**Objectives:**

Rosetta Stone like engine enabling Ontology (Data, Schema / Information, Knowledge / Behavior) discovery, matching and integration.

Reactive Service Bus for pluggable integration of application and translation of gestures between domains business systems allowing workflows alignment and discovery of application systems behavior.

**Layers: RDF Quads Representation. Augmentation / Inference Matrix Models**

Patterns:

(Context, Occurrence, Attribute, Value);

(Dimension, Measure, Unit, Value);

(Dimension, Resource : SPO, Kind, Statement);

Sets Model Layers Structure:

(Dimension, Resource, Kind, Statement);

(Statement, Dimension, Resource, Kind);

(Kind, Statement, Dimension, Resource);

(Resource, Kind, Statement, Dimension);

(Dimension, Resource, Kind, Statement);

Dimension: U

Resource: SPO

Kind: SPO Intersections (pairs)

Subject Kind: P intersection O

Predicate Kind: S intersection O

Object Kind: S intersection P

Statement: SPO Intersection (of the three sets)

**Models: Sets, Individuals, Mappings**

Models which are instances of the Sets Layers Model Structure. Model Properties:

Metaclass, Class, Instance, Occurrence, Context, Role, Attribute, Value.

Functional. Mappings / Transforms. T-Box / A-Box. Sets, Groups, Categories: TBD.

Types Model:

Types (types in sets roles):

(Relation : Statement, Relationship : Kind, Role : SPO, Dimension : U);

(Dimension, Context, Class, Resource);

(Resource, Dimension, Context, Class);

(Class, Resource, Dimension, Context);

(Context : Statement, Class : Kind, Resource : SPO, Dimension : U);

Individuals Model:

Individuals (individuals / sets types instances):

(Context : Statement, Class : Kind, Resource : SPO, Dimension : U);

(Dimension, Context, Class, Resource);

(Resource, Dimension, Context, Class);

(Class, Resource, Dimension, Context);

(Context : Statement, Class : Kind, Resource : SPO, Dimension : U);

Mappings Model:

Mappings (type / individual relationships):

(Context : Dimension, Occurrence : Measure / SPO, Attribute : Unit / Kind, Value : Value / Statement);

Models metadata, properties and upper alignments / augmentations relationships Model data.

(Value, Context, Occurrence, Attribute);

(Attribute, Value, Context, Occurrence);

(Occurrence, Attribute, Value, Context);

(Context, Occurrence, Attribute, Value);

**Layers: Augmentations / Inference**

Activation: Classification (Context types Occurrences Attributes).

Activation: Which Attributes has Context Occurrence (according to its Kind in Context / Role) in this Occurrence.

Alignment: Regression (Context types Occurrences Attributes Values).

Alignment: Context Occurrence Attributes Values (according to its Kind in Context / Role).

Aggregation: Clustering (Context types Occurrences).

Aggregation: Context type instance aggregates type instance child Occurrences (parent Context type instances) matching grouping criteria (Encoding).

Augmentations:

(Context, Occurrence) : Value;

Activation:

(Statement, Resource) : Kind;

Alignment:

(Kind, Statement) : Resource;

Aggregation:

(Resource, Kind) : Statement;

**Model Semantics:**

Data: Individuals. Mappings. Data Occurrences Aggregation.

Data: Individuals Model.

(Dimension, Context, Class, Resource);

Mappings (type / individual relationships):

(Context : Dimension, Occurrence : Measure / SPO, Attribute : Unit / Kind, Value : Value / Statement);

Information: Types. Mappings. Type Occurrences Attributes. Activation.

Information: Types Model. Schema.

(Dimension, Relation, Relationship, Role);

Mappings (type / individual relationships):

(Context : Dimension, Occurrence : Measure / SPO, Attribute : Unit / Kind, Value : Value / Statement);

Knowledge: Individuals / Types Mappings (Attributes) Values. Alignment.

Knowledge: Behaviors.

Mappings (type / individual relationships):

(Context : Dimension, Occurrence : Measure / SPO, Attribute : Unit / Kind, Value : Value / Statement);

**Ontology Matching: Relations / Relationships**

Entity Relationship instance asserted as a reified concept with its type and attributes or as a series of triple statements which describes the given Entity Relationship instance via individual assertions. Bidirectional translation.

aPerson loves anotherPerson.

Person loverOf Person.

loverOf predicate: Kind of aPerson. Domain / Range. Dataflow (Functional Augmentations).

Loving: loverOf Kind.

aLoving: loves Kind.

TBD: Relationship / Relation

Reify Kinds as SPOs : Types Model

Reify Statements as / Kinds / SPOs : Mappings Model

Augmentations (Aggregation).

**Ontology Matching: Dimensional Alignments (Mappings):**

Explain Layer Context, Occurrence, Attribute, Value Pattern for Models SPO Statements functional mappings expansion:

(Context, Occurrence, Attribute, Value);

For a given CSPO Quad:

(C, S, P, O);

Expansion:

(C, P, S, O);

(C, O, P, S);

TBD.

Mappings (set / individual relationships):

(Context : Dimension, Occurrence : Measure, Attribute : Unit, Value : Value);

Order. Comparison. Relations. Upper Ontology assertions. Augmentations. TBD.

Relation / Relationship: Tabular / OGM (Object Graph Mapper):

I/O: (Class, ClassID, Attribute, Value);

Class: Table / Object Type.

ClassID: PK / Object ID.

Attribute: Column / Member.

Value: Cell / Field Value.

Subject Kind: Relation / Domain.

Predicate Kind: Relationship.

Object Kind: Mapping / Range.

Dataflow: Reactive Functional Augmentation / Integration APIs.

Indices: Apply functional mappings expansion.

**Functional API: Monads / Transforms**

Resource / Layer?

Context / Occurrence / Mapping?

Mapping: Selector Monad. Matching Resource / Role set?

Context / Occurrence Monads wrapping Layers Hierarchy Contexts.

Entity Alignment / Matching resolution via Functional Augmentations: Agggregations / Activation / Alignments (upper / dimensional matchings). Versioned graph: stateless / functional. Mappings assertions matching.

APIs: Augmentations, Query, Traversal, Matching, Transforms. Functional APIs Query / Browse / Traversal / Transforms examples. Encoding / Matching.

**Encoding: Functional Mappings**

Masks: Predicates of Set memberships. Functional Mappings. ID encoded state / transforms. Models merge. Ontology Matching. Mappings Model: Types / Instances Models merge (upper) Augmentations.

Mappings / Functional Encoding: Upper Dimensional Matchings / Augmentations. Mappings Model masks matchings reflects / leads to Types / Individuals Models Augmentations / Assertions.

Mappings / Functional Encoding: Relation Statements / Relationship views / matchings examples.

Mappings Upper Alignments examples (dates, marital status, hiring). Relation Relationship statements order / context properties (Dimensional Alignments).

IDs: metaclass, class, instance, context, role, occurrence, previous, next ID roles relations for Model Set Contexts.

Augmentations / Transforms: Model / Domains functional mappings. Order. Dimensions. Axes. Flows. Hierarchies. Inference / Population.

Levels: Augmentations. Mappings.

Levels: Resource, Kind, Statement.

Levels: Reify Statement as Kind, Kind as Resource, Resource as Statement.

Levels: Reify Resource as Kind, Kind as Statement, Statement as Resource.

**TODO Items:**

* Dimensions Encoding: Given Dimensional Contexts (CSPO Models set layouts) having four dimensional sets (Types Model, Individuals Model, Mappings Model, State Model) each representing (nested) CSPO inputs / parts of a recursively aggregated CSPO layout (i.e. aggregated layout Context is Mappings Model, Subject is State Model, etc.) having this setting (Models types / layers class / instance IDs) reified in this fifth "Focus" Model which represents a "snapshot" of current state and available transitions (Focus shifts).
* Models: CSPO Layers (matrix) layout.
* Focus Mapping Model. Axes (X / Y: Model instances matrices, cycles), intersection (Z: Model instance matrix):
* Model patterns:
* (Dimension, Unit, Measure, Value);
* (Context, Occurrence, Attribute, Value);
* Context / Dimension / Context:
* Occurrence / Unit / Subject:
* Attribute / Measure / Predicate:
* Value / Value / Object:
* X Model: Context / Schema / Information / Relationships
* Y Model: Data / Relations
* Z Model: Interaction / Context instance Data state calculated intersection. Behavior
* Upper Y / Lower Y: Previous / next data state.
* Left X / Right X: Previous / next context state.
* Augmentations calculate current, previous, next Model states.
* ImplementationItems & drafts documents topics.
* Diagrams (TO DO):

