

OOA of MASL

xtuml.org

cortland.starrett@onefact.net



OOA of MASL

outline

- goals
- strategy
- OOA of MASL
- MASL loader
- next steps

goals

- capture MASL as a model
- parse and load correct MASL
- anticipate cloud architecture



strategy

- port MASL model compiler java model
- leverage working starting point
- low-risk with huge number of artifacts
- script conversion of java to MASL
- import into BridgePoint
- refine with action language

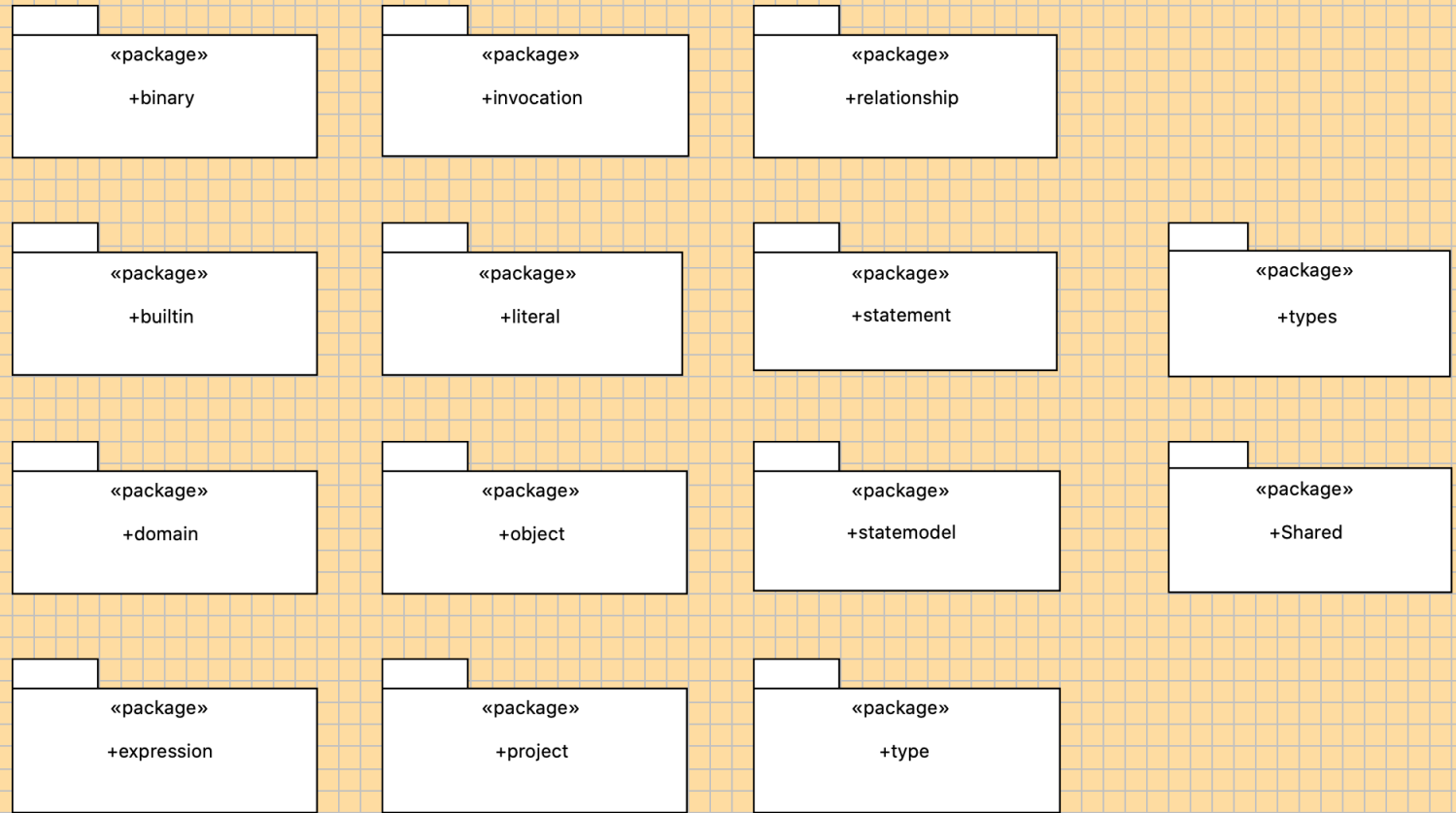


OOA of MASL

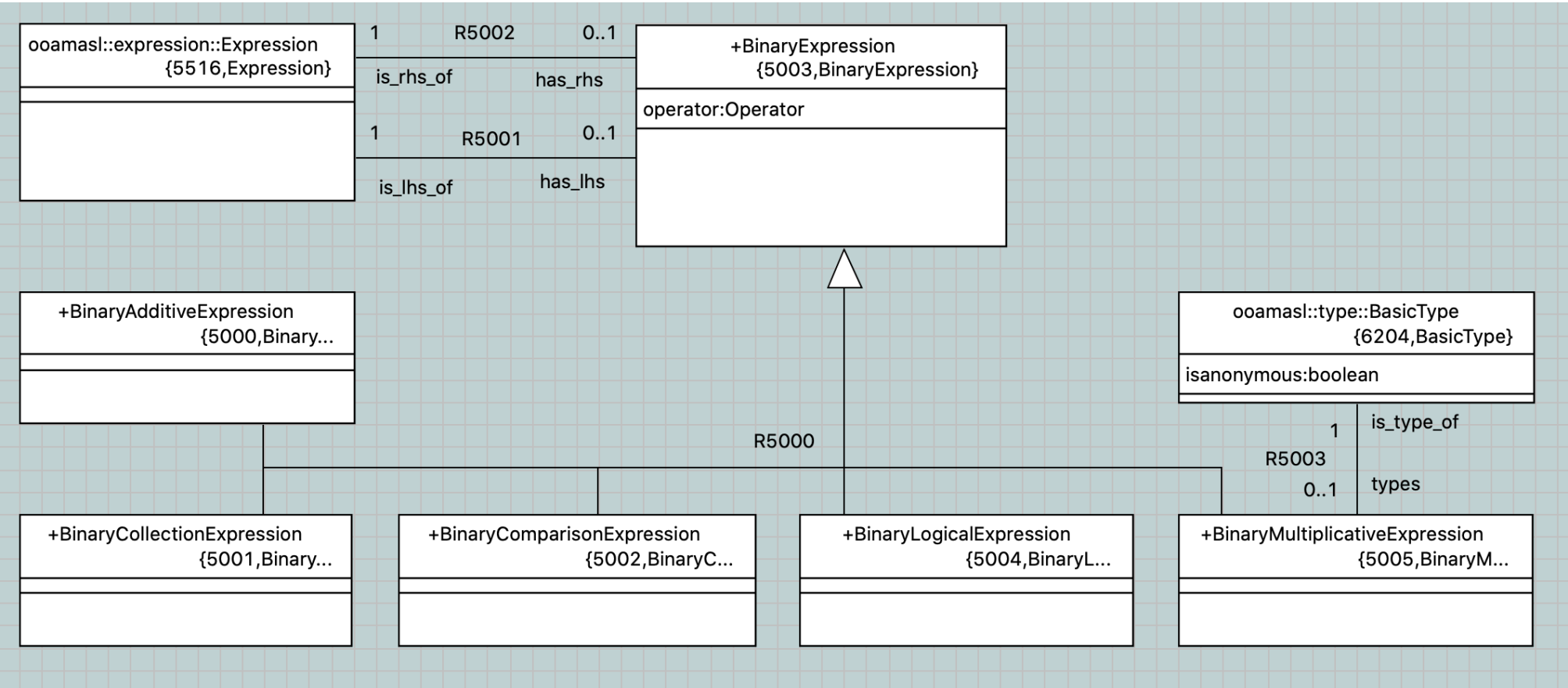
- Show ooamasl subsystem packages (ME and canvas)
- package references and "naked" OOA
- statistics
- deployment component (ME and canvas)
- tour of each subsystem

OOA of MASL: ooamasl packages

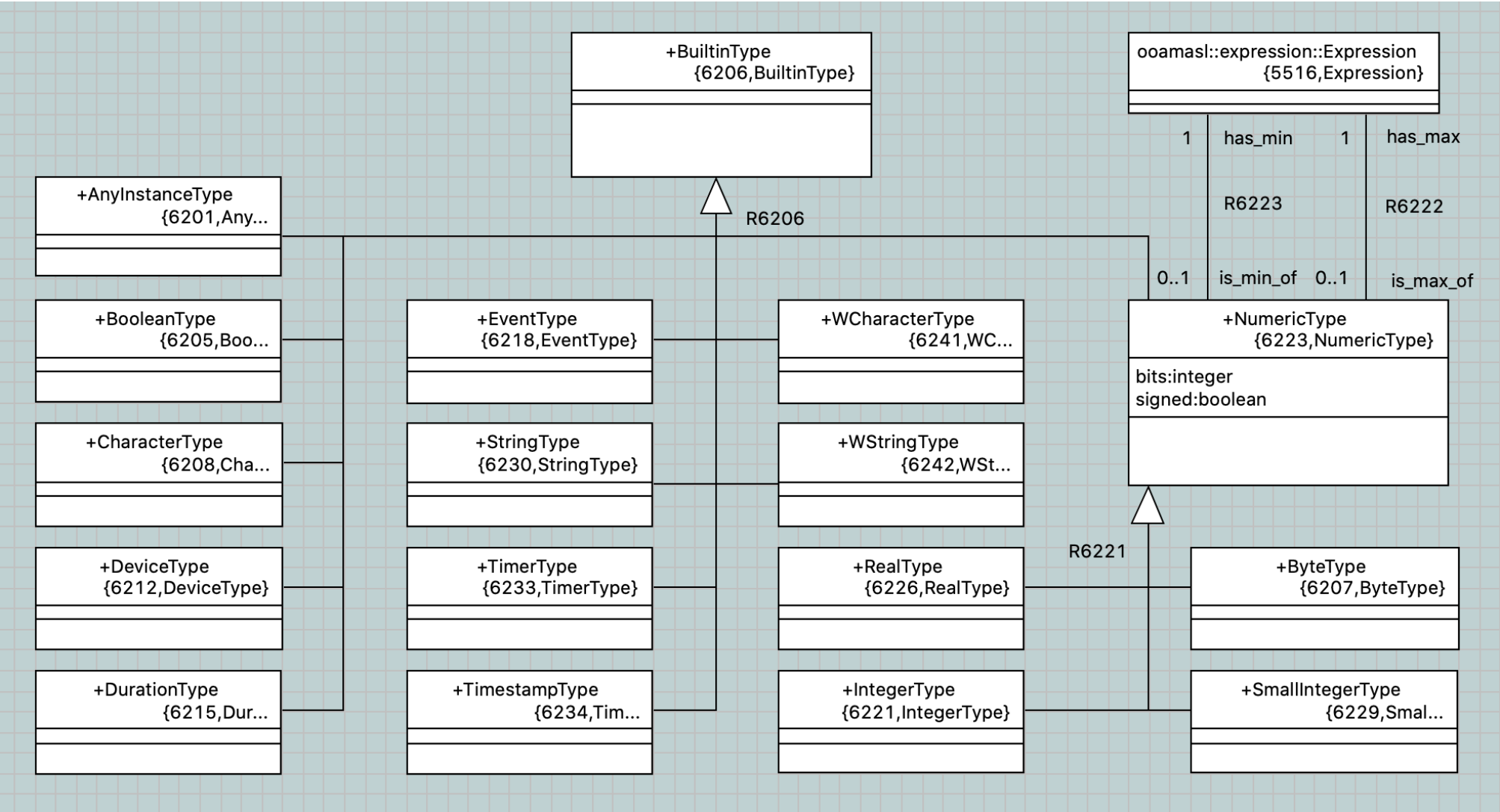
- ▼ ooamasl [mc 11745_loadmasl3]
 - ▶ binary
 - ▶ builtin
 - ▶ domain
 - ▶ expression
 - ▶ invocation
 - ▶ literal
 - ▶ object
 - ▶ project
 - ▶ relationship
 - ▶ Shared
 - ▶ statement
 - ▶ statemodel
 - ▶ type
 - ▶ types
- ▶ runtime [ciera master]
- ▶ > stratus [mc 11745_loadmasl3]



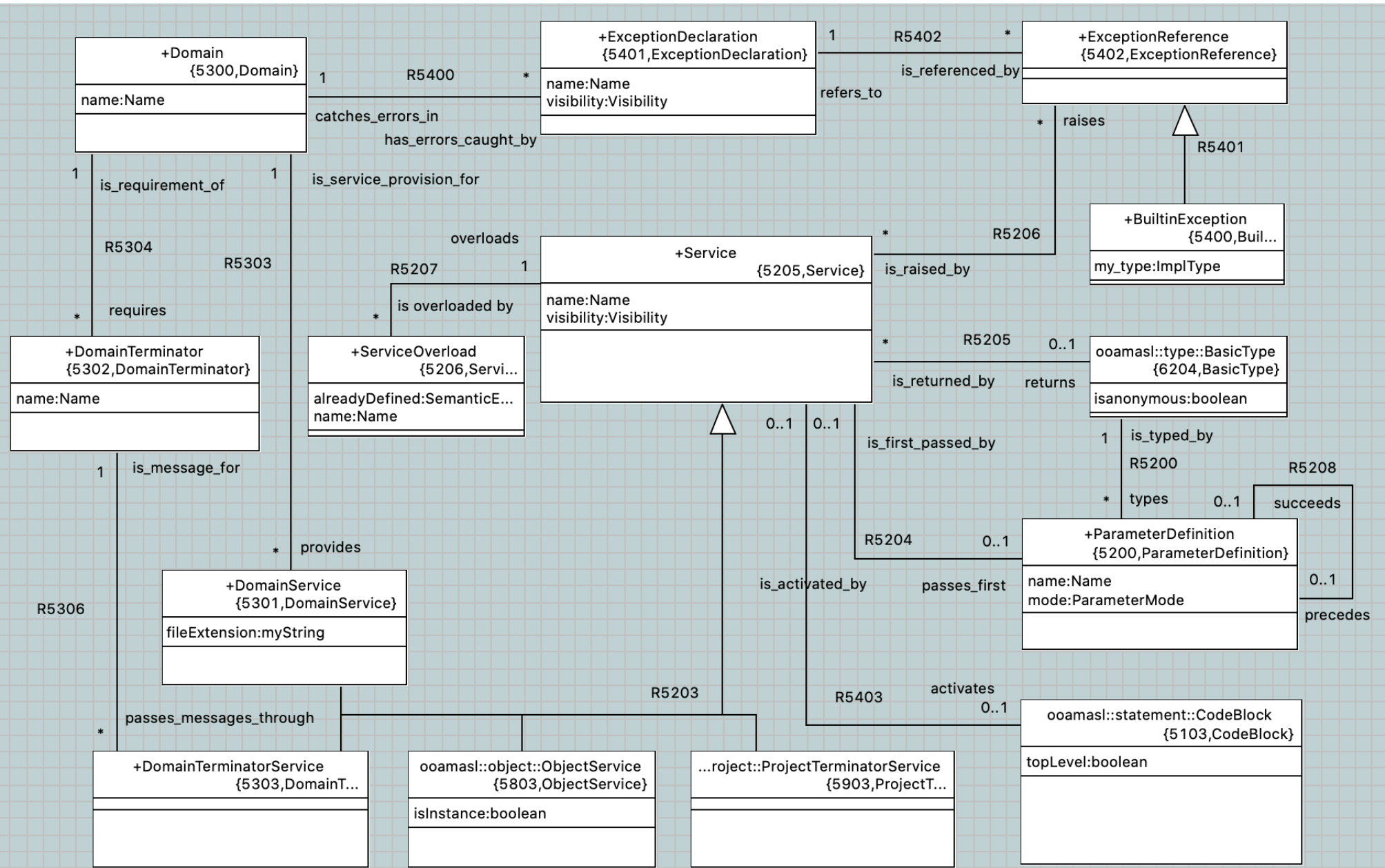
OOA of MASL: binary



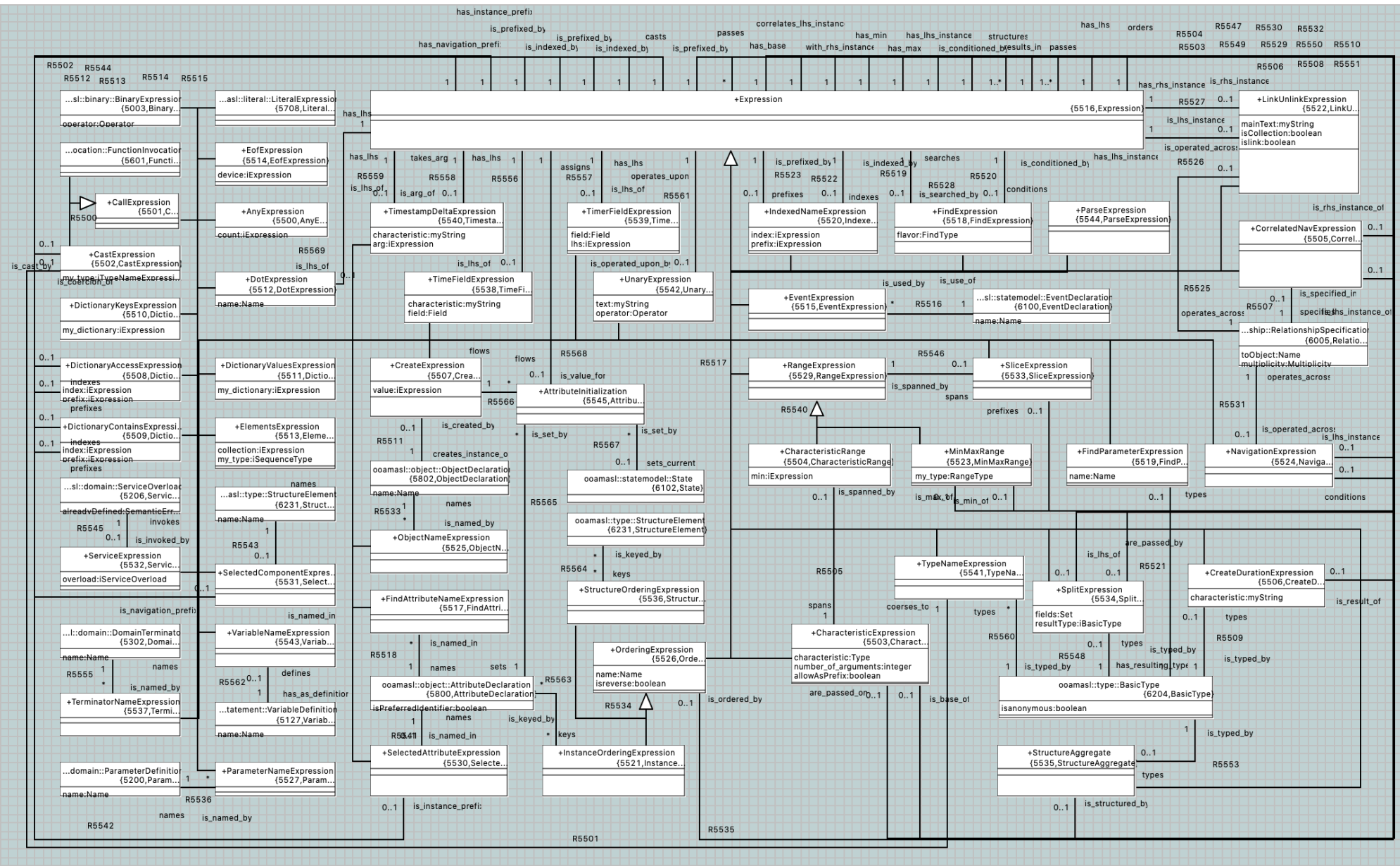
OOA of MASL: builtin



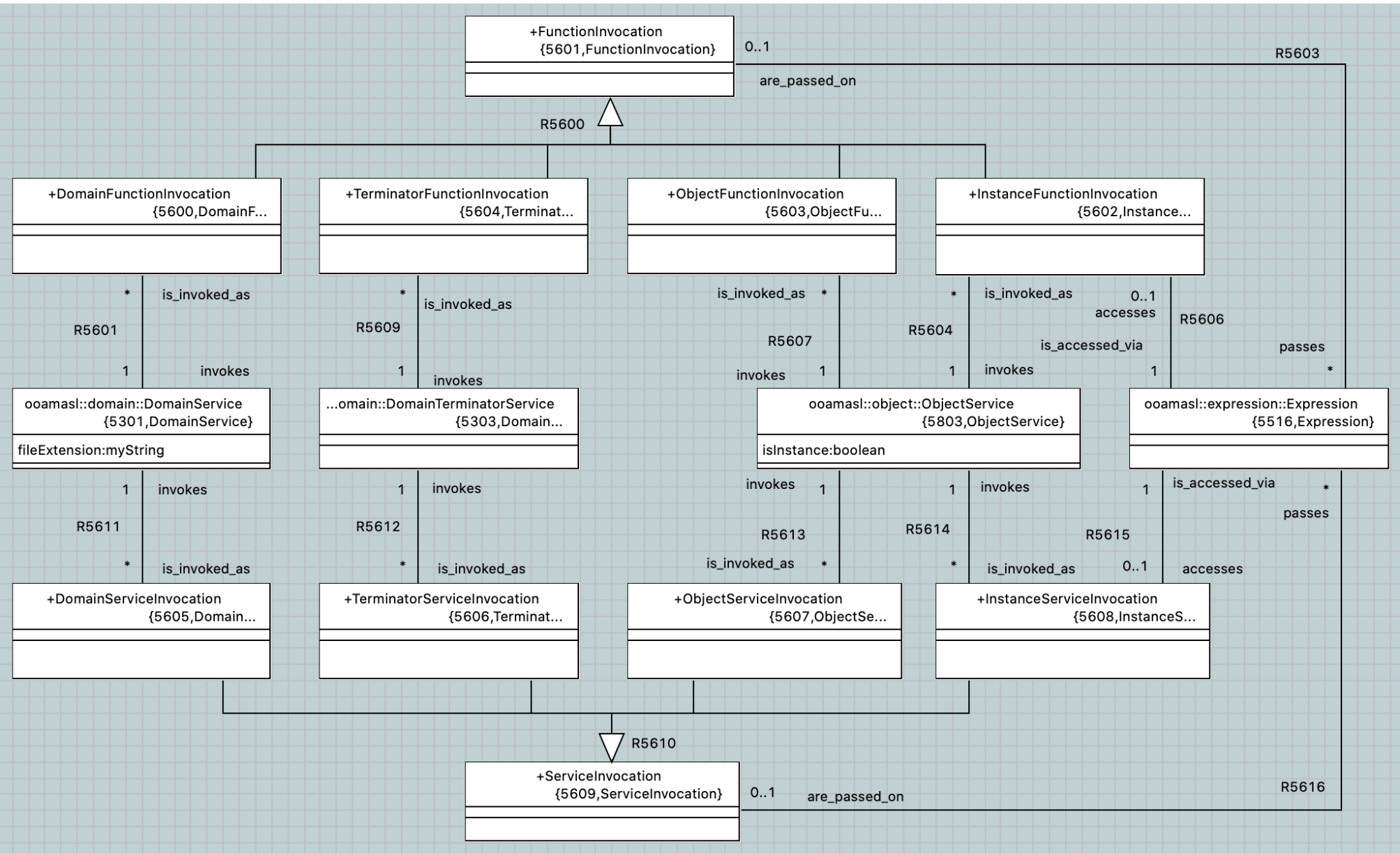
OOA of MASL: domain



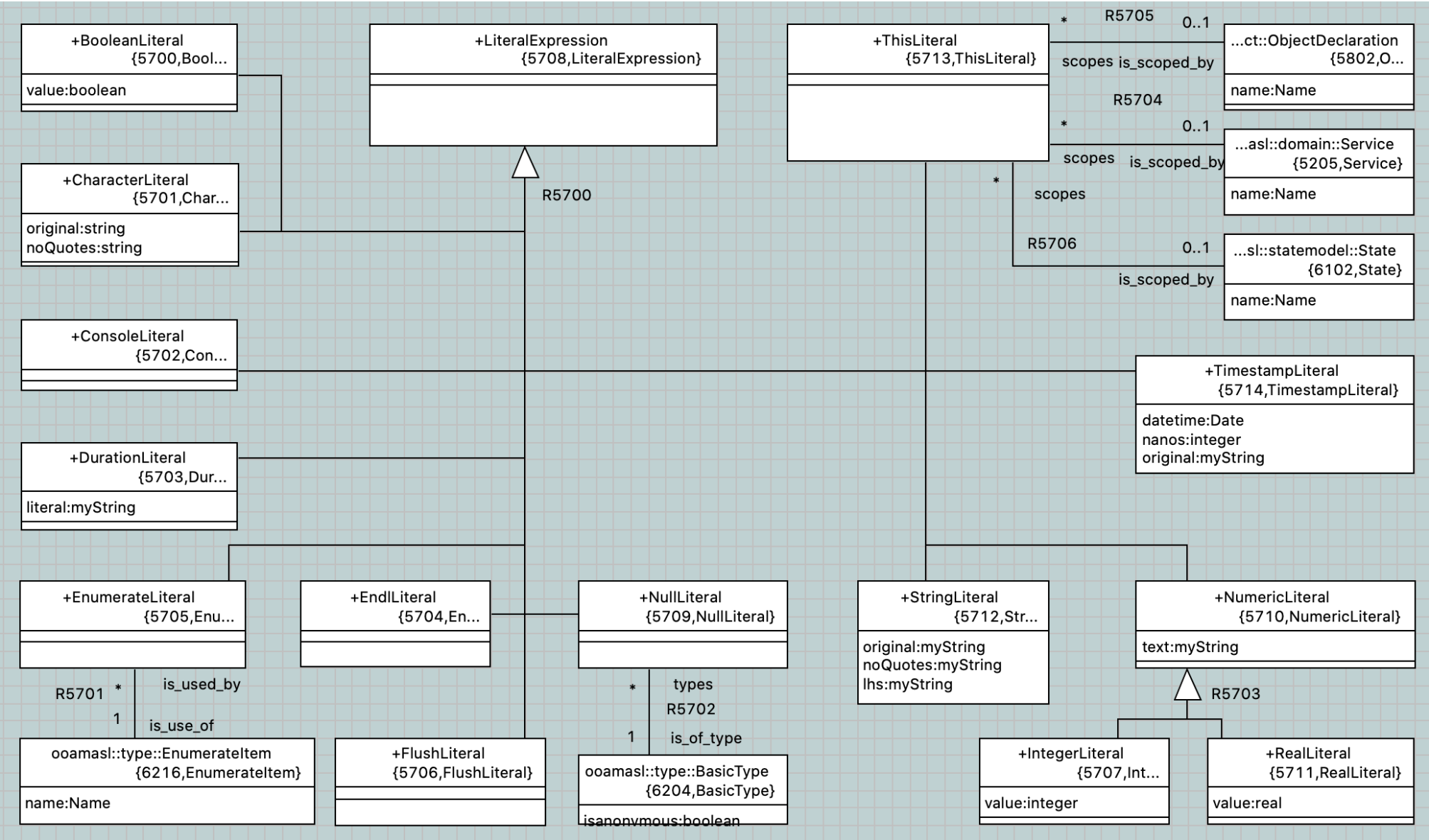
OOA of MASL: expression



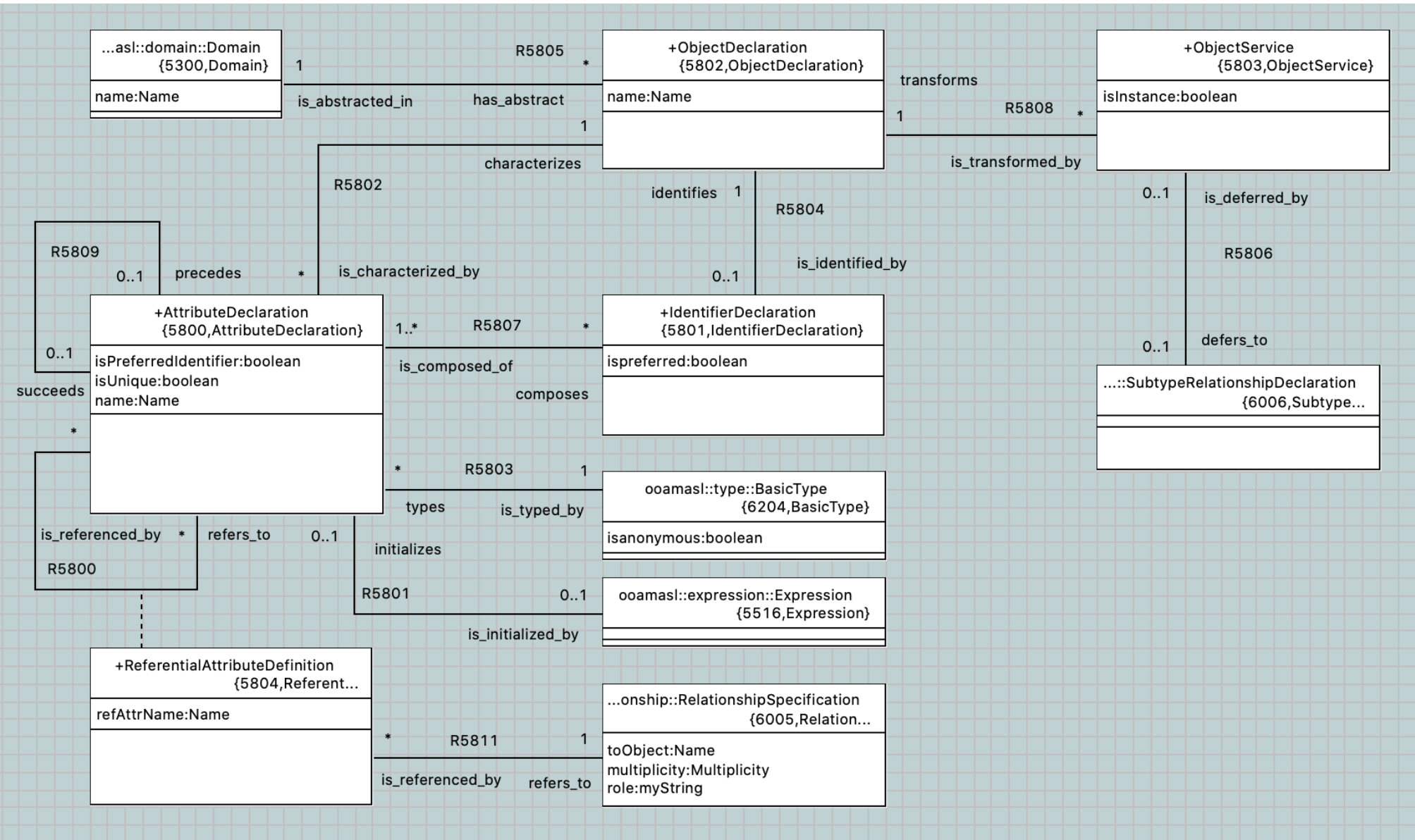
OOA of MASL: invocation



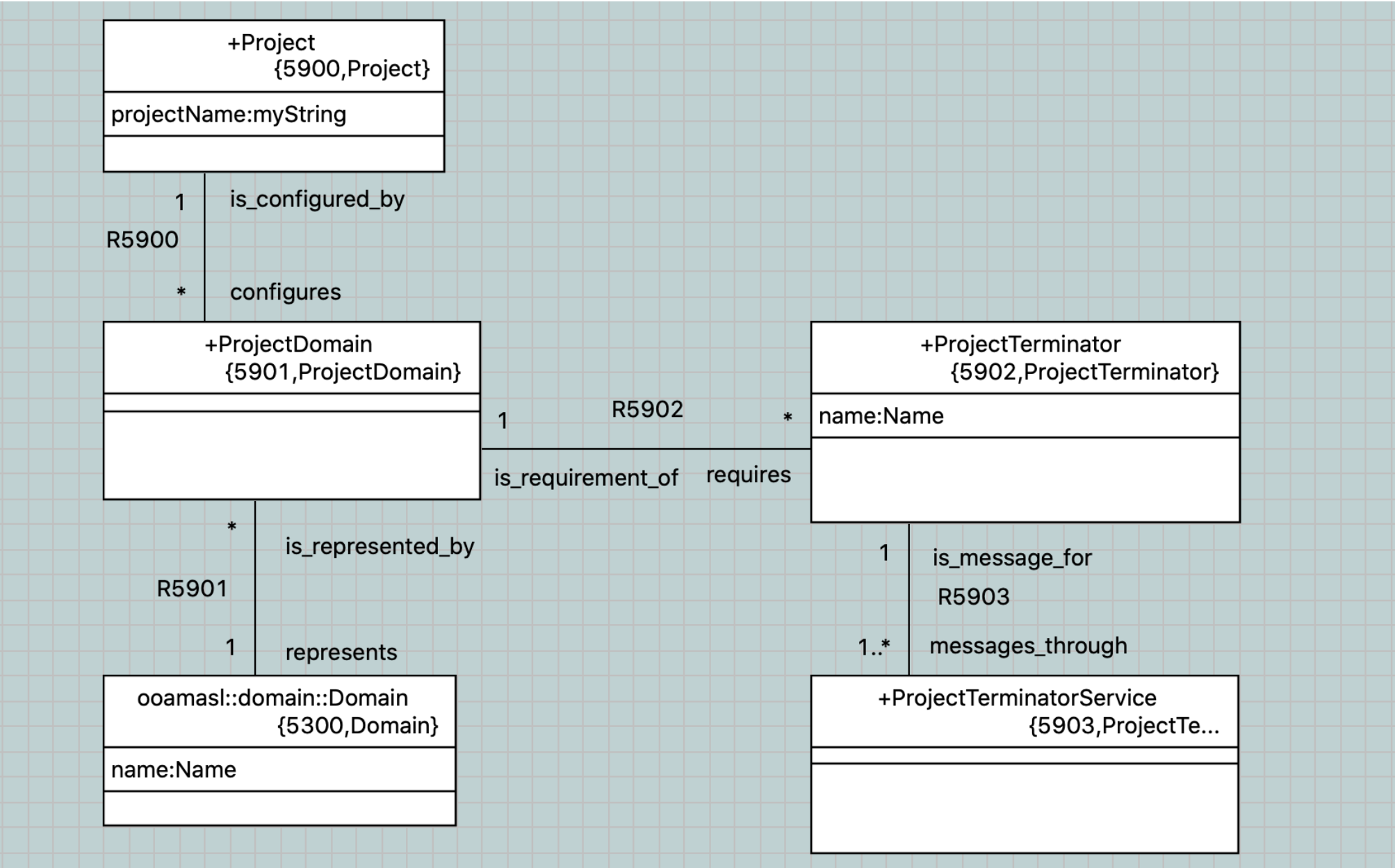
OOA of MASL: literal



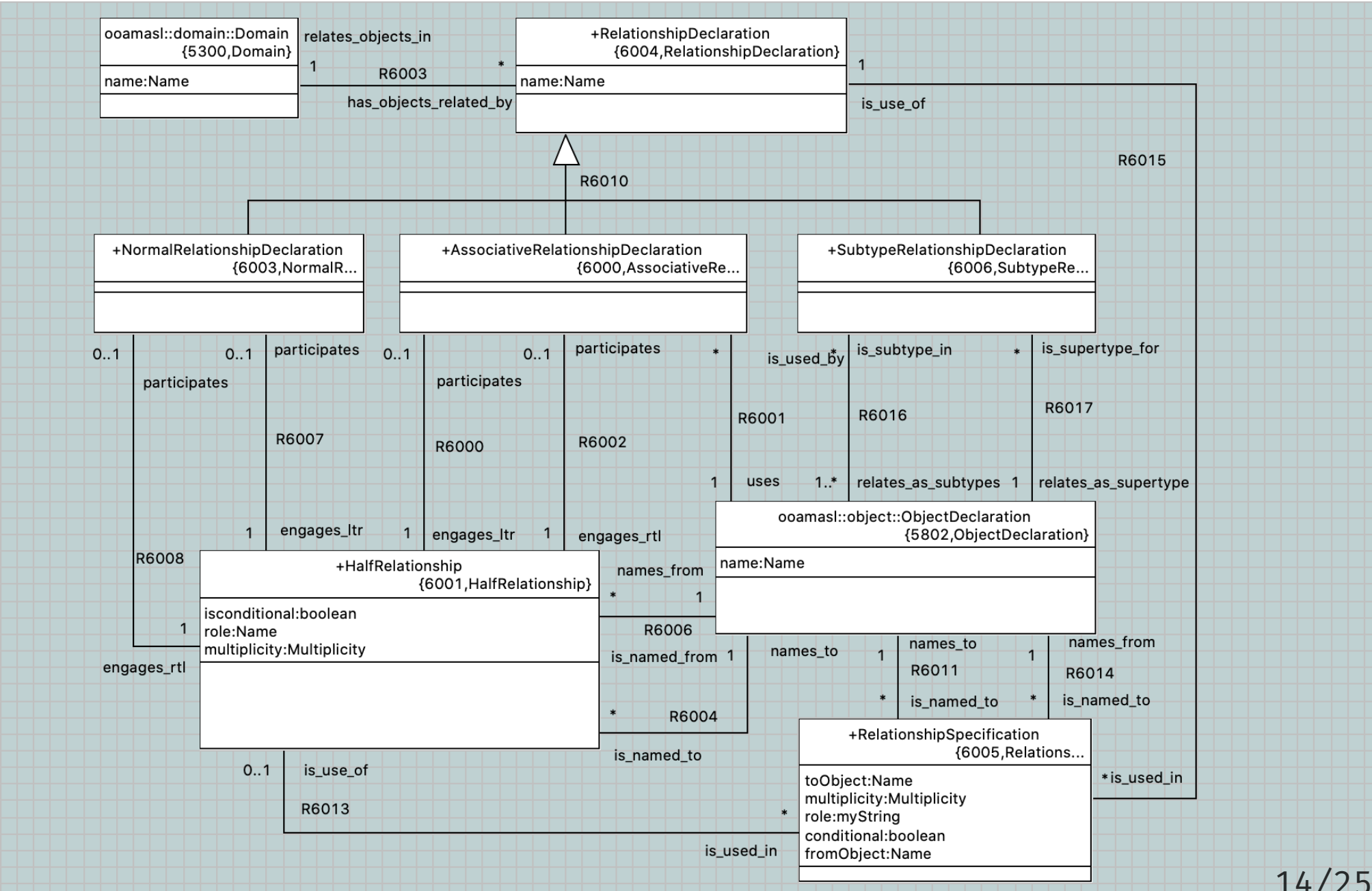
OOA of MASL: object



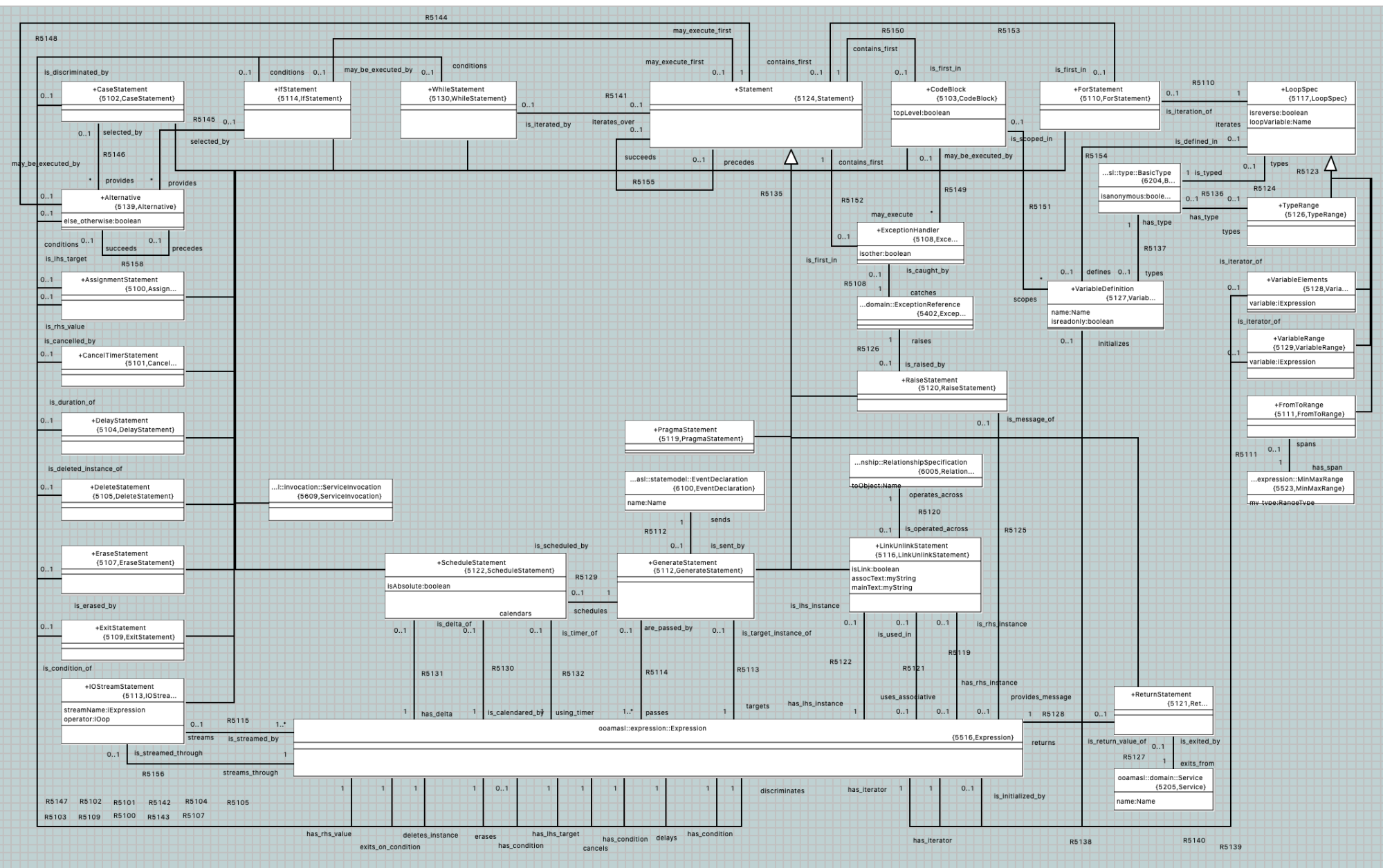
OOA of MASL: project



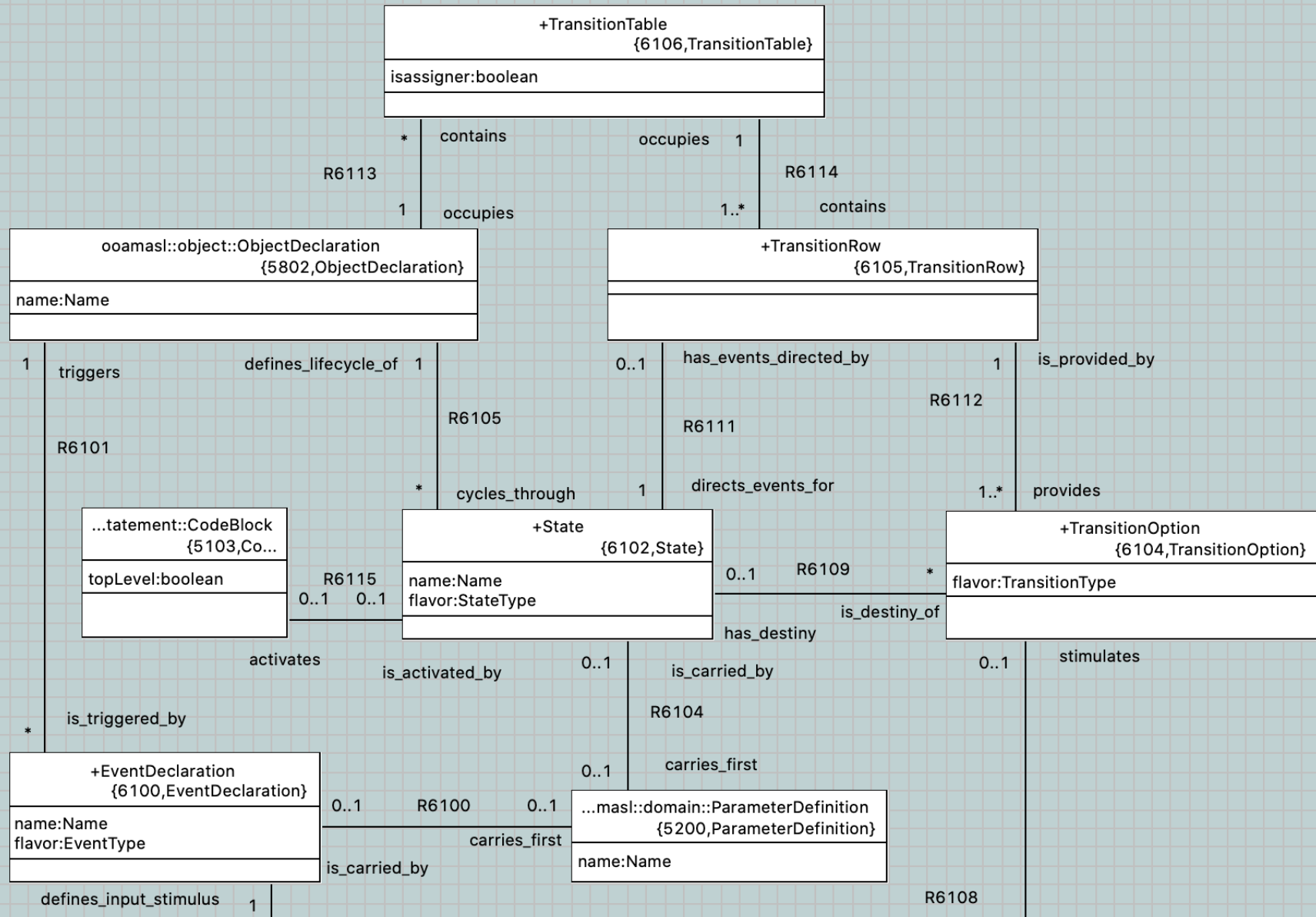
OOA of MASL: relationship



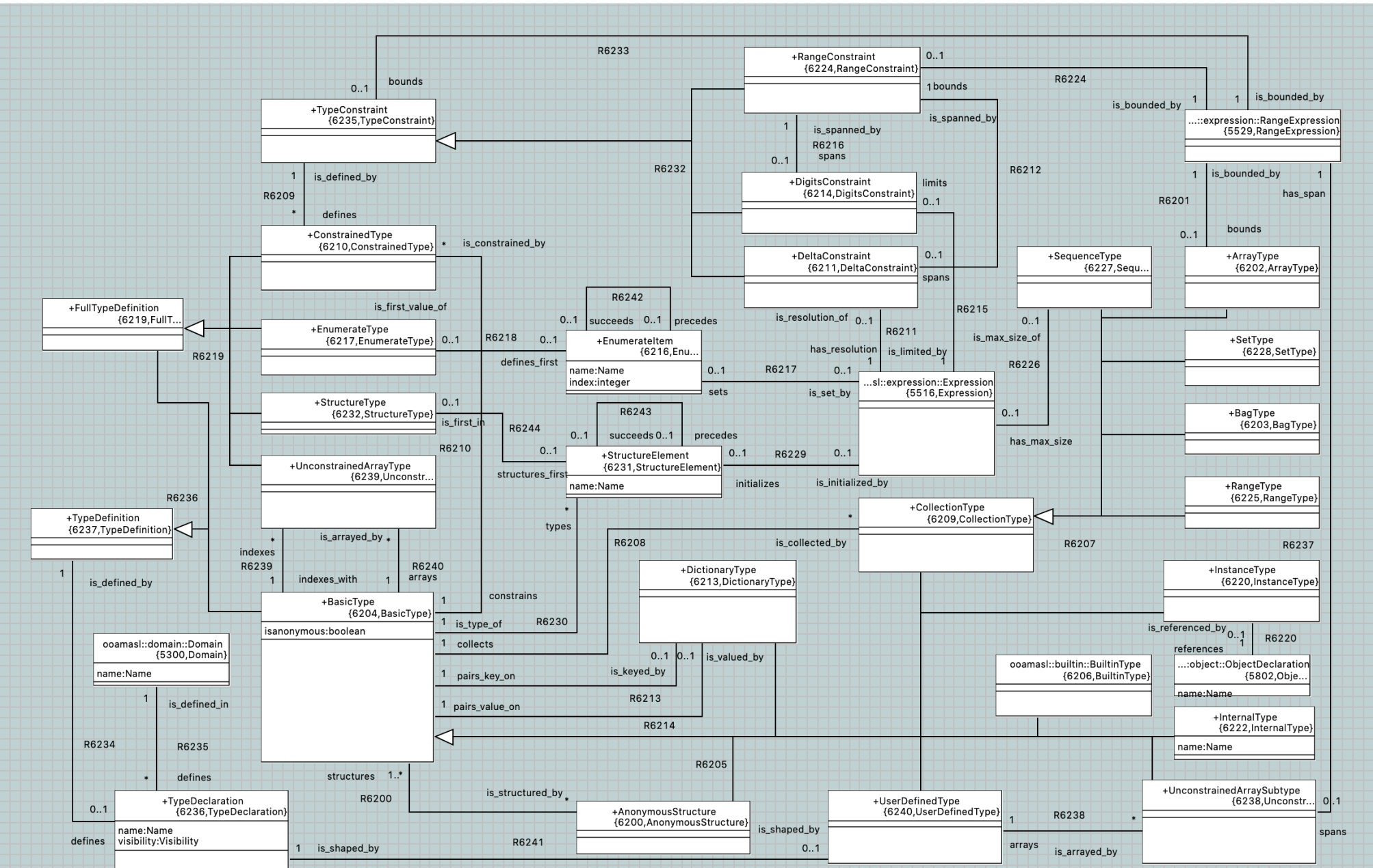
OOA of MASL: statement



OOA of MASL: statemodel



OOA of MASL: type



package references and "naked" OOA of MASL

- A package reference is a package linked to another package in another location.
- Package references allow reuse and "import" of subsystems.
- The term *naked* OOA refers to a class model with no state or functional activity intended to be used as the schema for multiple projects.
 - A meta-model is most useful as a naked class model.

Statistics

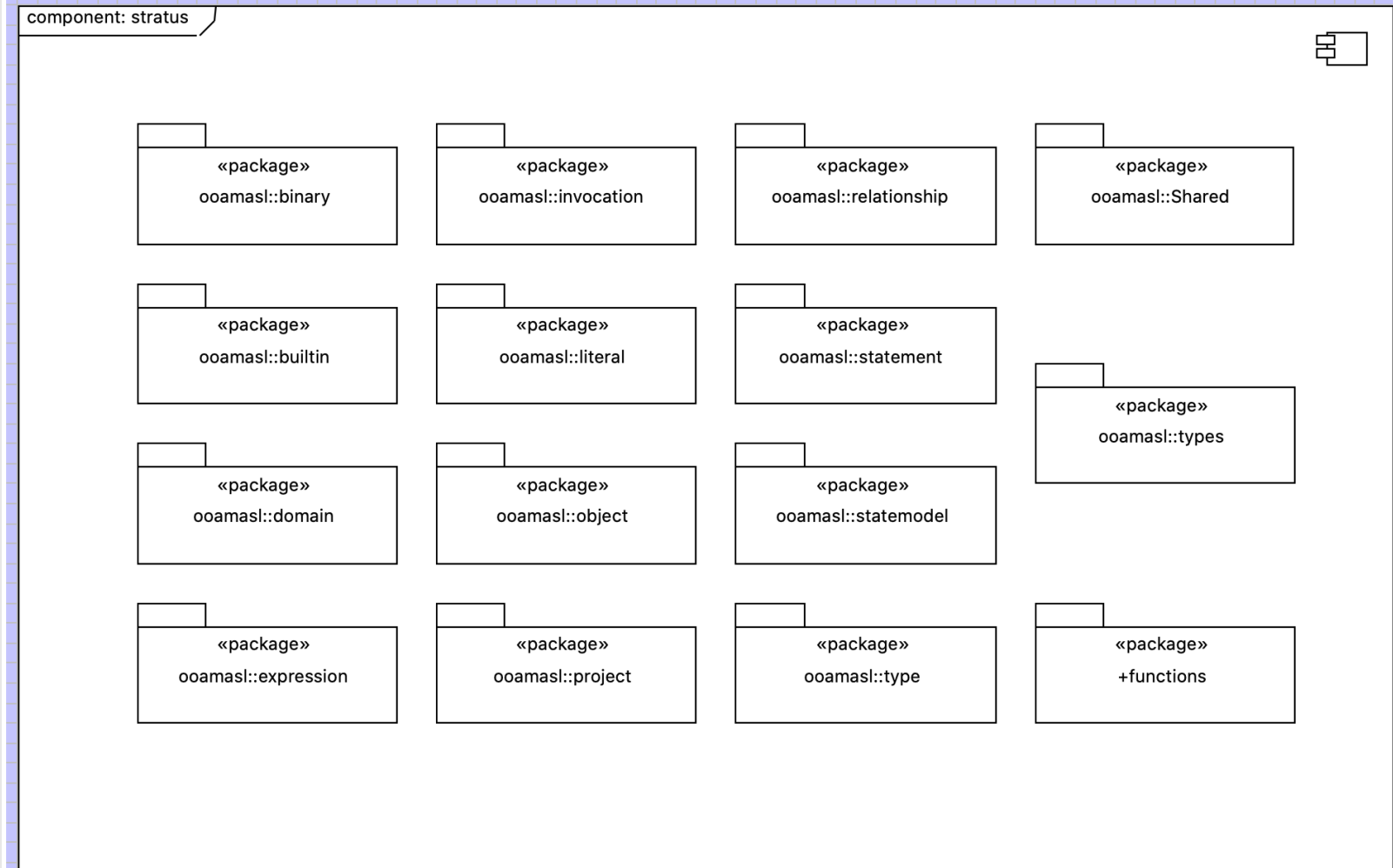
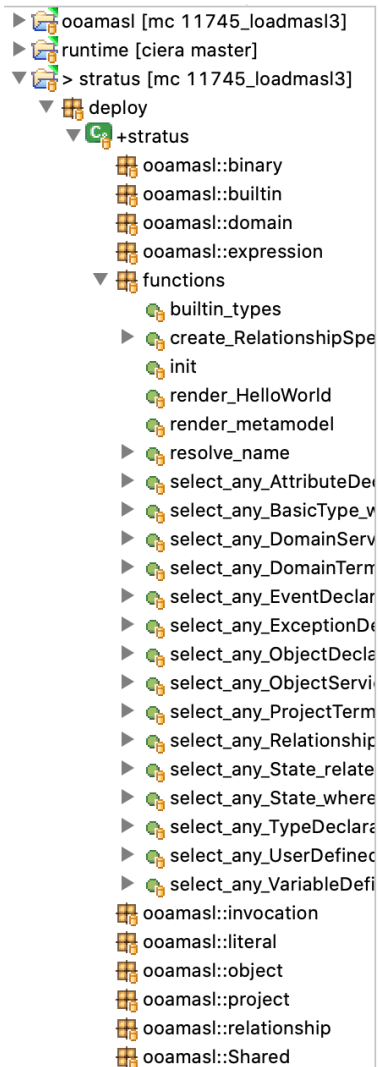
Statistics

subsystems	12
classes	176
attributes	134
relationships	234
production rules	166

MASL loader

- deployment model
 - imported OOA of MASL subsystem packages
 - parser, loader, model API functions

OOA of MASL: Stratus deployment



MASL loader

- Show functions
- Show parser interface (LOAD).
- Ciera build
- Ciera generated code
- Load ALU (and other test models).
- Load stratus.
- Generate abstract classes with stratus of stratus.
- Generate inventory report of major elements sorted in some order.

MASL loader: functions

- init
- select_any_Object_where_name
- resolve_name
- create_RelationshipSpecification

MASL loader: cont...

- Show parser interface (LOAD).
- Ciera build
- Ciera generated code
- Load ALU (and other test models).
 - masl_round_trip (load all test models)
- Load stratus.
 - Export stratus as MASL.
 - Load exported MASL.
- Generate abstract classes with stratus of stratus.
- Generate inventory report of major elements sorted in some order.

next steps

- review and further test
- template and rendering HOWTO
- cloud template integration
- Java translation of MASL activities