* Models
  + Reference Model: FCA / Hashing. DIDs: URNs Resources. Events Sourcing.
  + URNs: URNFactor (ContextFactor, SubjectFactor, PredicateFactor, ObjectFactor);
  + Factors: URNs as CSPO Roles occurrence values product URNFactor (primes / bitstring). FCA Contexts Objects (Statements) / Attributes (URNs Roles Factors: bitstring / primes values hash lattice).
  + Augmentations: Matching / Inferencing. Upper / Matching URNs Role values Factors of Statements / URNs Role Factors.
  + Statement Aggregation Inference: Statement URN Roles Factors of URNs CSPO Factors values. Inferences embedded in Resources URNs Factors product of Statement URN Roles values.
* FCA Contexts:
* Statement (objects) CSPO Attributes:
* (URNFactor (ContextFactor x SubjectKindFactors : class, SubjectFactors : instance, PredicateFactors : attribute, ObjectFactors : value))
* Reified Statements (attributes), SPO Factors x Kinds (objects):
* (StatementFactors : SPO Factors x Provenance (SubjectFactors x SubjectKind, PredicateFactors x PredicateKind, ObjectFactors x ObjectKind))
* Provenance: Entailment. Provenance x reified SPO StatementFactors / source URNs. Statement entails / entailed by Factors product relations: transitive, reflexive, symmetrical (cause / effect, etc.).
* Alignment / Matching: Factors of matching URNs / Statements aggregated by product.
* Graph Statements (materialize / align) URNFactors attributes. Objects: CSPO Factor:
* (URNFactor, URNFactor, URNFactor, URNFactor)
* SubjectKind attributes. Objects: aggregated Subject Factors:
* (Context : super SKs factors, SubjectKind : matching factors / same Predicates Subject factors, Predicate, Object)
* PredicateKind attributes. Objects: aggregated Predicate Factors.
* (Context : super PKs factors, Subject, PredicateKind : matching factors / same Object Predicate factors, Object)
* ObjectKind attributes. Objects: aggregated Object Factors.
* (Context : super OKs factors, Subject, Predicate, Object : matching factors / same Subject Object factors)
* Order / Flow Mappings attributes. Objects: aggregated State Factors.
* (Context, SubjectKind, PredicateKind, ObjectKind)
* Context: State factor (Kinds) previous / next Kinds Mapping Layout. SPO Kinds State (Kinds factors). Super / Sub Context Alignment. Kinds "joins": materialize / align order / flow Statements. Kinds flow: Statement Kinds / SPO States (Contexts). Order: upper / super / sub hierarchies (Kinds Contexts / factors). Joins matching Kinds factors (flows). Provenance state flows.
  + Sets. Data Aggregation.
  + ResourceURNs (Occurrences). Contexts Activation.
  + DOM (Dynamic Object Model Object Graph Mapper): Functional Dataflow. OpenRDF Sesame Elmo / Alibaba. Interactions Alignments.
* Augmentations
  + Aggregation.
  + Activation.
  + Alignment.
* Ontology Matching: FCA / Hashing
  + Data. CSPO.
  + Schema. Kinds.
  + Behavior. Contexts.
* Inferences: FCA / Hashing
  + Data Ontology Matching.
  + Contexts Ontology Matching.
  + Interactions Ontology Matching.
* DDD: Forms Dialogs Protocol: Suggestions (infer Purposes from Gestures: DCI Matching)
* Detectar mediante lo que yo llamo "Gestures" el Propósito de una Interacción de un usuario (persona o servicio: flujos de navegación) en un Contexto dado, según los Datos del "diálogo" en un protocolo que permita inferir y facilitar la intención o el objetivo de la Interacción mediante "Suggestions".
* Detect by means of user "Gestures" (person or service browsing flows) the Purpose of an Interaction, in a given Context, following Data of a of a "dialog" in a protocol such that the Interaction intention or objectives may be inferred and guided by means of "Suggestions".