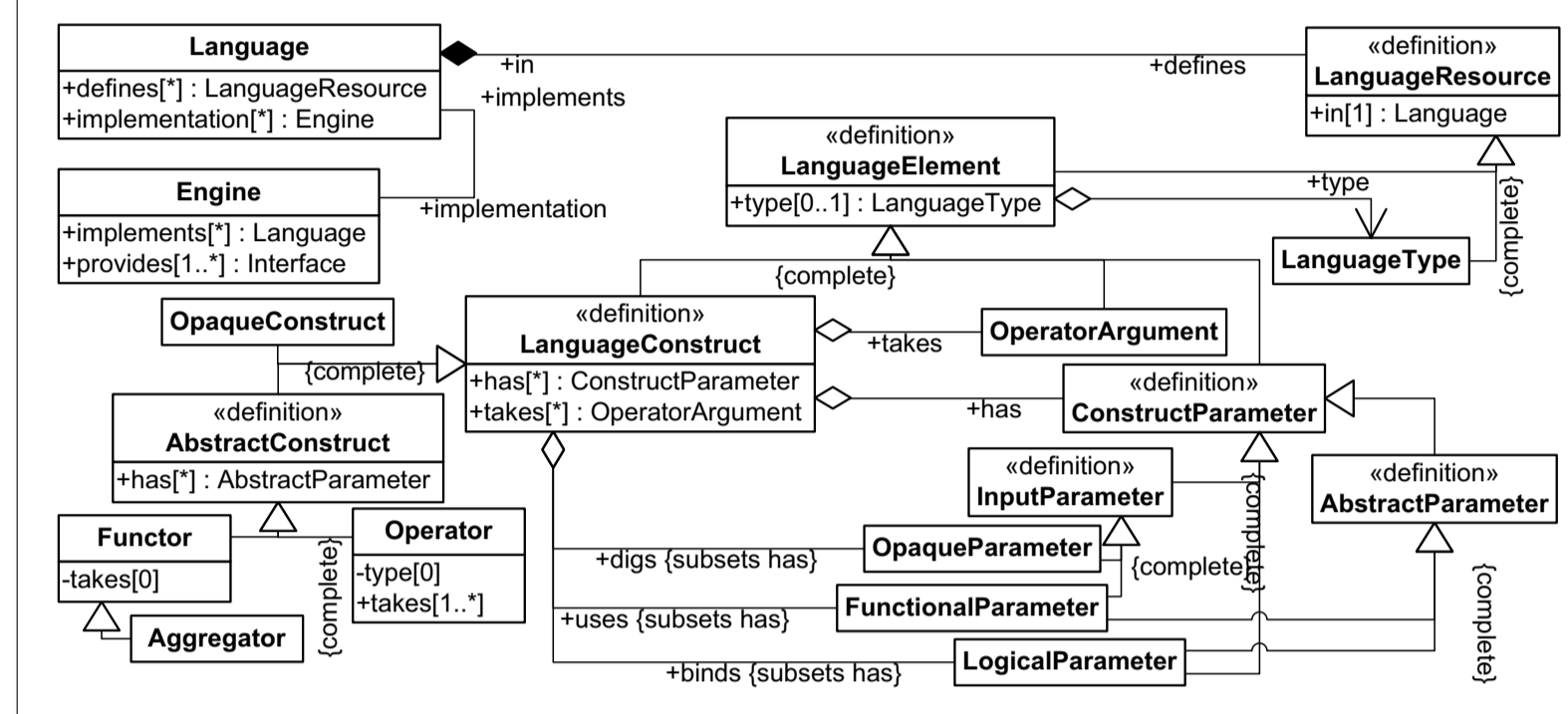


The building elements of such structure are described at a **meta-level** as language elements.

The constructs of a particular language are implemented by specific **engines** available on the Semantic Web.

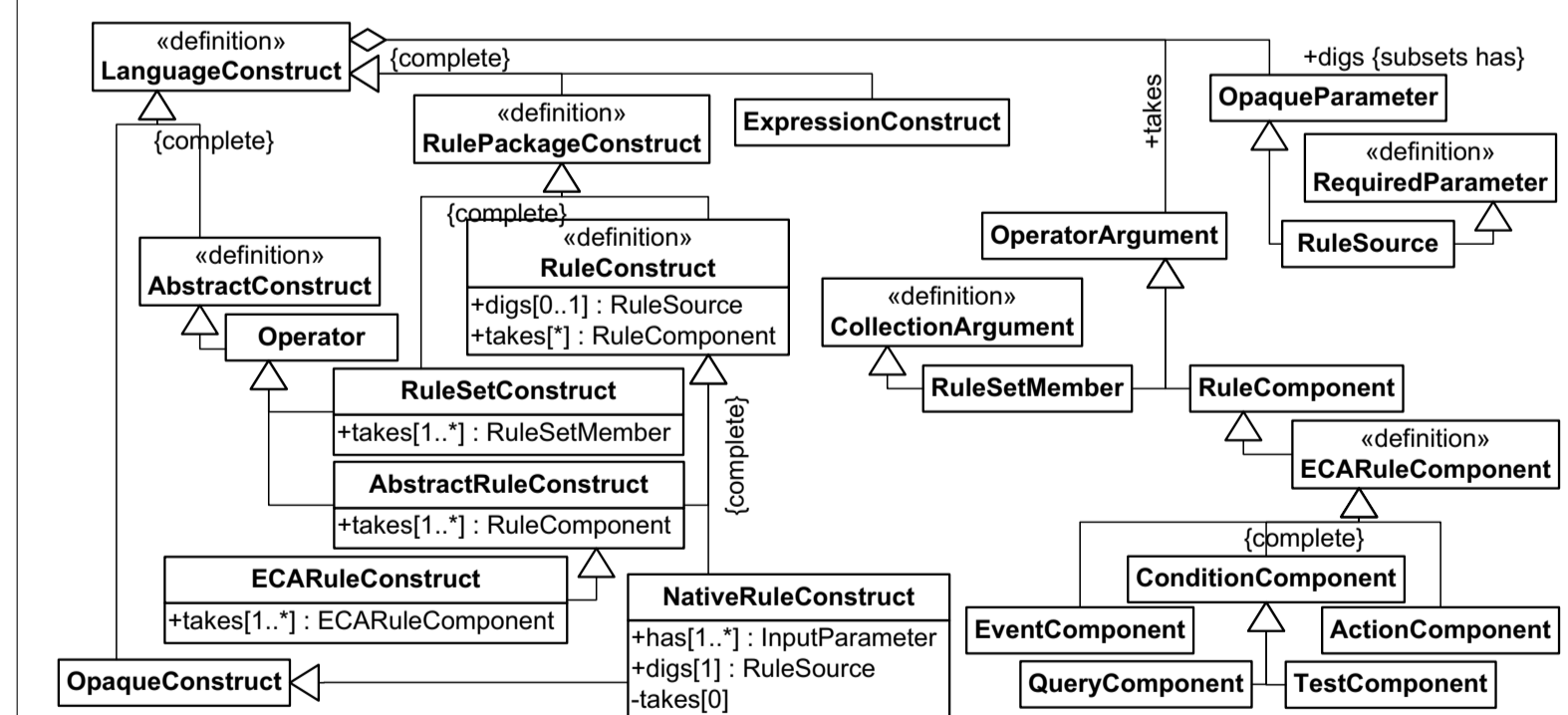
## Describing Language Elements

Language constructs include **atomic functors** and **compositional operators**.



```
event:sequence a :Operator; :in event:algebra;
:takes event:first, event:next.
travel:booking-place a :Functor; :in travel:domain;
:binds travel:client, travel:flightnr, travel:seat.
travel:flight-info a :Functor; :in travel:domain;
:uses travel:flightnr;
:binds travel:date, travel:origin, travel:destination.
rental:request-quotation-for-flight a :Functor;
:in rental:application
:uses rental:client, rental:flightnr.
rental:get-client a :Functor; :in rental:application
:uses rental:client;
:binds rental:client-name,
rental:favorite-class, rental:max-price.
rental:get-available-cars a :Functor;
:in rental:application
:uses rental:office, rental:date;
:binds rental:car, rental:car-class, rental:price.
mail:send a :Functor; :in mail:library;
:uses mail:from, mail:to, mail:subject, mail:body.
text:join a :Aggregator; :in text:library;
:uses text:template, text:separator.
text:replace a :Aggregator; :in text:library;
:uses text:template.
fol:and a :Operator; :in fol:algebra;
:takes fol:some, fol:more.
```

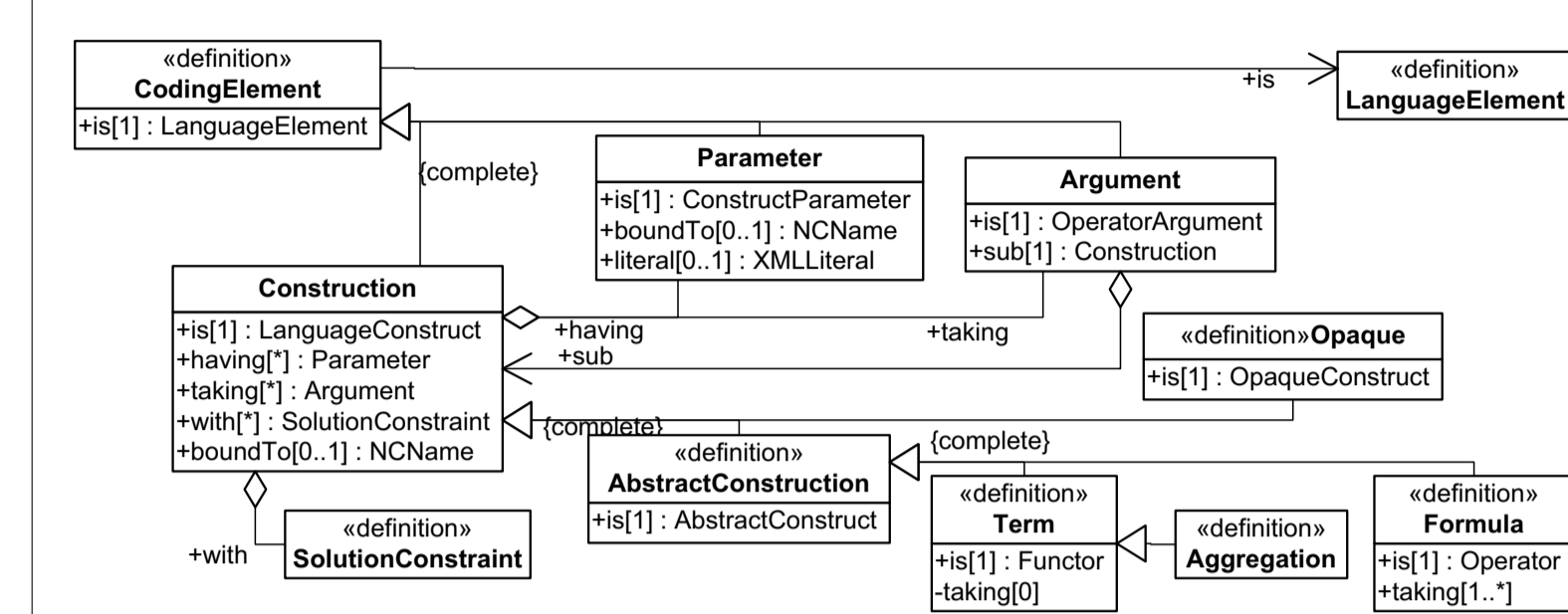
**Abstract parameters may be functional (given) or logical (bound).**



```
eca:rule a :ECARuleConstruct; :in eca:m1;
:takes eca:event, eca:condition, eca:action.
eca:collect-and-test a :Operator; :in eca:m1;
:takes eca:collect, eca:test.
eca:native a :NativeRuleConstruct; :in eca:m1;
:uses eca:lang; :digs eca:source.
eca:opaque a :ExpressionConstruct; :in eca:m1;
:uses eca:lang; :digs eca:literal.
eca:m1 :defines
eca:event, eca:condition, eca:action,
eca:collect, eca:test.
```

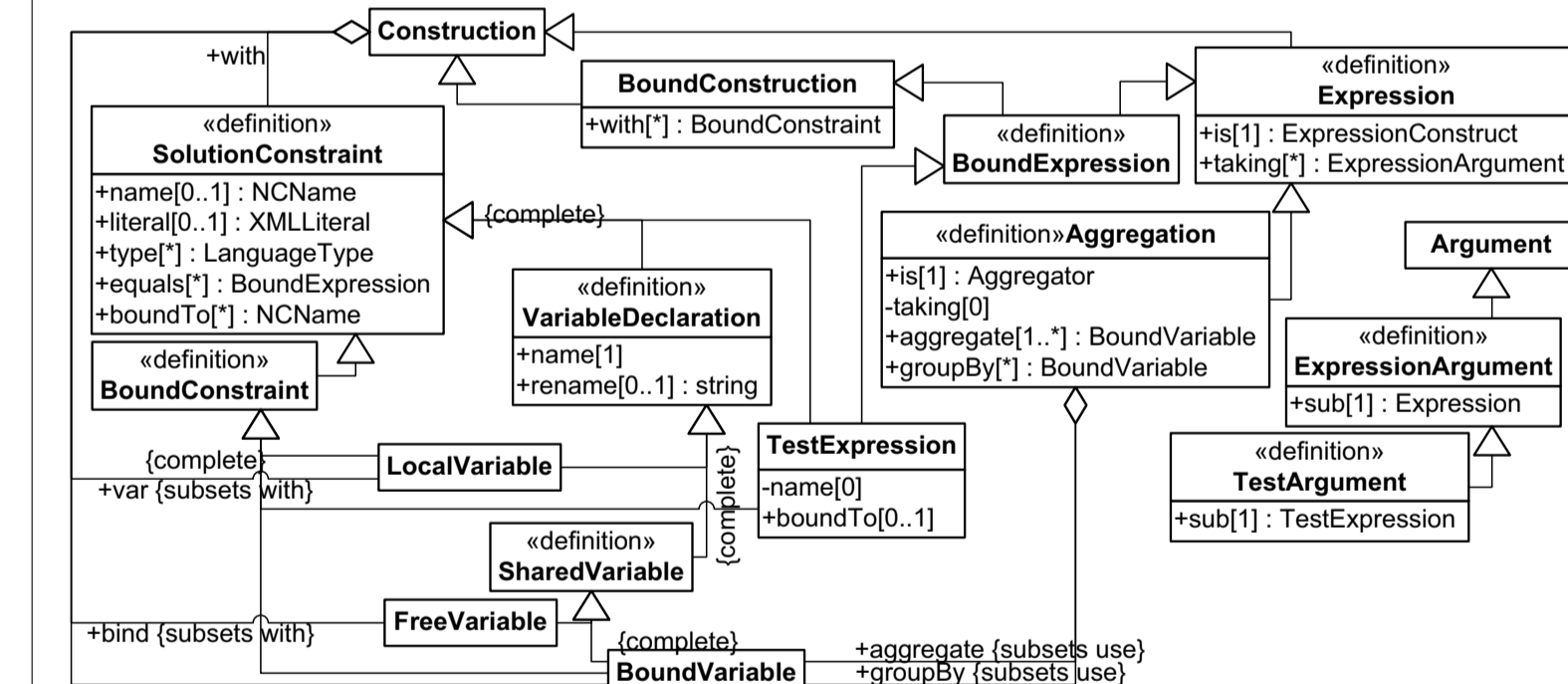
Opaque constructs allow ad-hoc textual languages (e.g. **database triggers**).

## Describing Rule Components



```
_:request-quotation :is event:sequence;
:taking [:is event:first; :sub {
:is travel:booking-place;
:having [:is travel:client; :boundTo "Mail"];
:having [:is travel:flightnr; :boundTo "Flight"]}];
:taking [:is event:next; :sub {
:is rental:request-quotation-for-flight;
:having [:is rental:client; :boundTo "Mail"];
:having [:is rental:flightnr; :boundTo "Flight"]}].
_:get-client :is rental:get-client;
:having [:is rental:client; :boundTo "Mail"];
:having [:is rental:client-name; :boundTo "Client"];
:having [:is rental:favorite-class; :boundTo "Class"];
:having [:is rental:max-price; :boundTo "Max-Price"].
_:get-available-cars :is rental:get-available-cars;
:having [:is rental:office; :boundTo "To"];
:having [:is rental:date; :boundTo "Date"];
:having [:is rental:car; :boundTo "Car"];
:having [:is rental:car-class; :boundTo "Class"];
:having [:is rental:price; :boundTo "Price"].
_:flight-info :is travel:flight-info;
:having [:is travel:flightnr; :boundTo "Flight"];
:having [:is travel:date; :boundTo "Date"];
:having [:is travel:origin; :boundTo "From"];
:having [:is travel:destination; :boundTo "To"].
```

Constructions may also include **variable declarations** and related constraints.



```
_:check-price :is eca:opaque;
:literal "true";
:use [:name "Price"];
:use [:name "Max-Price"; :rename "MaxPrice"];
:having [:is eca:lang;
:literal "http://www.w3.org/XMLSchema"];
:having [:is eca:literal;
:literal "$Price <= $MaxPrice"].
_:send-quotation :is mail:send;
:var [:name "Text"; :equals :quotation-message];
:having [:is mail:from; :boundTo "Rental-Mail"];
:having [:is mail:to; :boundTo "Mail"];
:having [:is mail:subject;
:literal "Car Rental Quotation"];
:having [:is mail:body; :boundTo "Text"].
_:quotation-message :is text:replace;
:use [:name "Quotation"];
:var [:name "Priced-Cars";
:equals [:is text:join;
:aggregate [:name "Car"], [:name "Price"];
:having [:is text:template;
:literal "|Car|=|Price|"];
:having [:is text:separator; :literal ", "];]];
:aggregate
[:name "Client"], [:name "Priced-Cars"],
[:name "Flight"], [:name "Date"], [:name "To"];
:having [:is text:template; :boundTo "Quotation"].
```

## Present State

A **working prototype** [1] of the r<sup>3</sup> framework is available, for a previous version of this ontology, including a supporting library used to implement several component languages. A **new version** of the r<sup>3</sup> ontology is currently being defined allowing the description of derivation rules for higher-level events and actions. This new version will constitute the basis for a full **re-implementation** of the prototype.

*Related Work.* To the present, there are only two ontology proposals for describing rules: **WRL** [3] and **SBVR** [7]. The latter is not targeted to the Semantic Web but does address language heterogeneity, and does not exclude reactive rules (waiting on [6]). It is also worth mentioning that the **MARS** project [8] (with which r<sup>3</sup> shares its ideas) is pursuing a complementary approach to these issues at the OWL-Full level [2].

## References

- [1] J. J. Alferes, R. Amador, E. Behrends, F. Bry, M. Eckert, T. Franco, O. Fritzen, H. Grallert, T. Knabke, L. Krippahl, W. May, P.-L. Pătrănjian, F. Schenk, and D. Schubert. Completion of the prototype scenario. REVERSE report I5-D7, CENTRIA, UNL, 2007.
- [2] J. J. Alferes, R. Amador, E. Behrends, O. Fritzen, T. Knabke, W. May, F. Schenk, and D. Schubert. Reactive rule ontology: RDF/OWL level. REVERSE report I5-D6, CENTRIA, UNL, 2007.
- [3] J. Angele, H. Boley, J. de Bruijn, D. Fensel, P. Hitzler, M. Kifer, R. Krummenacher, H. Lausen, A. Polleres, and R. Studer. Web Rule Language (WRL). <http://www.w3.org/2005/09/09/2005-09-09/>.
- [4] W. May, J. J. Alferes, and R. Amador. Active rules in the Semantic Web: Dealing with language heterogeneity. In *International Conference on Rules and Rule Markup Languages for the Semantic Web (RuleML)*, volume 3791 of LNCS, pages 30–44. Springer, 2005.
- [5] W. May, J. J. Alferes, and R. Amador. An ontology- and resources-based approach to evolution and reactivity in the Semantic Web. In *Ontologies, DataBases, and Applications of Semantics (ODBASE)*, volume 3761 of LNCS, pages 1553–1570. Springer, 2005.
- [6] Object Management Group. *Production Rule Representation (PRR)*. OMG, 2003. <http://www.omg.org/cgi-bin/doc?br/2003-9-3>.
- [7] Object Management Group. *Semantics of Business Vocabulary and Business Rules (SBVR)*. OMG, 2006. <http://www.omg.org/cgi-bin/doc?dtc/2006-03-02>.
- [8] MARS: Modular Active Rules for the Semantic Web. <http://www.dbis.informatik.uni-goettingen.de/MARS/>. DBIS, Informatics, Universität Göttingen.
- [9] Resourceful Reactive Rules (r<sup>3</sup>). <http://reverse.net/I5/r3/>. CENTRIA, UNL.