DOM OntResource: Models Meta Meta Model. Reference Model. Runtime / dataflow APIs.

Parsing: AST Object URL monad layers hierarchy. Parser combinators. Monad Zippers (tree encoding). Order. Flows (streams). Parser combinators.

OntResource:

Signatures: observer (topics): ObjectID events stream. URL events stream. Role events stream. Layers hierarchy events set (reified: Resource, Statement, Kind, Relation events). Filter. Template / abstract methods. Core OntResource API behaviors. Dataflow.

Signatures: observer (topics): ObjectID events stream. URL events stream. Role events stream. Layers hierarchy events set (reified: Resource, Statement, Kind, Relation events). Filter. Template / abstract methods. Core OntResource API behaviors. Dataflow.

OntResource abstract APIs:

Abstract: Meta Model objects occurrences (aggregated by Object URL and Roles):

OntResource::ObjectID;

OntResource::Occurrences<URL, Role> : OccurrenceID;

OntResource represents aligned Objects / Resources of which the same Object (ObjectID) occurs in different Roles and (possibly) diferent URLs. Matching and alignments by reifying layers into abstract resources / OntResources.

Meta Model(s) APIs (Augmentation, Dataflow, Dimensional) reflects reference model events reacting to and producing signature matching messages on Meta Model(s).

OntResource's Roles are the DOM (Dynamic Object Model) types of the Meta Model(s) layers Occurrences / Objects Contexts. FCA Lattice contexts occurrences metadata. DOM APIs (reifying hierarchy layers, abstract templates):

Role: occurrence / object in CSPO slots. Denotes resource types in positions in statements (i.e.: Kind in Relation).

(Instance, Class, Metaclass, Occurrence);

(Context, Occurrence, Attribute, Value);

Meta Model(s). Augmentation APIs hierarchy:

Object<URL> : OntResource;

Abstract. OntResource aligned URL / template method design pattern.

Object<URL>;

Abstract. ObjectID.

Occurrence : Object, Object<URL> : OntResource;

Abstract ObjectID, OccurrenceID. Reify aggregated hierarchies.

Role : Occurrence, Occurrence, Object<URL>;

(Resource : Role, Role, Occurrence, Object<URL>);

(Statement : Resource, Resource, Role, Occurrence);

(Kind : Statement, Statement, Resource, Role);

(Relation / Model : Kind, Kind, Statement, Resource);

Class hierarchy APIs:

(Instance, Class, Metaclass, Occurrence);

(Context, Occurrence, Attribute, Value);

Meta Model(s). Dataflow APIs hierarchy:

(Augmentation, Mapping / Predicate, Transform / Template, Resource);

Meta Model(s). Dimensional APIs hierarchy:

(Dimension, Unit, Measure, Value);

Domain Models. Service APIs. Templates.

Dataflow, Dimensional and Domain Models declaratively stated in root Meta Model.

Messages. Encoding / I/O:

(Class, Instance, Member, Value);

Augmentation:

Domain (Templates)

Meta Model (Activation)

Meta Meta Model (Augmentations

Meta Model (Activation)

Domain (Templates)

Encodings:

(a (b (c (d, nil) : First / Rest binary tree.

(K: (K: inst, V: cls), V: (K: mcls, V: occur))

Object / Occurrence: Reify layers Object / Occurrence keys / values. Reference Model along with OntResource.

Message I/O:

(Class, Inst, Attr, Val);

Templates: I/O Dataflow (aggregated messages) declarative bindings. Aggregated message facade (Augmentation, render / apply layer entity schema / data). Forms / Flows.

Forms / Flows: HATEOAS HAL. MVC. REST. Meta Models DCI based protocol.

HATEOAS: Forms / Flows Operations / Dataflow Representation / State IO (CRUD) prototypes / templates. Dialog. Prompts. Gestures. Context: navigation state (i.e.: pick operation value prompt shows value type Forms). DDD DOM.