* Semiotic Layer (Verbs Relationships).
* DOM Layer.
* FCA Layer (Augmentations).
* Sets Layer (Angular Encoding). Resource Monad / Transforms.
* Verbs: action (rel end: amante) / passion (rel end: amado) / state (rel: amor: ama / amado). Relation parts attributes. Semiotic Statements Source.
* DOM: (Class, Instance, Attribute, Value);
* Semiotics: CSPO Schema. DOM Statements Source.
* Occurrences: Objects.
* Concepts: Non Terminals.
* Signs: Terminals.
* Semantics / Pragmatics:
* (Context, Occurrence, Sign, Concept);
* (Context, Occurrence, Concept, Sign);
* Grammar:
* (Context, Concept, Occurrence, Sign);
* (Context, Concept, Sign, Occurrence);
* Syntax:
* (Context, Sign, Concept, Occurrence);
* (Context, Sign, Occurrence, Concept);
* FCA:
* Object: Subject.
* Context: Predicate.
* Attribute: Object.
* Augmentations (Sets Statements Source : FCA over DOM Statements):
* Schema Aggregation: Type / Relationships (Kinds / Roles) Inference. Clustering: Unsupervised Features Learning.
* Data Alignment: Type (Feature) Attributes Value Inference. Classification: (gender, salary range: scaling).
* Behavior Activation: Available Transforms (State Browsing) Inference. Regression: State (class attributes values in scenario: relationship flow).
* REST Browsing State Based Dialog Wizard. DOM / DCI / CDI / Augmentations (FCA). Structured Prompts / Responses (Statements Flow, Relationships / Roles).
* URNs: Semantic Identifiers. Encoding. DIDs (Distributed IDs). Angular Encoding.
* Conversational State Transfer (COST): Distributed (P2P).
* Node.js: JSON-LD. Functional Resources (Monads, run-at: request client / server peer. Context State available Functors / Transforms).
* Parsing: Transforms.
* Sets Encoding / Augmentations: Angular URNs. Angular Encoding:
* Class / Instance: Aggregated from FCA.
* Ontology Matching: Concept Lattice shape.
* SPOs: Points. X: Class, Y: Instance;
* Point (X Angle, Y Angle) : Angles;
* Aperture: degrees relative to axes / diagonal) : Angles;
* Angular Transforms:
* SPOs: Resource Monads. Points.
* Kinds: Monads Transforms. Distance.
* Translation : Point(Distance) : Point; Distance: Kinds SPOs Points; Translation: Subject Point Distance Points Product.
* Subject Points: SubjectKind Predicates / Objects.
* Reify Subject: (SubjectKind, Subject, Predicate, Object);