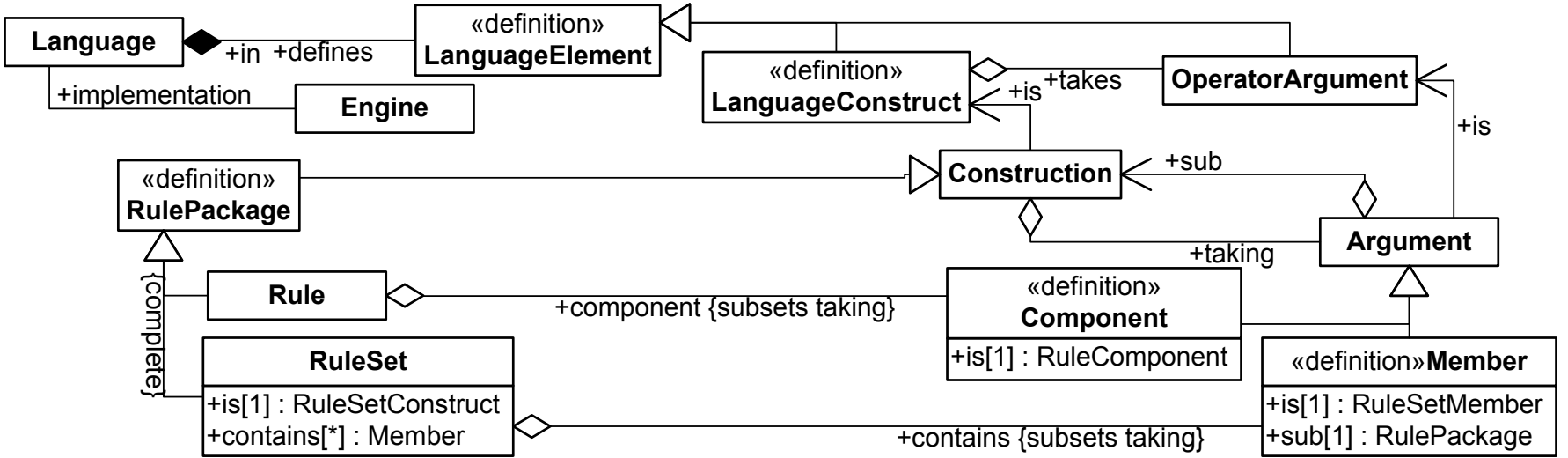
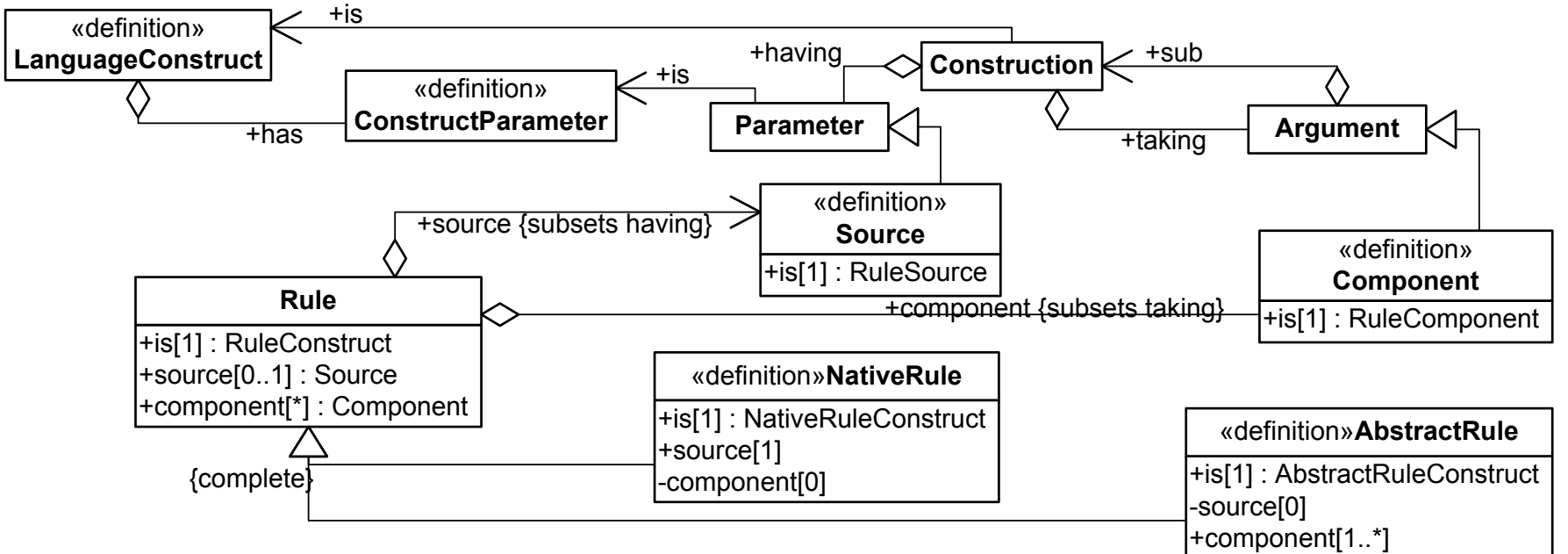
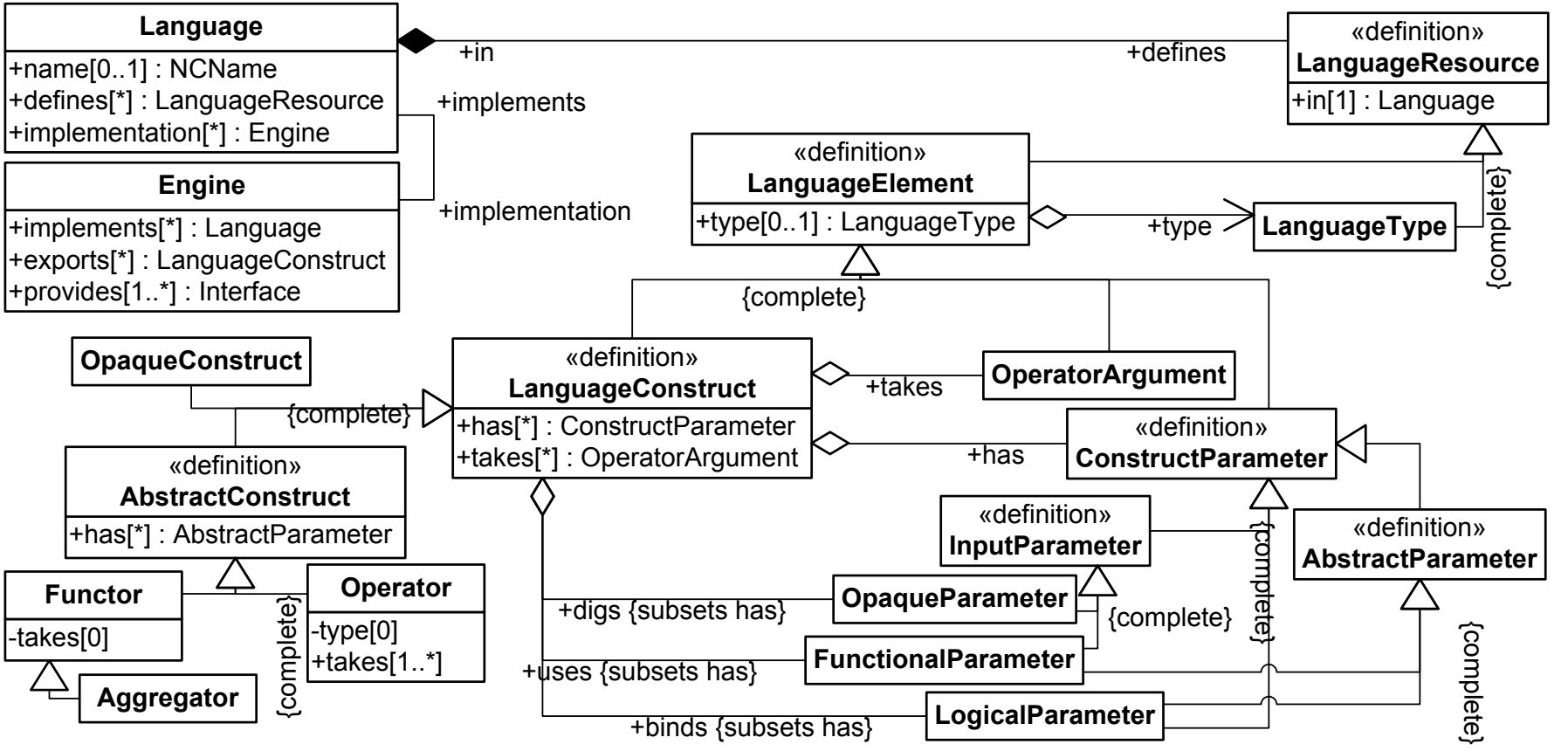
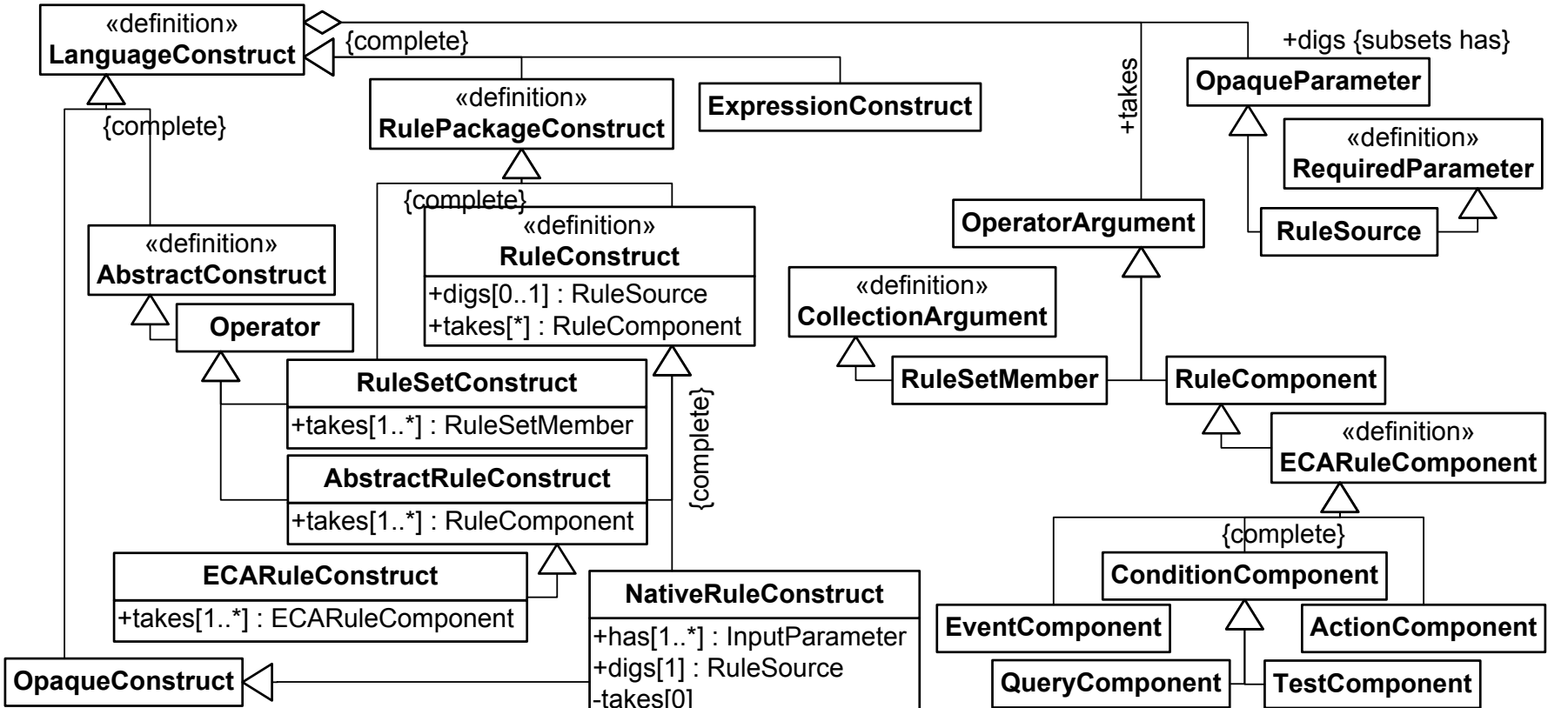
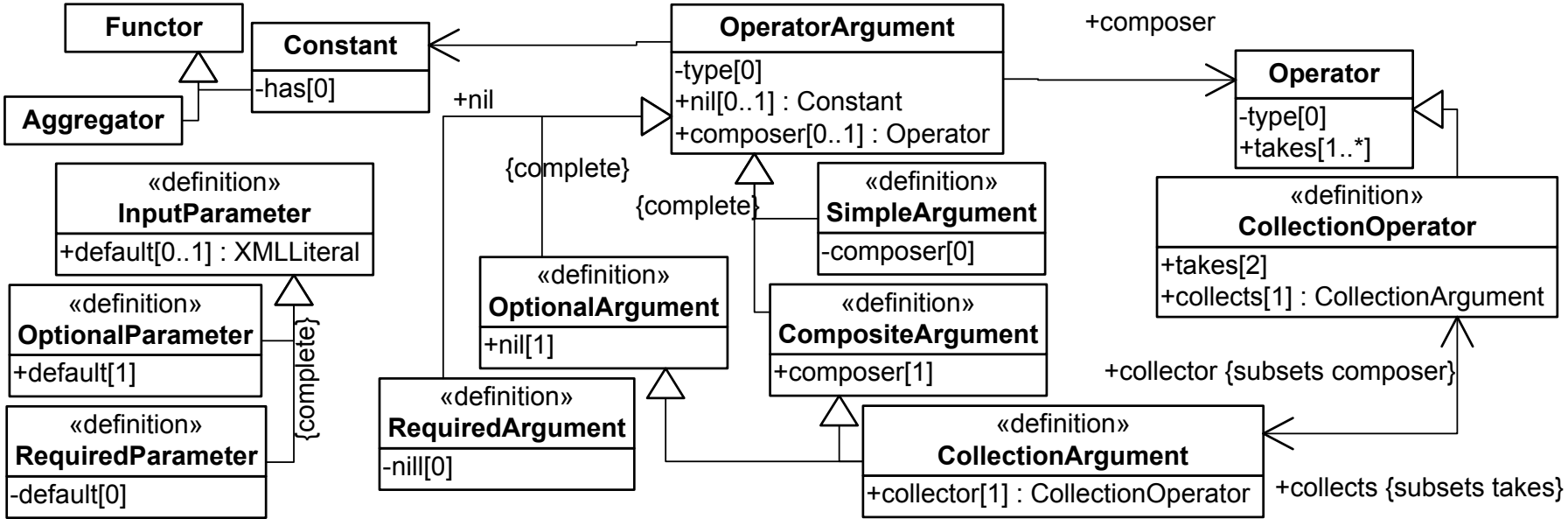
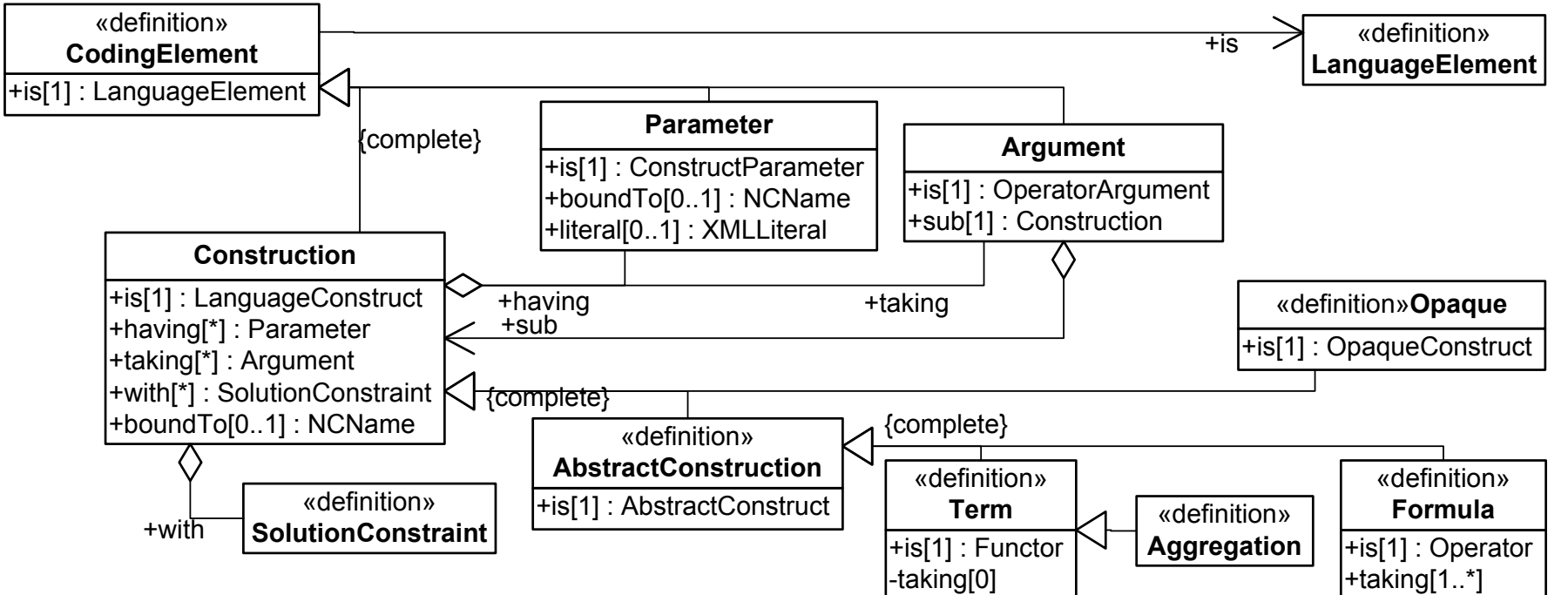

Language
+in
+defines
«definition»
LanguageElement
+implementation
Engine
«definition»
LanguageConstruct
+is
+takes
OperatorArgument
+is
«definition»
RulePackage
Construction
+sub
Argument
+taking
{complete}
Rule
+component {subsets taking}
«definition»
Component
+is[1] : RuleComponent
RuleSet
+is[1] : RuleSetConstruct
+contains[*] : Member
+contains {subsets taking}
«definition»Member
+is[1] : RuleSetMember
+sub[1] : RulePackage

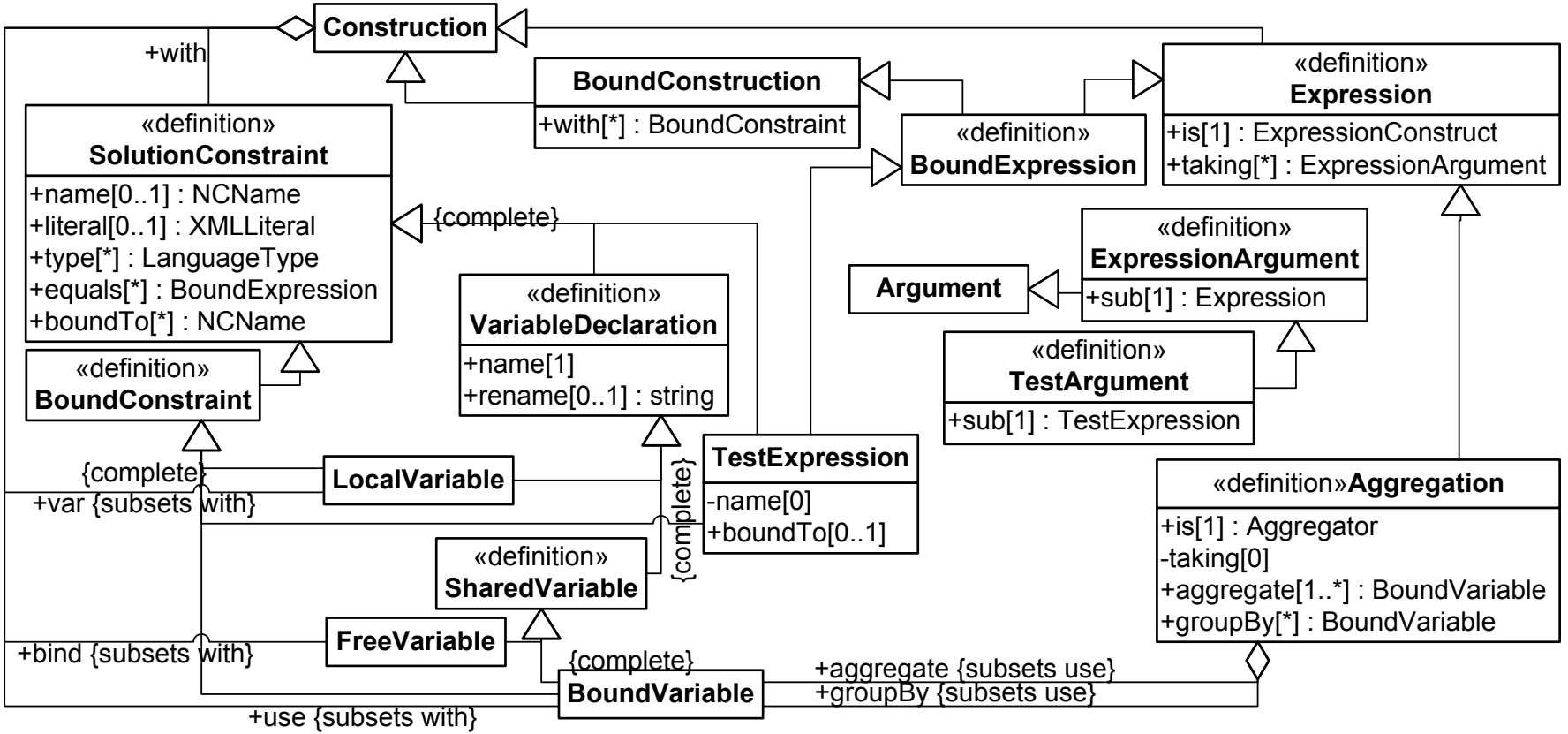

«definition»
LanguageConstruct
+is
+has
«definition»
ConstructParameter
+is
+having
Construction
+sub
Parameter
+taking
Argument
«definition»
Source
+is[1] : RuleSource
+source {subsets having}
«definition»
Component
+is[1] : RuleComponent
+component {subsets taking}
Rule
+is[1] : RuleConstruct
+source[0..1] : Source
+component[*] : Component
«definition»NativeRule
+is[1] : NativeRuleConstruct
+source[1]
-component[0]
{complete}
«definition»AbstractRule
+is[1] : AbstractRuleConstruct
-source[0]
+component[1..*]

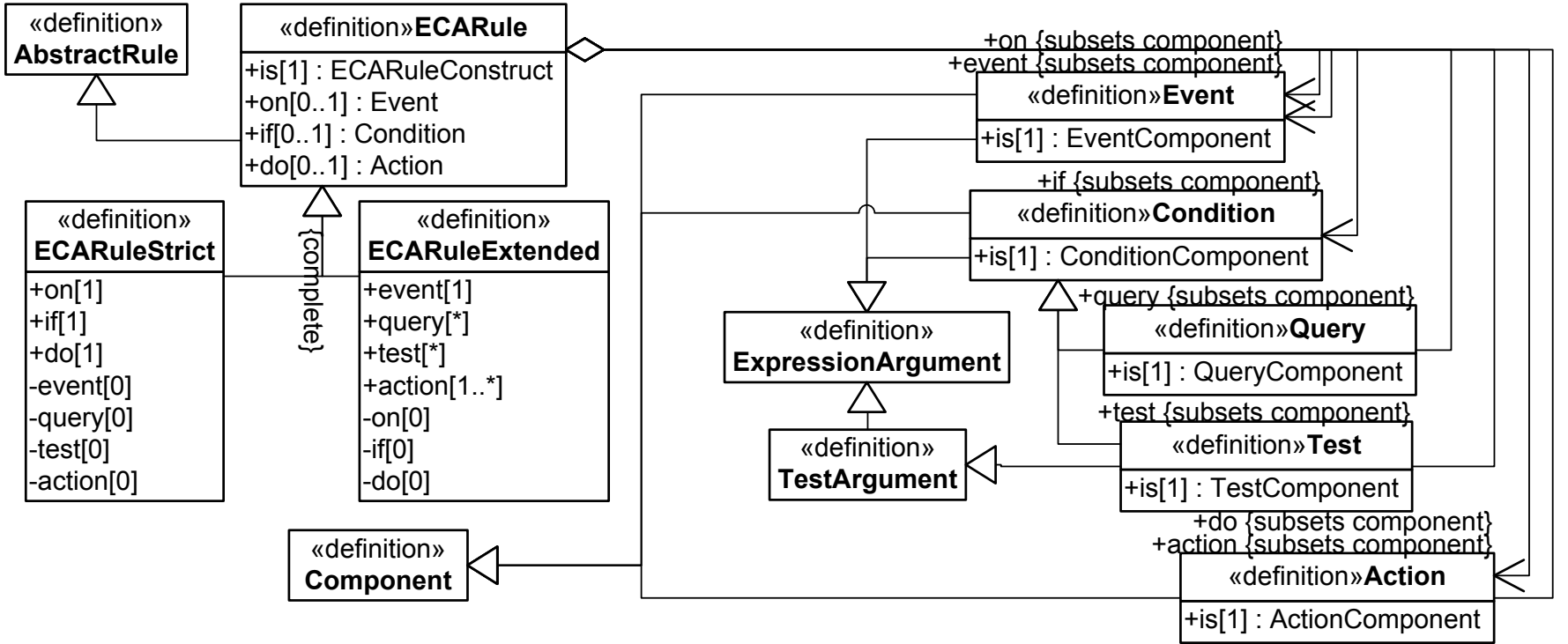

Language
+name[0..1] : NCName
+defines[*] : LanguageResource
+implementation[*] : Engine
+in
+defines
«definition»
LanguageResource
+in[1] : Language
+implements
+implementation
Engine
+implements[*] : Language
+exports[*] : LanguageConstruct
+provides[1..*] : Interface
«definition»
LanguageElement
+type[0..1] : LanguageType
+type
LanguageType
{complete}
{complete}
OpaqueConstruct
{complete}
«definition»
LanguageConstruct
+has[*] : ConstructParameter
+takes[*] : OperatorArgument
+takes
OperatorArgument
+has
«definition»
ConstructParameter
«definition»
AbstractConstruct
+has[*] : AbstractParameter
«definition»
InputParameter
«definition»
AbstractParameter
Functor
-takes[0]
Operator
-type[0]
+takes[1..*]
Aggregator
{complete}
+digs {subsets has}
OpaqueParameter
{complete}
{complete}
+uses {subsets has}
FunctionalParameter
+binds {subsets has}
LogicalParameter
{complete}

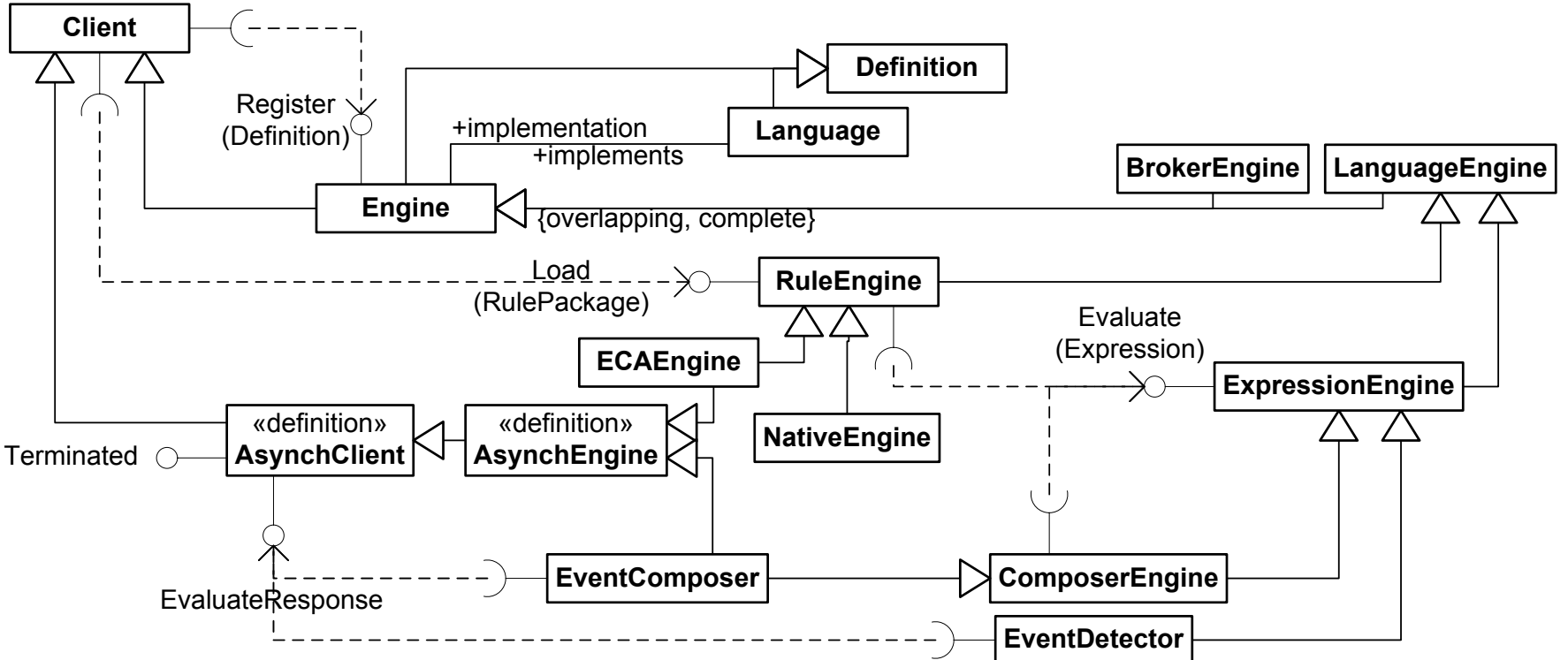

«definition»
LanguageConstruct
{complete}
{complete}
«definition»
RulePackageConstruct
ExpressionConstruct
+takes
+digs {subsets has}
OpaqueParameter
«definition»
RequiredParameter
RuleSource
{complete}
«definition»
RuleConstruct
+digs[0..1] : RuleSource
+takes[*] : RuleComponent
OperatorArgument
«definition»
AbstractConstruct
Operator
«definition»
CollectionArgument
RuleSetMember
RuleComponent
RuleSetConstruct
+takes[1..*] : RuleSetMember
{complete}
«definition»
ECARuleComponent
AbstractRuleConstruct
+takes[1..*] : RuleComponent
{complete}
ECARuleConstruct
+takes[1..*] : ECARuleComponent
NativeRuleConstruct
+has[1..*] : InputParameter
+digs[1] : RuleSource
-takes[0]
ConditionComponent
EventComponent
ActionComponent
QueryComponent
TestComponent
OpaqueConstruct

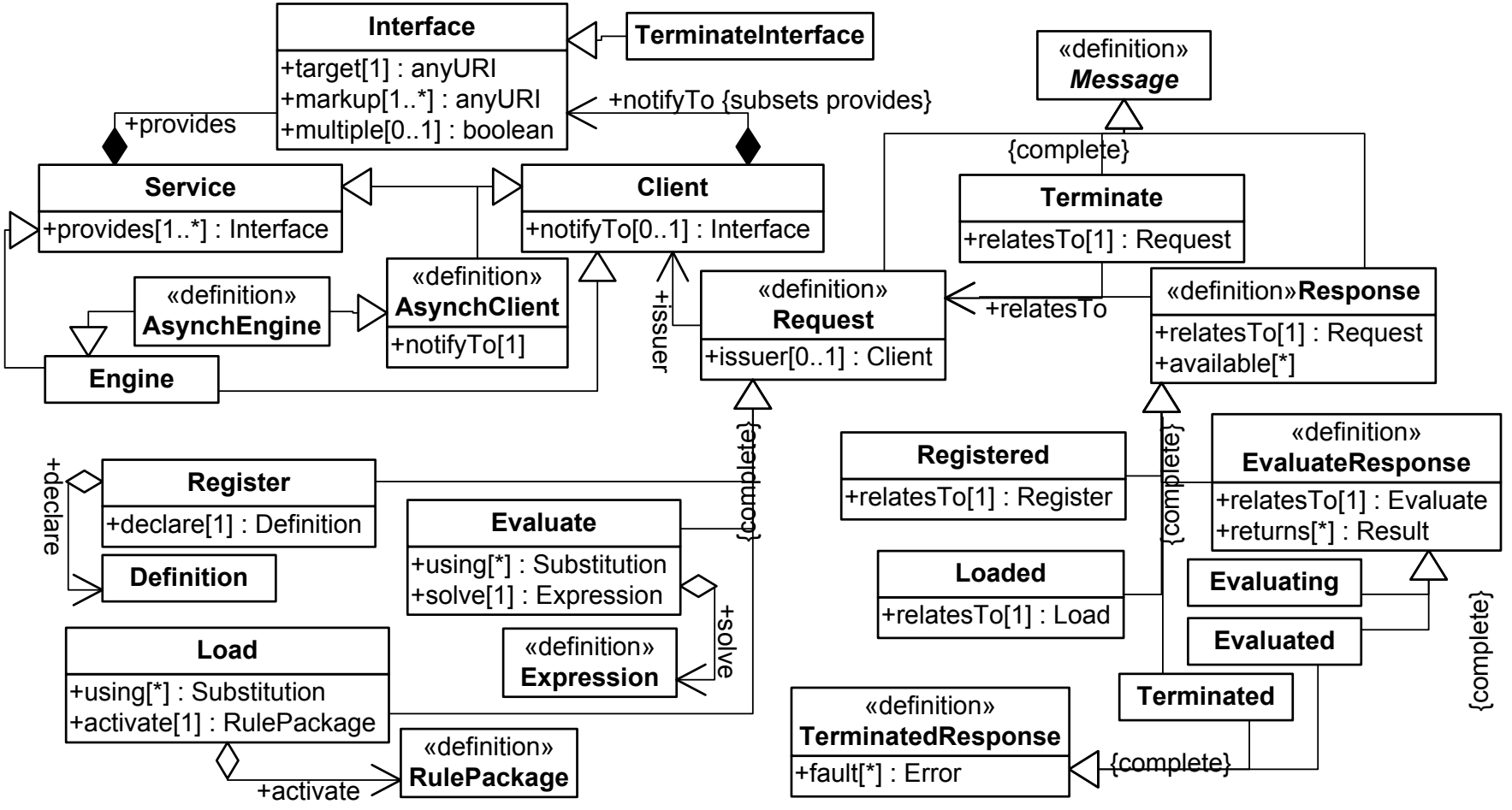

Functor
Aggregator
Constant
-has[0]
OperatorArgument
-type[0]
+nil[0..1] : Constant
+composer[0..1] : Operator
+composer
Operator
-type[0]
+takes[1..*]
+nil
{complete}
{complete}
«definition»
InputParameter
+default[0..1] : XMLLiteral
«definition»
SimpleArgument
-composer[0]
«definition»
CollectionOperator
+takes[2]
+collects[1] : CollectionArgument
«definition»
OptionalParameter
+default[1]
«definition»
OptionalArgument
+nil[1]
«definition»
CompositeArgument
+composer[1]
{complete}
+collector {subsets composer}
«definition»
RequiredParameter
-default[0]
«definition»
RequiredArgument
-nill[0]
«definition»
CollectionArgument
+collector[1] : CollectionOperator
+collects {subsets takes}

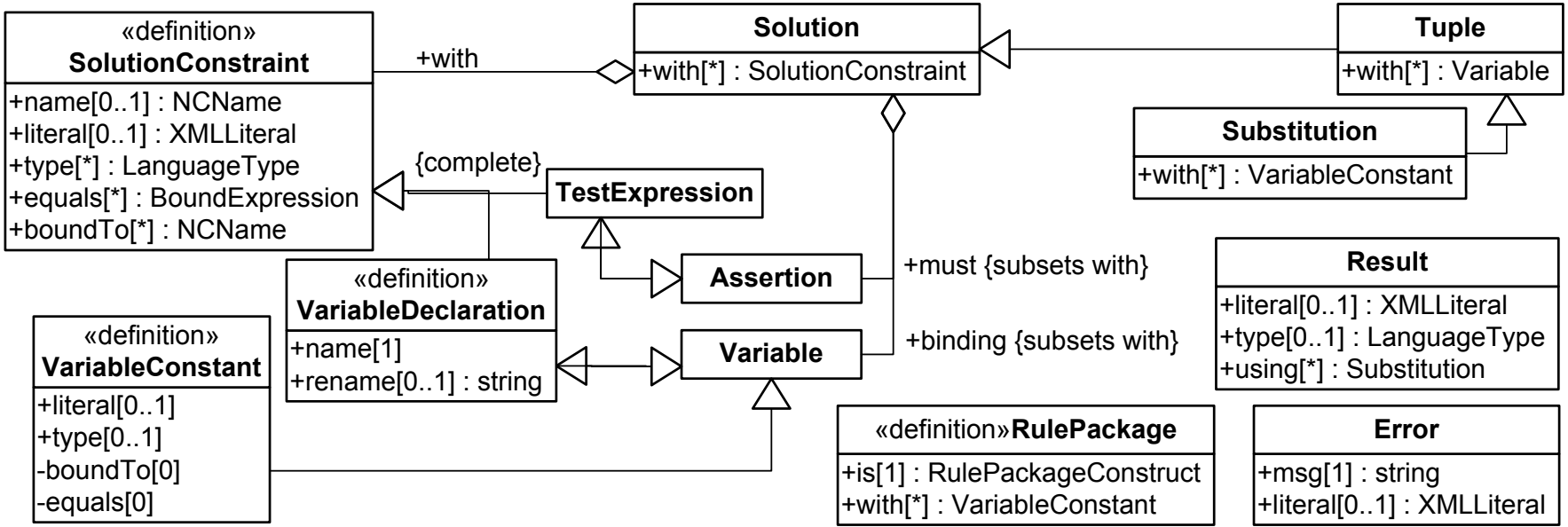

«definition»
CodingElement
+is[1] : LanguageElement
+is
«definition»
LanguageElement
{complete}
Parameter
+is[1] : ConstructParameter
+boundTo[0..1] : NCName
+literal[0..1] : XMLLiteral
Argument
+is[1] : OperatorArgument
+sub[1] : Construction
Construction
+is[1] : LanguageConstruct
+having[*] : Parameter
+taking[*] : Argument
+with[*] : SolutionConstraint
+boundTo[0..1] : NCName
+having
+sub
+taking
«definition»Opaque
+is[1] : OpaqueConstruct
{complete}
«definition»
AbstractConstruction
+is[1] : AbstractConstruct
{complete}
«definition»
Term
+is[1] : Functor
-taking[0]
«definition»
Aggregation
«definition»
Formula
+is[1] : Operator
+taking[1..*]
+with
«definition»
SolutionConstraint

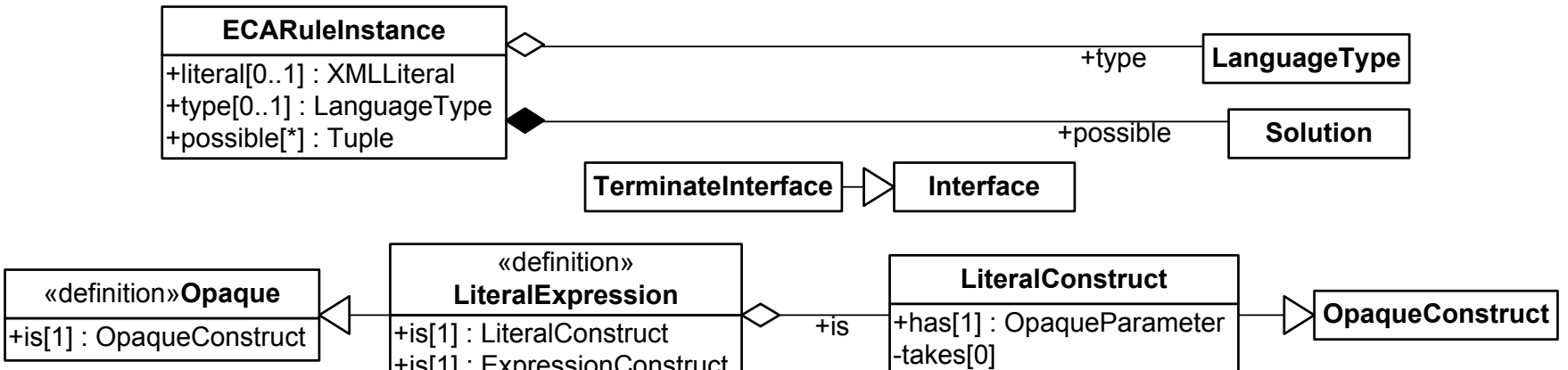

Construction
+with
BoundConstruction
+with[*] : BoundConstraint
«definition»
Expression
+is[1] : ExpressionConstruct
+taking[*] : ExpressionArgument
«definition»
SolutionConstraint
+name[0..1] : NCName
+literal[0..1] : XMLLiteral
+type[*] : LanguageType
+equals[*] : BoundExpression
+boundTo[*] : NCName
{complete}
«definition»
BoundExpression
«definition»
ExpressionArgument
+sub[1] : Expression
Argument
«definition»
VariableDeclaration
+name[1]
+rename[0..1] : string
«definition»
TestArgument
+sub[1] : TestExpression
«definition»
BoundConstraint
{complete}
LocalVariable
+var {subsets with}
TestExpression
-name[0]
+boundTo[0..1]
{complete}
«definition»Aggregation
+is[1] : Aggregator
-taking[0]
+aggregate[1..*] : BoundVariable
+groupBy[*] : BoundVariable
«definition»
SharedVariable
FreeVariable
+bind {subsets with}
{complete}
BoundVariable
+aggregate {subsets use}
+groupBy {subsets use}
+use {subsets with}

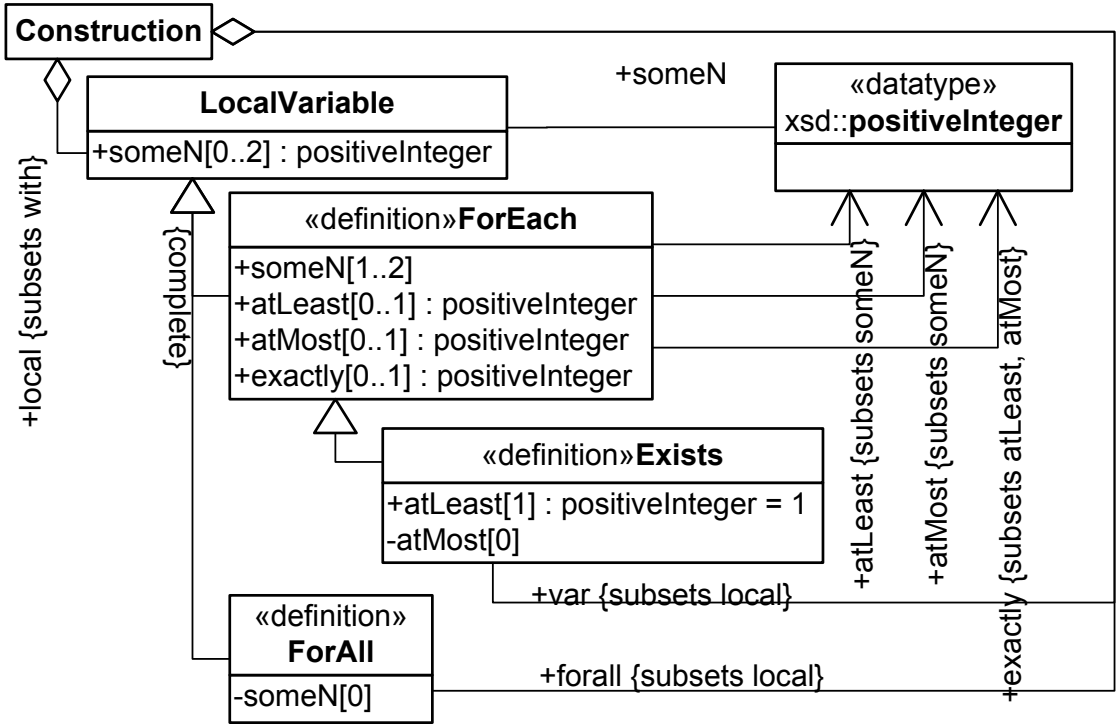

«definition»
AbstractRule
«definition»ECARule
+is[1] : ECARuleConstruct
+on[0..1] : Event
+if[0..1] : Condition
+do[0..1] : Action
+on {subsets component}
+event {subsets component}
«definition»Event
+is[1] : EventComponent
+if {subsets component}
«definition»Condition
+is[1] : ConditionComponent
+query {subsets component}
«definition»Query
+is[1] : QueryComponent
+test {subsets component}
«definition»Test
+is[1] : TestComponent
+do {subsets component}
+action {subsets component}
«definition»Action
+is[1] : ActionComponent
«definition»
ECARuleStrict
+on[1]
+if[1]
+do[1]
-event[0]
-query[0]
-test[0]
-action[0]
{complete}
«definition»
ECARuleExtended
+event[1]
+query[*]
+test[*]
+action[1..*]
-on[0]
-if[0]
-do[0]
«definition»
ExpressionArgument
«definition»
TestArgument
«definition»
Component

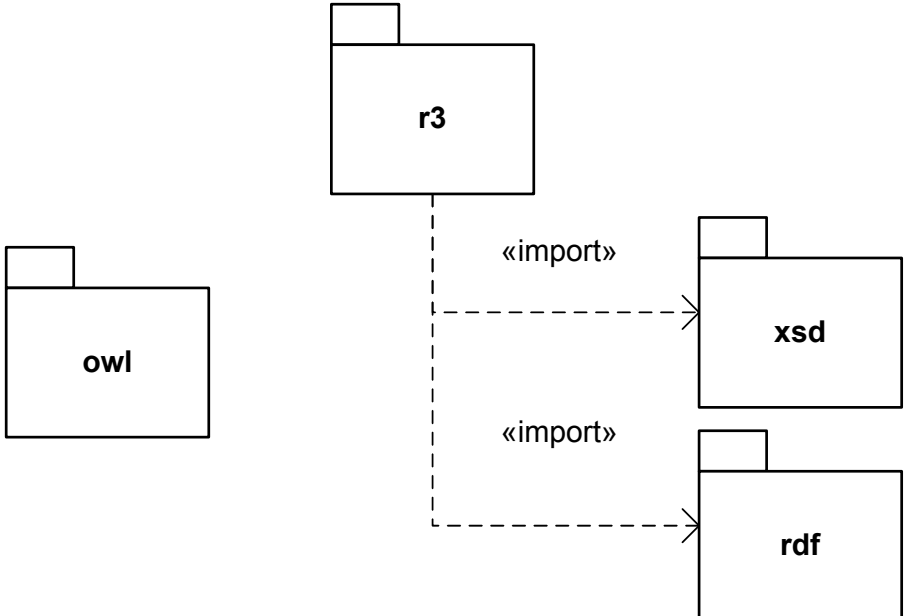

Client
Definition
Register
(Definition)
+implementation
+implements
Language
BrokerEngine
LanguageEngine
Engine
{overlapping, complete}
Load
(RulePackage)
RuleEngine
Evaluate
(Expression)
ECAEngine
ExpressionEngine
«definition»
AsynchClient
«definition»
AsynchEngine
NativeEngine
Terminated
EvaluateResponse
EventComposer
ComposerEngine
EventDetector


Interface
+target[1] : anyURI
+markup[1..*] : anyURI
+multiple[0..1] : boolean
TerminateInterface
+provides
+notifyTo {subsets provides}
Service
+provides[1..*] : Interface
Client
+notifyTo[0..1] : Interface
«definition»
AsynchEngine
«definition»
AsynchClient
+notifyTo[1]
Engine
+issuer
«definition»
Request
+issuer[0..1] : Client
«definition»
Message
{complete}
Terminate
+relatesTo[1] : Request
+relatesTo
«definition»Response
+relatesTo[1] : Request
+available[*]
{complete}
Register
+declare[1] : Definition
+declare
Definition
Evaluate
+using[*] : Substitution
+solve[1] : Expression
+solve
«definition»
Expression
{complete}
Load
+using[*] : Substitution
+activate[1] : RulePackage
+activate
«definition»
RulePackage
Registered
+relatesTo[1] : Register
Loaded
+relatesTo[1] : Load
«definition»
EvaluateResponse
+relatesTo[1] : Evaluate
+returns[*] : Result
Evaluating
Evaluated
Terminated
{complete}
«definition»
TerminatedResponse
+fault[*] : Error
{complete}


«definition»
SolutionConstraint
+name[0..1] : NCName
+literal[0..1] : XMLLiteral
+type[*] : LanguageType
+equals[*] : BoundExpression
+boundTo[*] : NCName
+with
Solution
+with[*] : SolutionConstraint
Tuple
+with[*] : Variable
Substitution
+with[*] : VariableConstant
{complete}
TestExpression
Assertion
+must {subsets with}
+binding {subsets with}
Result
+literal[0..1] : XMLLiteral
+type[0..1] : LanguageType
+using[*] : Substitution
«definition»
VariableDeclaration
+name[1]
+rename[0..1] : string
Variable
«definition»
VariableConstant
+literal[0..1]
+type[0..1]
-boundTo[0]
-equals[0]
«definition»RulePackage
+is[1] : RulePackageConstruct
+with[*] : VariableConstant
Error
+msg[1] : string
+literal[0..1] : XMLLiteral


ECARuleInstance
+literal[0..1] : XMLLiteral
+type[0..1] : LanguageType
+possible[*] : Tuple
+type
LanguageType
+possible
Solution
TerminateInterface
Interface
«definition»Opaque
+is[1] : OpaqueConstruct
«definition»
LiteralExpression
+is[1] : LiteralConstruct
+is[1] : ExpressionConstruct
+is
LiteralConstruct
+has[1] : OpaqueParameter
-takes[0]
OpaqueConstruct


Construction
LocalVariable
+someN[0..2] : positiveInteger
+someN
«datatype»
xsd::positiveInteger
+local {subsets with}
{complete}
«definition»ForEach
+someN[1..2]
+atLeast[0..1] : positiveInteger
+atMost[0..1] : positiveInteger
+exactly[0..1] : positiveInteger
+atLeast {subsets someN}
+atMost {subsets someN}
+exactly {subsets atLeast, atMost}
«definition»Exists
+atLeast[1] : positiveInteger = 1
-atMost[0]
+var {subsets local}
«definition»
ForAll
-someN[0]
+forall {subsets local}


r3
owl
«import»
xsd
«import»
rdf