Reactive Resources Stream Pipeline

Main Classes:

Resource Monad : Resource<ResourceOccurrence>;

ContentType (Data / Transforms)

* onOccurrence transform
* getOccurrences(S, P, O) transform
* getOccurringContexts(S, P, O) transform
* fromRepresentation(Representation)
* toRepresentation(ContentType)
* ContentType(s) (modelType/encoding):
  + ID/reference-tmrm
  + IDOccurrence/reference-tmrm
  + Subject/reference-tmrm
  + Predicate/reference-tmrm
  + Object/reference-tmrm
  + SubjectKind/reference-tmrm
  + PredicateKind/reference-tmrm
  + ObjectKind/reference-tmrm
  + Statement/reference-tmrm
  + Graph/reference-tmrm
  + Model/reference-tmrm

TODO: Embed in ContentType / URN Resource instance type and context. Example: Model/FCA/reference-tmrm, urn:graph:subjectKind1.

Representation : ContentType instance

* ContentType
* Encoded State

ResourceOccurrence

* Representation
* onOccurrence(ResourceOccurrence)
* getOccurrences(S, P, O)
* getOccurringContexts(S, P, O)
* getAttributes() : Attributes (by means of occurrences / schema)
  + getAttribute(Attribute)
  + setAttribute(Attribute, Value)
* Hierarchies (TODO)

ResourceOccurrence(s)

* ID (URN : String, PrimeID : long)
* IDOccurrence : ID (occurringContext : IDOccurrence, embedding : long)
* SPO : IDOccurrence (Occurrence)
  + Subject : IDOccurrence
  + Predicate : IDOccurrence
  + Object : IDOccurrence
* Kind<Player, Attribute, Value> (Role / Type)
  + SubjectKind : Subject implements Kind<Subject, Predicate, Object>
  + PredicateKind : Predicate implements Kind<Predicate, Subject, Object>
  + ObjectKind : Object implements Kind<Object, Predicate, Subject>
* Statement (Data, SPOs Occurrences) : SPO
* Statement (Kinds, Kinds Occurrences) : SPO
  + Ex. (SK1, AttrX, ValY)
* Graph (Statements Occurrences given their SPOs / Kinds contexts)
* Model (Graph Occurrences)

ResourceOccurrence(s) Activation (ContentType handled, Resource Monad bound):

ResourceOccurrence::onOccurrence(ResourceOccurrence) Events:

ID::onOccurrence(IDOccurrence)

IDOccurrence::onOccurrence(SPO / Kinds)

SPO / Kinds::onOccurrence(Statement)

Statement::onOccurrence(Graph)

Graph::onOccurrence(Model)

Model::onOccurrence

ResourceOccurrence::getOccurrences(S, P, O) : S, P, O filter /criteria / matching.

Leverages CPPE / RCV / FCA / Kinds / Alignment schema / instances inference / filter / query / traversal.

Model::getOccurrences(S, P, O)

Graph::getOccurrences(S, P, O) : Models

Statement::getOccurrences(S, P, O) : Graphs

SPO / Kinds::getOccurrences(S, P, O) : Statements

IDOccurrence::getOccurrences(S, P, O) : SPO / Kinds

ID::getOccurrences(S, P, O) : IDOccurrence

ResourceOccurrence::getOccurringContexts(S, P, O): S, P, O filter criteria / matching.

Leverages CPPE / RCV / FCA / Kinds / Alignment schema / instances inference / filter / query / traversal.

Model::getOccurringContexts(S, P, O) : Graphs

Graph::getOccurringContexts(S, P, O) : Statements

Statement::getOccurringContexts(S, P, O) : SPO / Kinds

SPO / Kinds::getOccurringContexts(S, P, O) : IDOccurrence

IDOccurrence::getOccurringContexts(S, P, O) : ID

ID::getOccurringContexts(S, P, O) : URN

Runtime:

Events: Model Messages.

Main Event Loop:

Naming, Registry, Index stream nodes Model Events Topic consumers / producers. Matches for Models ContentType(s).

Topic streaming:

Stream nodes consume Model Events and publish augmented Model Event Context back to the stream for further augmentation. Updates augmented Model ContentType.

Resource Activation: each stream node unfolds consumed Model Event and invokes occurrences events, traversing occurrences / occurring contexts getters. Node augmentation logic in Resources Representations ContentType(s) transforms.

Datasource node: Produces Models Events published to the topic and listens for Model Events for syncing back backends state.

Producer node: consumes Model Events, publishes Activation API from Models and produces API interactions Model Events.

* Augmented Model in Events Context
* Naming Node: Resource Factory. URN Crafting / Matching. Aggregation (type / state / order inference).
* Augmentation Node Model Context ResourceOccurrence events:
* ID::onOccurrence
* IDOccurrence::onOccurrence
* SPO / Kinds::onOccurrence
* Statement::onOccurrence
* Graph::onOccurrence
* Model::onOccurrence
* Registry Node: Resource Repository. URNs Resolution / CRUD. Alignment (equivalence / upper matching, link prediction).
* Augmentation Node Model Context ResourceOccurrence events:
* ID::onOccurrence
* IDOccurrence::onOccurrence
* SPO / Kinds::onOccurrence
* Statement::onOccurrence
* Graph::onOccurrence
* Model::onOccurrence
* Index Node: Resource Contents URNs Resolution (inferences, transforms). Activation (possible verbs / state changes / transforms).
* Augmentation Node Model Context ResourceOccurrence events:
* ID::onOccurrence
* IDOccurrence::onOccurrence
* SPO / Kinds::onOccurrence
* Statement::onOccurrence
* Graph::onOccurrence
* Model::onOccurrence