Distributed Integration and Consistency for Knowledge Semantic Interoperability

Outline (drafts)

Purpose (integration, augmentation, extension)

Description

Analysis (mission)

Design (vision)

Deployment Use Case: Goals App Solution

Features

Connectors

Clients

Adapters

Declarative (model / domain driven):

Domains alignment / matching: aggregations / learning (data, schema, behavior information / knowledge augmentation)

Sources augmentation (reactive synchronisation)

Shared data consistency: Governance over (distributed) MDM, BPM, Business Rules. Distributed inference repository

Problem spaces facades: inter domain / inter apps unified dashboards. Goals gestures / dialogs (Form / Flows) translation.

Domains extension (Protocols / APIs):

Microservices Hub API. Declarative (model described) services APIs / Protocols (Activation integrable)

Protocol: hierarchical contexts selection / edition of role values (reified possible contexts. Services Activation context state protocol (Activation grammars)

Distributed Persistence:

Event sourcing / Saga pattern: distributed addressables queue. Reconciliation. Trust. Transactions: Request IDs.

Addressable Interactions. CQRS. Reified interactions / gestures. Logs. Reified (resource) Message Activation (dataflow). Forms / Flows

Implementation:

Model: RDF CSPO Quads store. RDFS / OWL inference.

Primitives / meta / upper resources. Grammars: patterns / signatures. Context Kinds (Message / Augmentation)

Kinds: hierarchies. Semiotic / Sets DM / RM encoding.

Layers. Streams. Events. Augmentation.

Resource Monad (REST). Addressable Interface. DOM (roles, rels, members, etc.) Functors.

Form / Flow Facades

Message Functor. LHS: domain: source / RHS: range: dest. Augmentation / Event pattern declaration. Mapping encoding. To do.

Model Context hierarchy (DOM). Layers CSPOs classes DOM: type / context, occurrence, attribute, value, prev, next, parent, children, etc. members. Functors.

Message: Augmentation (event). declaration.

Core: primitive / meta / upper resources. Reified layers contexts. Order (dataflow) encoding.

Core: Augmentation. Aggregation (pragmatics, roles). Alignment (semantics, rels). Activation (levels: schema, grammar, syntax of roles / rels).

Core Contexts. Layers. Augmentations.

Domains (Context DOM instances).

Dataflow: Events I/O. Message augmentations. Connector / Client / Adapter.

Message Mapping Augmentation perform. XML / XSL over RDF (cons) lists. DOM members resolution (functors) / embedding.

Aggregation / Matching (Model / Entity, Behavior / Measure). OntResource embedding (contextual functor).

Events I/O "fans in/out" model layers performing Augmentation Activation.

Graph key / value (properties) encoding. Map Reduce event flow tasks.

Deployment:

Container: Forms / Flow DCI / MVC Facade APIs. Resource (REST) Monad and Context Functors (reified Message resources). Protocol (Connector / Client Adapters).

Low level Resource API.

DOM / OGM rendering / activation APIs.

Model Context DOM members (layers): order (prev, next, parents, children) in axis. Metaclass, class, instance, occurrence (roles / rels in contexts). Functors / DOM events API: request reified Message resource in "possible" (domain, mapping, range) context. Bindings (subscriptions).