Workflows (Domain Goals) general purpose ontology matching integration framework.

Components:

Bus: Reference Model encoded Messages. Component Monad of Node type: Node type handler of Message I/O translation.

Component.of(node);

bus.connect(component);

(for bus.components)

Component next = component.flatMap(NodeType::consumeCurrentMessage);

(next until end of depth / list of nodes)

Reference Model (Component Node Message Adapters). Component Monads of Component Nodes: Functional events (bus) dataflow (selector signatures)

I/O. Persistence. Events. DIDs Components Nodes

Sets Component Node

Layers Augmentation: Aggregation (layers), Alignment (ontology), Activation (dataflow) Component Node

Layers Quads Component Node

Triple Store Component Node

FCA (Monads AST Builder. Updates Quads Productions) Component Node

Layers Monads / Parser Monads (Messages : Rules / Productions). Functional events dataflow (selector signatures : Layer instance Activation) Component Node

Layer production = Layer.of(resource);

Layer rule = production.flatMap(ResourceType::matchRule);

Forms / Flows (Grammar / Protocol Builder. Prompts) Component Node

Augmented Resources Contexts / Interactions Services Component Node

OGM / Client Drivers Services Component Node

Services / Mappings:

Upper Ontology. Grammars. Primitives. Ontology Matching

DOM Hierarchy:

Resource<OntResource[]>;

Kind<Resource[]>;

Statement<Kind[]>;

Relation<Statement[]>;

Entity<Relation[]>;

Relationship<Entity[]>;

Flow<Relationship<Entity[]>;

Domain<Flow[]>;

Meta Model (Layers / DOM):

CellValue

ColumnField

ID : occurrence (PK)

Context : instance (table)

Role : metaclass (CSPO)

Resource : class. Monad Value (instance)

Kind : selector / transform (Functor mapping). Monad Value Type (metaclass / role)

Statement (context)

Relation : Kind Grammar (Productions). Monad Instance (occurrence)

Entity : Kind Grammar (Rules). Monad Type (class)

Relationship

Flow

Domain

Ontology Matching:

Matching: Resource occurs as context / occurrence / atribute / value or class / occurrence / context / metaclass / instance in equivalent occurrence contexts (kinds / order / shapes / type hierarchies).

Meta Model encodes mappings for equivalence / relations hierarchies for entities instance occurrences in roles in contexts for concepts recursively till upper onto / primitive terms / relations.