

- Monad Wrapper Role Types:
- Template : Context : Statement : Resource;
- URN : Resource (alignments). Primitives.
- Monads Wrapped Roles types / values. Transforms / Mappings. Objects.
- Resource Roles: Reified CSPO Resources types / values, Kinds types / values, Statements, Class, Instance, Occurrence, Occurring, Attribute, Context, Value, Role, Models Roles type / values getters (populated Mappings / Transforms).
- Resources Objects hierarchies / APIs: (Reified Mappings / Transforms): DTOs / Dynamic Functional, DTOs (hashmap) of CSPO roles getters / model domain browsing getters (Functional Transform parameterized Mappings). Flow context: referrers / keys types / values: address::city::street Aggregation / Map. Templates / Mappings: reified declarative Augmentation Data Flows.
- Ontology alignments: Data / Schema / Behavior Augmentations. Model / Schema / Upper / Domains: purposes / gestures (MVC / DCI Mappings / Transforms) layers. Example:
- Transforms: unaEmpresa::unEmpleado::unaPosicion::salary; Salary inferred by context and unaPosicion, unEmpleado, unaEmpresa.
- Class / Instance / Reified (occurrence / occurring) / Mappings / Transforms:
- Statement
- Kind
- ContextKind (SubjectKind / ObjectKind flow signature)
- Statement
- SubjectKind
- PredicateKind
- ObjectKind
- Subject
- Predicate
- Object
- Resource
- Context
- Template
- Class
- Instance
- Occurrence
- Value
- Occurring / Context (Statements / Kinds)
- Sets
- Graphs
- Roles (Metaclass, Class, Occurrence, Context, Role)
- Dimensional (Dimension, Measure, Unit, Value)
- Discrete (Relationship, Relation, Kind, Entity)
- MVC / DCI Mappings / Transforms. Example: Forms, Purpose, Gestures, Actors, Roles. Data / Schema / Behavior alignment.
- ESB: Endpoints, Features, Interfaces, Service Process Description / Discovery. Reactive Events Subscriptions. OSGi HATEOAS Endpoints "autowiring".
- BPM: Process, Steps, Flows, etc.
- Inferred / Reified / resolvable Data Flows.
- Graph Reified Grammars (upper). Terminal / Non Terminal. Rules / Productions. Mappings / Transform: browse grammar, rules, productions:

- (Rule, Context, lhs, rhs)
- Graph Embeddings: ML Backend Services (ML Predictions Augments Mappings / Transforms).
- Deep ML Embeddings. Data: classification, Schema: clustering, Behavior: regression.
- (...)
-
- Layers, input layer: Model Roles. Aggregation: Layer Roles shifting until full Layers Roles Statements. Layer: Augments Models.
-
- Core Statements Roles Resource Interleaved Model (Infer Types, PKs, FKs):
- Occurrence: (Class, Instance, Attribute, Value);
- Occurring: (Class, Instance, Occurrence, Role);
- Statement Transforms / Relations: order, equivalence, roles, etc.
- Resource of Resource Monad: Occurrence / Occurrences Quads CSPO Members Aggregation Transforms.
- $\text{Resource}(t : T) :: \text{contexts} :: \text{subjects} :: \text{predicates} :: \text{objects} : \text{Resource}(u : U)$
- Resource of URNs: Aggregated CSPOs Transforms of Occurrences / Occurrences Matchings.
- [Resource.of](#)(Resource / Class, Instance, Occurring Attribute / Occurring URN, Attribute Value / Occurring Role) :: contexts :: subjects :: predicates :: objects : Resources(CSPO / Resource / occurrence / occurring : URN).
- Resources of Resources: Occurrences / Occurring CSPOs Transforms Matching Wrapped / Wrappings / Transforms Resource Types Shapes Matching Templates. Data flow, apply transforms, order, lists.
- [Resource.of](#)(templ : Templ) :: contexts : Resources(c : Context).
-
- Layer Template Mappings:
- Template : Context : Statement : Resource Layer Roles Monads.
- Layers Quads Aggregation: rotating value role types from previous layer to next layer from V to C):
- Canonical Template Mapping Layers: Aggregation of Template Matching Models Layers until first layer Value is wrapped into final layer Context. Perform Layers Augmentations.
- Monads. Wraps Models Roles. Matching: Patterns wrapped Resource Roles values: placeholders, variables, wildcards, reified instances. Example: Template wraps previous layer Value as Context, Layer Context Value wrapped as Subject value (shifting).
- Core Roles (Wrappers):
- $\text{Resource} : \text{Monad}(x : \text{Resource});$
- $\text{Statement} : \text{Monad}(x : \text{Resource}) : \text{Resource}, \text{Quad DTO}.$
- $\text{Context} : \text{Monad}(x : \text{Resource}) : \text{Statement};$
- $\text{Template} : \text{Monad}(x : \text{Resource}) : \text{Context}.$
- $\text{Template} : \text{Context} : \text{Statement} : \text{Resource}$
- Input Layers: Core Statements Roles Resource Shapes: (interleaving graphs models) : Statements CSPO.
- Occurrence Input: (Class, Instance, Attribute, Value);
- Occurring Input: (Class, Instance, Occurrence, Role);
- Output Layer: (Template, Context, Statement, Resource); Input OPSC Resources wrapped in output Layer Roles.
- Sets, Graph, Roles: Services / Augmentations helper Models / Facades.

- Example:
- (Amantes, Pedro, Ama, María);
- (MariaAmadaPor, Amantes, Pedro, Ama);
- (Amar, MariaAmadaPor, Amantes, Pedro);
- (PedroAmaA, Amar, MariaAmadaPor, Amantes);
- (Amor, PedroAmaA, Amar, MariaAmadaPor);
-
- Augmentations:
- Model Augmentations:
- Alignment: Data Matching. Resources.
- Alignment::match
- Alignment::perform
- Alignment::greaterThan
- Alignment::equals
- Alignment::lessThan
- Activation: Schema Matching. Kinds.
- Activation::match
- Activation::perform
- Activation::superTypeOf
- Activation::sameTypeOf
- Activation::subTypeOf
- Aggregation: Behavior Matching. Contexts Flows.
- Aggregation::match
- Aggregation::perform
- Aggregation::beforeThan
- Aggregation::contains
- Aggregation::containedIn
- Aggregation::afterThan
- Domain Augmentations:
- Transforms Reified in Layers Contexts. Pattern Matching Template Layer resolved:
- Mapping::match
- Mapping::apply
- Mapping::Context
- Mapping::Subject
- Mapping::Predicate
- Mapping::Object
-
- Encode reified Template Mappings / Transforms. Patterns:
- Model / Domain Augmentations Mappings / Transforms:
- (Wrapper, Wrapped, Mapping, Transform);
- Next Layer step: match / apply Augmentations.
- Layers Template: Layer of CSPO Data Flow Patterns Resolution Resources : Reified (meta) Resources.
-
- Once Models Layers Matrix are built and populated / aggregated:
- Layers steps Augmentations: perform shifting and wrapping of aggregation values. Layer step Template Mapping: Layer::nextLayer : Layer, for each Layer, match / performs.
-

- Template, Mapping, Pattern
- (Template, Context, Statement, Resource);
- Resource :: occurrences :: roles :: contexts : Resource
-