Subject Information Mediation for Integrated Access to Heterogeneous Sources

San Diego Supercomputer Center
"Computational Science Seminar Series"
June 13, 2002

L. A. Kalinichenko

Institute of Informatics Problems
Russian Academy of Science

Laboratory for compositional information systems development

Various forms of compositions are studied, e.g. :

- Interoperable compositions of pre-existing components for IS design;
- Compositions of heterogeneous information collections;
- Workflow compositions;
- Type compositions in database operations over object collections;
- Heterogeneous mediators compositions.

Web site of the group: http://www.ipi.ac.ru/synthesis/

Talk outline

- Subject Domain Mediation
- Mediators’ Projects: Brief Overview
- Query Planning Methods
- Infrastructure of the mediator aiming at semantic interoperability of collections
- Summary

Subject Domain Mediation

<u>Outline :</u>

- **Objectives of information integration**
- **The mediator's concept**
- **Mediator's classes**
- **Consolidation of a mediator**
- **Advantages of the subject domain mediation approach**

Web Search Engines

1 billion Web pages.

Search engines remain to be the main mechanism to access pages. Key words queries. Dozens of general purpose search engines, thousands of specialized engines (regional, thematic, corporal).

The following kinds of general purpose Web search engines can be distinguished:

- basic engines: AltaVista, HotBot, Infoseek, Lycos, WebCrawler, Yahoo, Rambler, Яndex, etc.
- portals: Skworm, Proteus, Instantseek, etc.
- metasearch engines: SavvySearch, Inference Find, ProFusion, etc.
- metasearch utilities: Copernic, BeeLine, SearchPad, etc.

"Metasearch" engines provide for requesting several search engines and composing combined response. It is assumed that such response more probably will contain relevant information.

Precision of search is very low (uncontrollable use of terms for indexing and search). This is unavoidable payment for simplicity of home pages "registration" for the whole Web.

What is the required level of information integration/dissemination

- Just putting information on the **Web** (creating a homepage, a Web site)

- Inserting a description of a resource into a suitable **Digital Library** (e.g, into NCSTRL, the Networked Computer Science Technical Report Library, a collection of institutional and archival CS research reports and papers)

- Using **subject gateways** for easier access to networked information resources in a defined subject area. Subject gateways work as intermediaries

- Applying a **community-oriented digital library** (a collection of documents built by a community of users which aims at observing or studying a phenomenon (e.g., in a context of a certain area)).

- Using heterogeneous **multidatabase systems**.

- Applying **subject mediators** to support representation and access to various subject domains. **Mediators should provide modelling facilities and methods for conversion of unorganized, nonsystematic population of collections registered by different collection providers into a well-structured set of sources supported by the integrated uniform specifications. Metainformation. Systematic registration of collections.**

What and for what is to be integrated

What kind of information is to be supported
structured, object, semi-structured, textual, multimedia

What kind of metainformation is needed
thesauri, classifiers, vocabularies, ontologies, schema definitions (data, objects, functions, workflows)

What to disseminate:
1. A document (paper) as a whole using additional document description
2. A document in XML
3. Content of a document

What to retrieve
1. To discover individual resources (Web pages, documents, papers)
2. To retrieve information relevant to a specific query contained in a collection of resources
3. To retrieve information as workflows, methods and/or data and use various compositions of those
4. To provide for interoperability of the information sources in process of problem solving:
   - technical level of interoperability
   - semantic interoperability

Digital repositories of knowledge

Digital repositories of knowledge in certain areas can be implemented, like: **Digital Earth, Digital Sky, Digital Bio, Digital Law, Digital Art, Digital Music**. Examples of **Microsoft TerraServer**, **Multi-Terabyte Astronomy Archives** are widely known.

An example: **DigiTerra (an Environmental Digital Library, Rutgers)** objective is to provide continuous land monitoring, fire detection, water and air quality testing, urban planning, as well as supporting research and instructional activities in related areas of science. Vast array of environmental data collected in DigiTerra should include images from a variety of space-borne satellites, ground data from continuous monitoring weather stations, maps, reports and data sets from federal, state and local government agencies, and serve diverse user community.

The Mediator Concept

The mediator architecture (Wiederhold, 1992) deals with the problem of integration of heterogeneous information. The sources are "heterogeneous" on many levels:

- data model and types of data used;
- the underlying data units (salaries could be stored on a per-hour or per-month basis);
- behavior of objects involved;
- the underlying concepts. A payroll database may not regard a retiree as an employee, while the benefits department does. Conversely, the payroll department may include consultants in the concept "employee" while the benefits department does not;
- the schema that the information may conform cannot be rigid in advance. Examples of "semi-structured" information include that found in XML documents, repositories used in the Human Genome Project, Lotus NOTES.

Mediator is to provide a uniform query interface to the multiple data sources, thereby freeing the user from having to locate the relevant sources, query each one in isolation, and combine manually the information from the different sources.

Mediation approaches

• integration information from **pre-selected sources** according to the **predefined information needs**. A procedural approach is known (TSIMMIS, Squirrel, WHIPS) to integrate information from sources through ad-hoc procedures. When information needs or sources change, a new mediator should be generated. This is known as **Global as View** (GAV) approach.

• integration information from **arbitrary sources** according to the predefined information needs. A declarative approach is known (Carnot, SIMS, Information Manifold, Infomaster). Mediators contain mechanisms to rewrite queries according to source descriptions. A rewritten query should be contained in the original query. This is known as **Local as View** (LAV) approach.

• combined LAV and GAV approaches (GLAV)

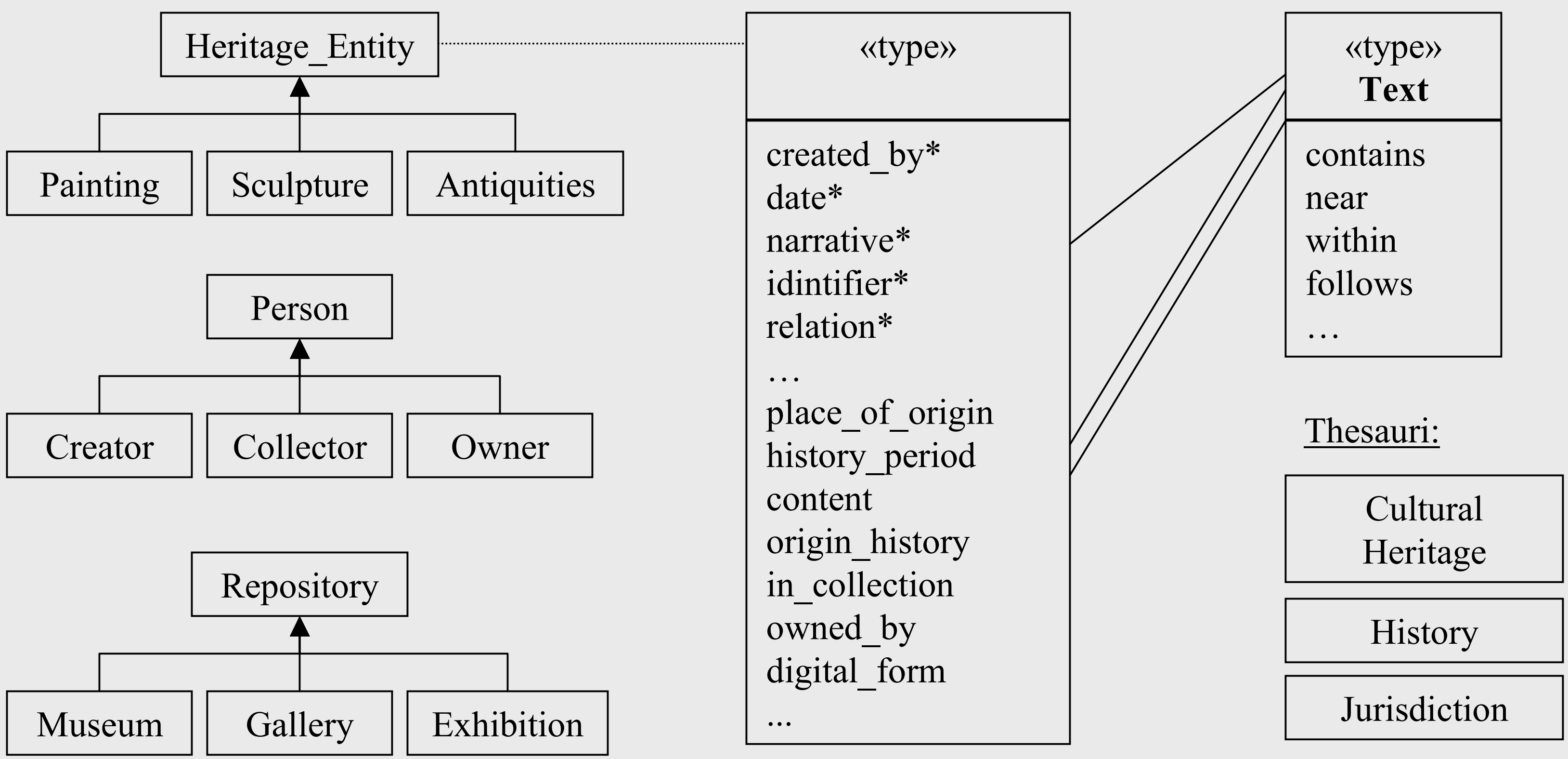
Mediator Definition as Subject Metainformation Consolidation

For the **mediator's scalability** two separate phases of the mediator's functioning are distinguished: **consolidation phase and operational phase.**

On the consolidation phase the efforts of the scientific community are focused on the mediator subject definition by declaring its metainformation. It is assumed that the top level researchers are involved in this process. The metainformation defined at the consolidation phase is assumed to be conservative for a certain period of time when it can only be extended. The well-known, representative collections of information in the subject domain are used during the process of metainformation definition. The metainformation created at the consolidation phase constitutes the federated level of the mediator.

During the operational phase arbitrary information collections can be registered at the mediator expressed in terms of the federated level. Process of the registration is autonomous and can be done by collection providers independently of each other. Users of the mediator know only the metainformation defining the mediator’s subject and formulate their queries in terms of the mediator’s subject. For a query the mediator decides what registered collections are relevant to the query.

Subject Mediator. Cultural Heritage Collections. Federated Level Metainformation



Subject Mediator. Cultural Heritage Collections. Collections Registration

Federated Level Metainformation

Local into Federated Level Mapping

CIMI Profile of z39.50

museum_object	creator_c
created_by* date_collected* description* object_id* relation* … content_general collection mrObject	nationality works

Louvre Museum Web Site

department	author
name description sections	name nationality works

Uffizi Museum Web Site

artist	canvas
name biography paint_list	title painter date history description to_image

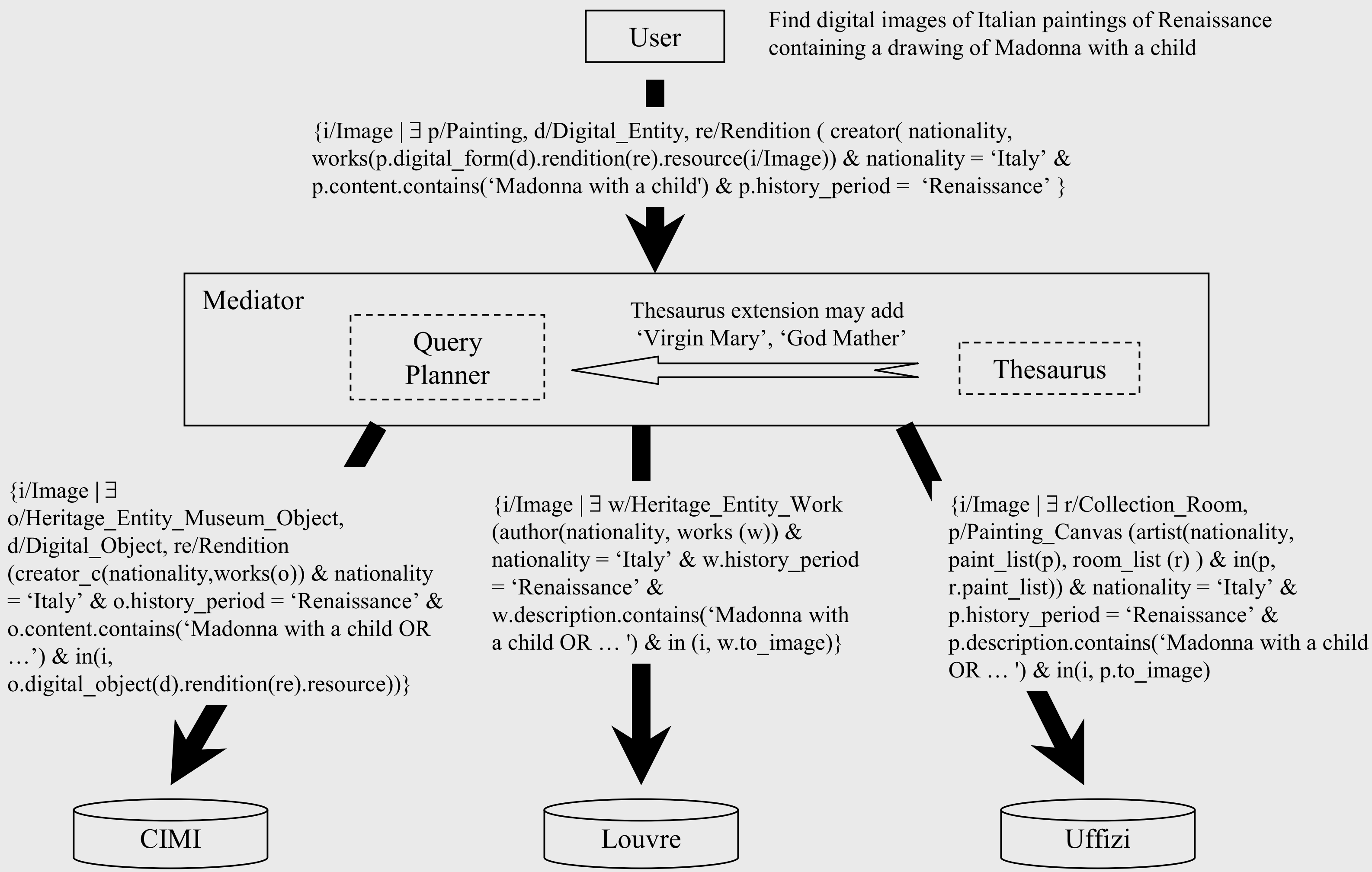
Local Views in Terms of Federated Classes

creator_c(c/Creator_Creator_Info [name, nationality, date_of_birth, date_of_death, works/{set_of:Heritage_Entity_Museum_Object}]) ⊆ creator(c[name, nationality, date_of_birth, date_of death, works])

author (a/Creator_Author[name/fname, nationality, works/{set_of:Heritage_Entity_Work}]) ⊆ creator(name, nationality, works (w)) & ∃ c,s (repository (c/Collection [contains(s/Section)]) & repository.name = 'Louvre' & in (w, s.contains))

artist(a/Creator_Artist[name, nationality, general_info/Text_Textual, works/{set_of: Painting_Canvas}]) ⊆ creator(a[name, nationality, general_info, works]) & repository (n/name, collection) & n = 'Uffizi' & ∃ col/Collection (¬ isempty (intersect (collection(col/ Collection).contains, works)))

Subject Mediator. Cultural Heritage Collections. Query Planning



Advantages of subject domain mediation

1. Subject mediation makes possible to reach **semantic integration of heterogeneous information** collections

2. **Users should know only subject definitions** that contain concepts, structures and methods as defined by the community

3. Information providers can **disseminate** their information for integration **independently** of each other and **at any time**. To disseminate they should **register their information at the subject mediator**. Users should not know anything about the registration activity.

4. Autonomous information collections **contexts, data model and languages used, implementation platforms** are absolutely **independent** on the mediator and its consolidated metainformation definitions

5. Querying the subject definitions, users have **integrated access to all information registered** at the mediators up to the moment of a query.

6. Mediators form **recursive structure**: each mediator can be registered at another mediator. Thus, **multiple subjects can be semantically integrated** defining mediators of the higher level.

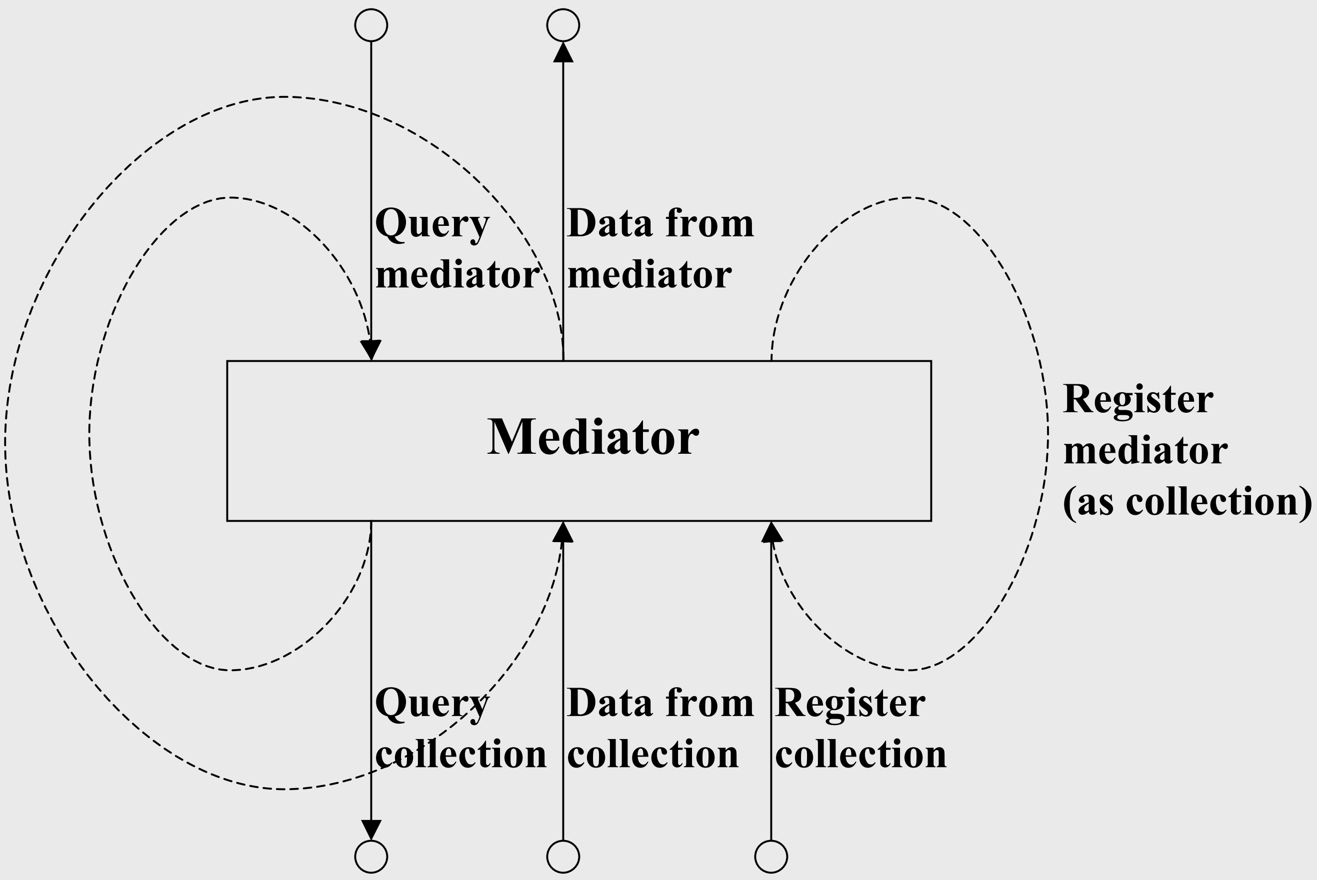
7. **Personalization** providing convenient views for specific groups of users can be formed **above the subject definitions**. This process is independent of the existing collection and their registration.

Disadvantages of subject mediation

1. Providing a subject definition requires that **a proper level of maturity and organization of application domain community (e.g., scientific community)** have to be reached (e.g., are the research and development groups in the area sufficiently **open, collaborative and motivated**). Subject consolidation is a collective, organized effort of the community.

2. **Process of registration** is not an easy one and **requires specific supporting tools**.

Mediator’s Recursion



Mediators' Projects: Brief Overview

<u>Outline :</u>

- **TSIMMIS (Stanford)**
- **Information Manifold (Univ. of Washington)**
- **GARLIC (IBM)**
- **InfoSleuth (MCC)**
- **XML as a middleware model**

Query Planning Methods for Mediators of Heterogeneous Information Sources

Outline :

- **Query Planning for LAV approach**
- **Query Containment Techniques**
- **Wrapper generation**

Query Containment Algorithms

Query containment has been considered for the purposes of query optimization, detecting independence of queries from database updates, rewriting queries using views, maintenance of integrity constraints, semantic data caching, etc.

- Containment mapping for conjunctive queries [1977]
- Uniform containment of Datalog programs [1988]
- Containment of Conjunctive queries with built-in subgoals [1989]
- Containment of conjunctive queries in Datalog program [1989]
- Uniform containment for Datalog$^{\neg}$ programs [1996]
- Containment for Queries with Complex Objects [1997]
- Boolean query containment [1997]
- Containment for Conjunctive Queries With Regular Expressions [1998]
- Query containment relative to views [1999]

Features supported by different query rewriting algorithms using views

1. Conjunctive queries
2. Source templates with binding patterns
3. Recursive queries
4. Views in description logic
5. Rewriting for semistructured data. Regular expressions rewriting, navigational plans
6. Boolean queries rewriting
7. Queries with union and aggregation
8. Type inferencing
9. Object fusion

Techniques:

1. Scalable technique
2. Inverse rules technique, etc.

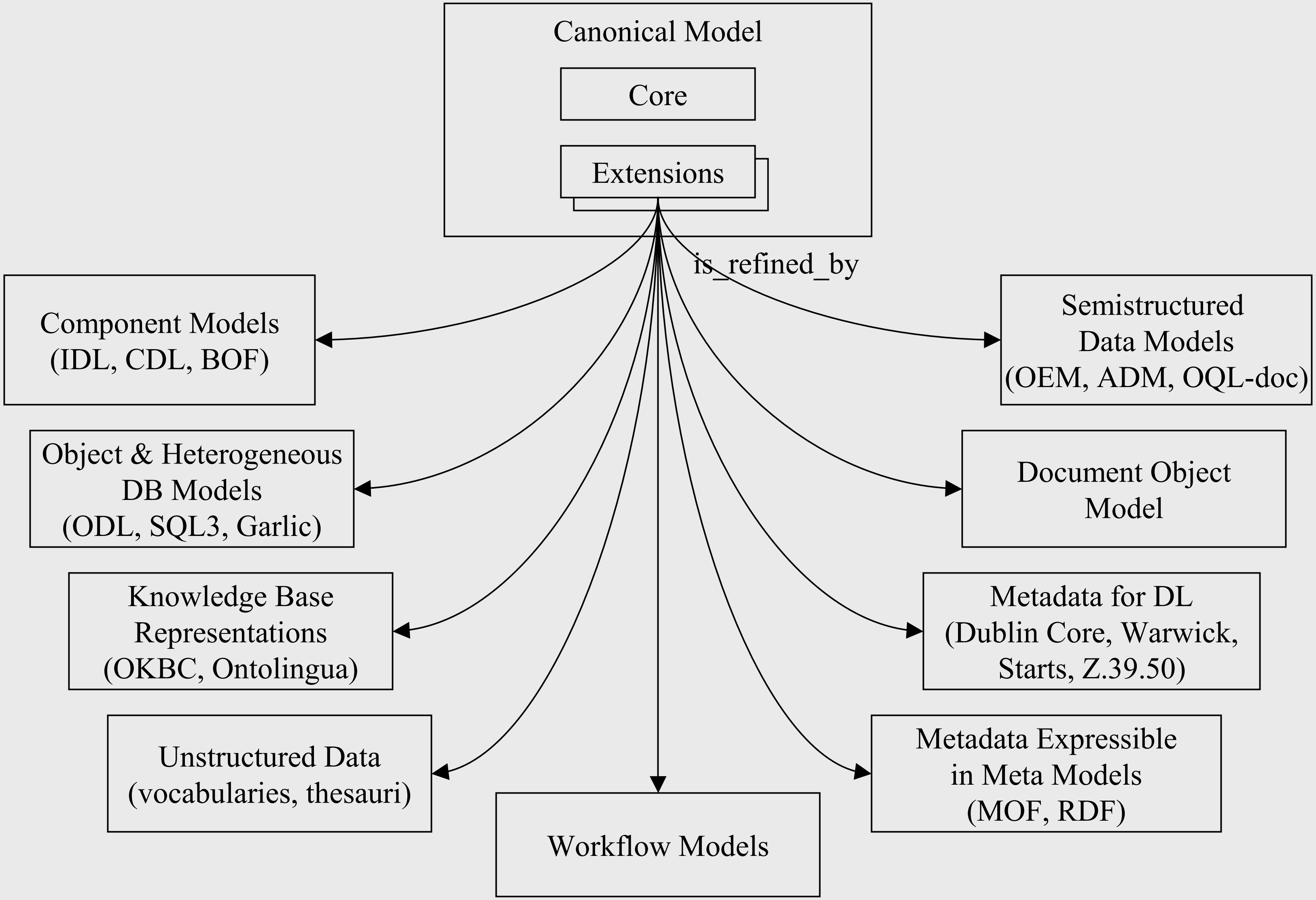
Infrastructure of the mediator aiming at semantic interoperability of collections

<u>Outline :</u>

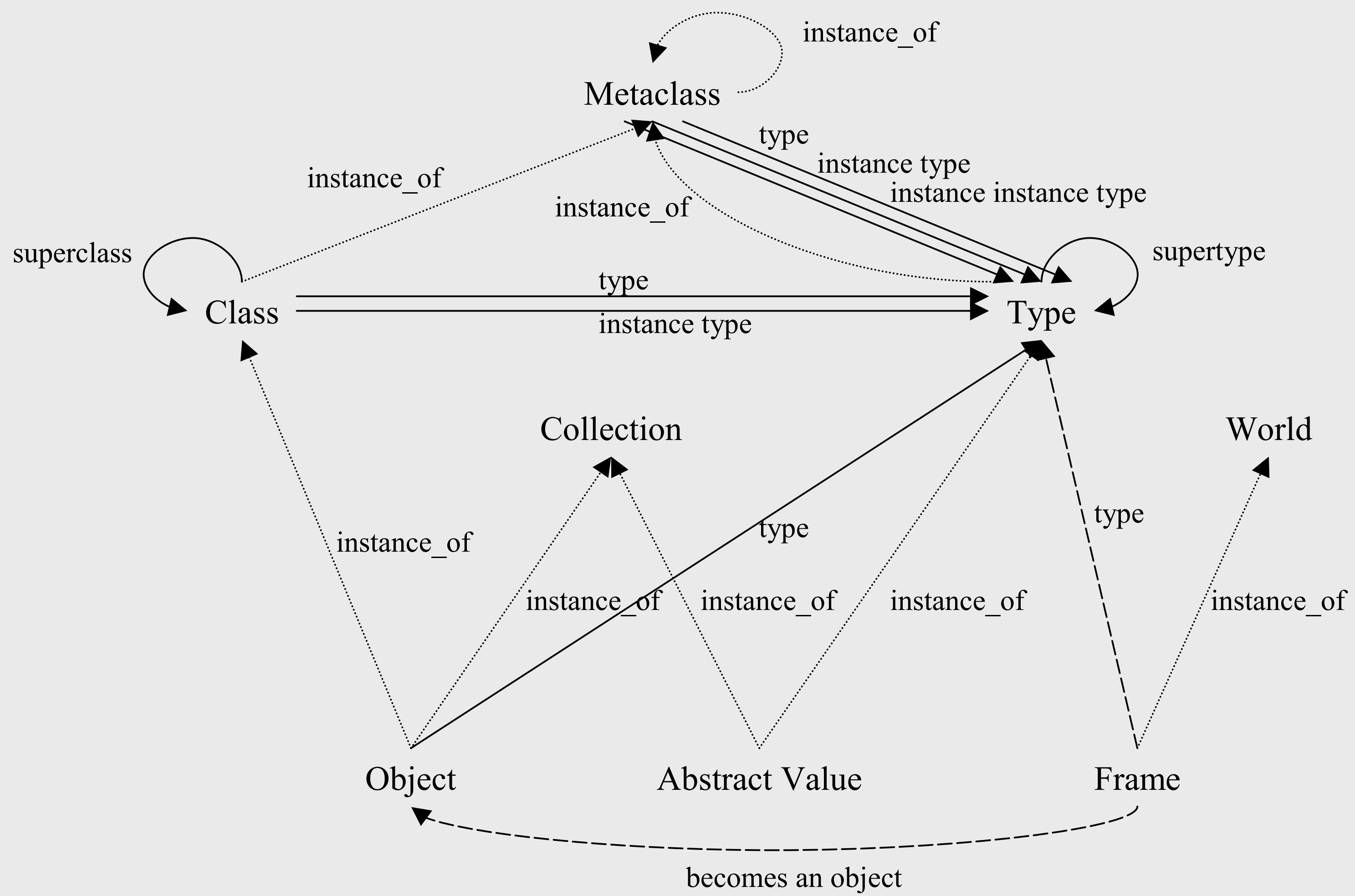
- **Canonical information model**
- **Infrastructure of the mediator**
- **Mediator’s metainformation**
- **Collection registration at a mediator as a process of compositional development**
- **Intermediator Protocol**
- **Extensible Ontological Modeling Framefork**
- **Query rewriting**

Canonical information model

Heterogeneous models absorbed by the canonical model



Canonical Model Entities



Canonical Information Model

A set of the canonical model facilities used for the uniform representation of the information resources includes the following:

• **Frame representation facilities**. Frames are treated as a special kind of abstract values introduced mostly for description of concepts, terminological and weakly-structured information. All specifications in canonical model have a form of frames that become a part of the metabase.

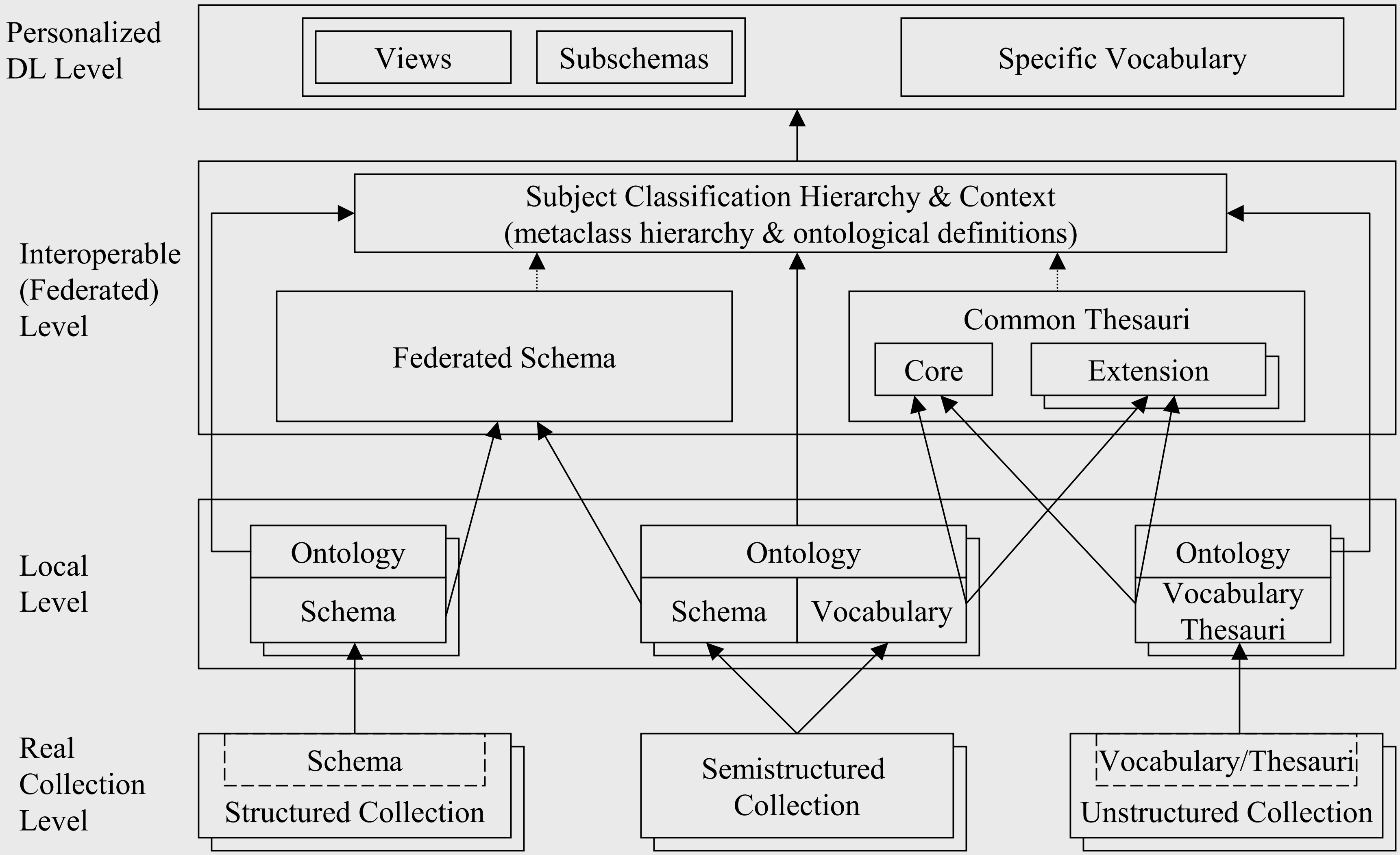
• **Unifying type system**. A universal constructor of arbitrary abstract data types as well as a comprehensive collection of the built-in types are included into a type system.

• **Class representation**. Classes provide for representing of sets of homogeneous entities of an application domain. Class instances (objects) have specific types.

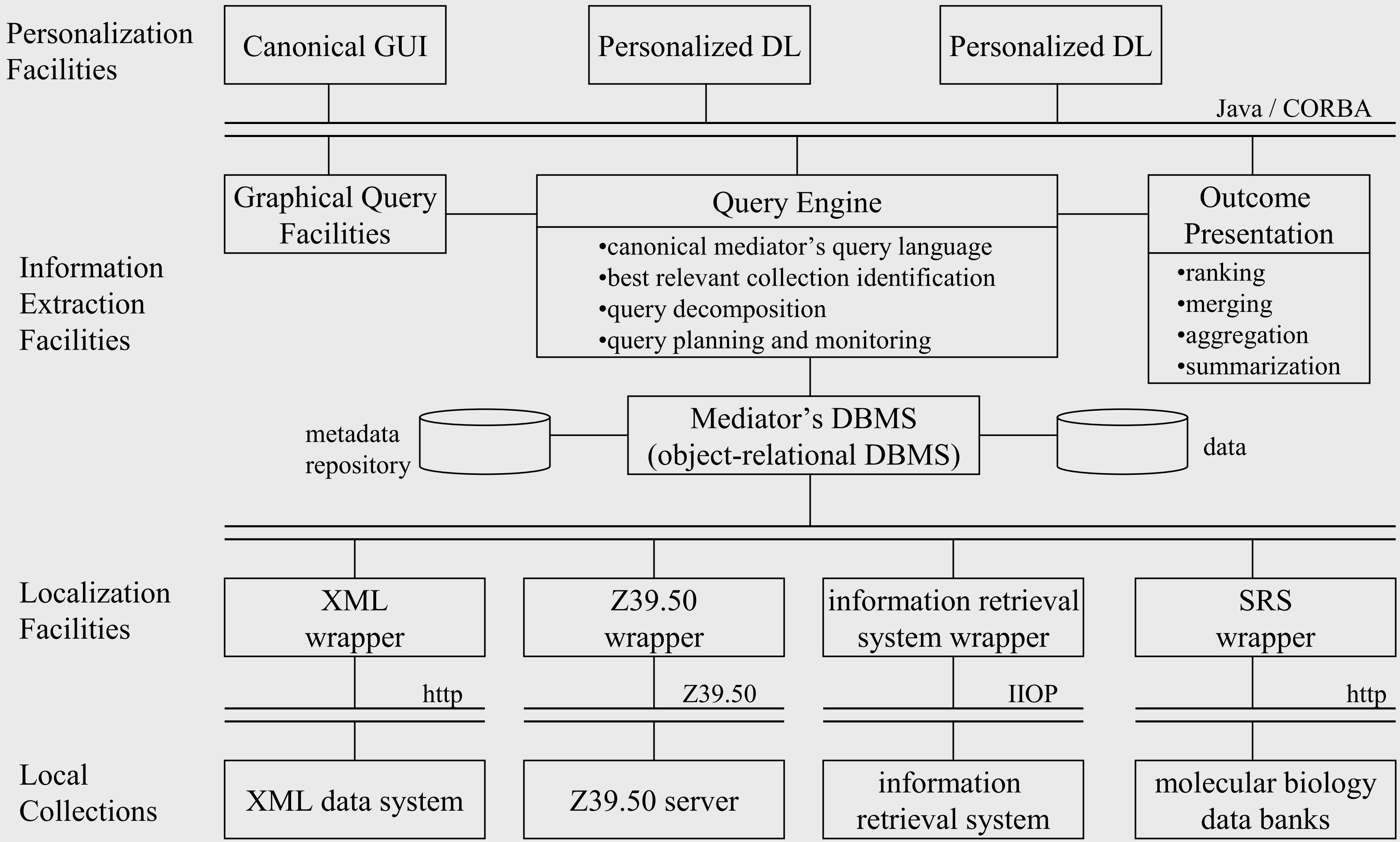
• **Multiactivity (workflow) representation**. These are used for the specification and implementation of interconnected and interdependent application activities, for the specificaton of declarative assertions and concurrent megaprograms over the information resources.

• **Facilities for the logical formulae expressions**. A multisorted object calculus (typed first-order language) is used for querying the integrated set of digital collections as well as for specification of constraints and behaviour.

Mediator's Metainformation Layering



Information Extraction Framework

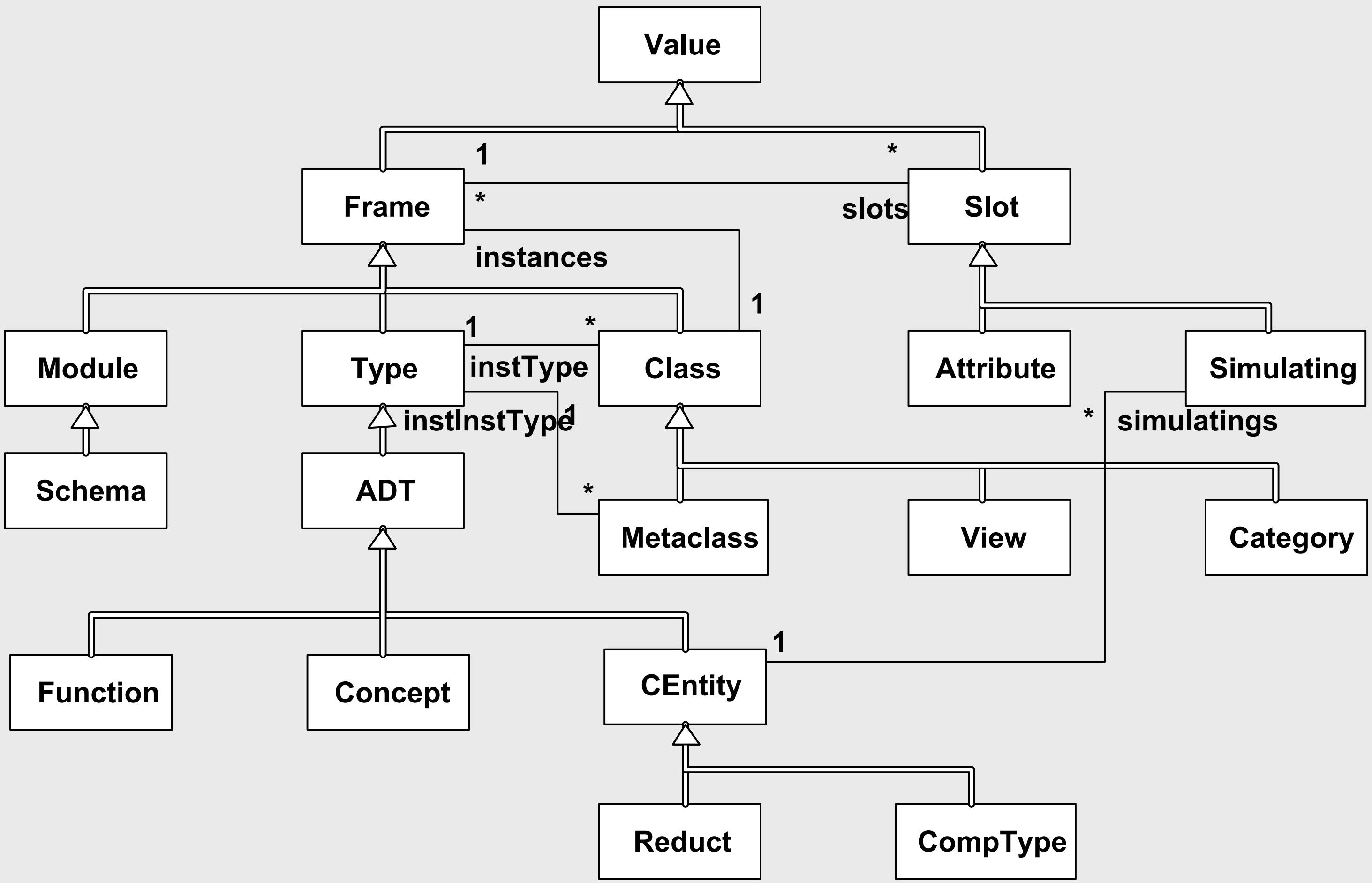


Mediator’s metainformation

Metainformation framework

- Ontological and terminological modeling is a part of the more general information modeling in the mediator, based on one and the same canonical model.
- Thesauri and classifiers are considered to be an important part of a subject domain definition. It means that ontological and terminological modeling are considered in an integrated way.
- Ontologies together with thesaurus definitions are used for semantic integration of information contexts.
- Contexts correlation includes procedures of mapping and integration of ontologies and thesauri.
- Ontologies form a basis for establishing semantic correlations of schema elements from various contexts.
- Approach to metainformation modeling is based on a uniform model of representation for thesauri, classifiers and ontologies, and uniform methods for manipulation of these sorts of information.

Metainformation Repository



Contextualization of Ontology

mapping of local ontological context to that of the mediator

- by names and relationships
- by natural language description
- applying structural integration to concept specifications
- introducing new concepts over existing ones

contextualization through structural correlation

- establishing loose ontological relevance of specification elements applying analysis of intercontext concept relationships
- establishing tight ontological relevance of specification elements introducing a subsumption relationship between concepts

Correlation of Ontological Concepts

evaluation of descriptor weights

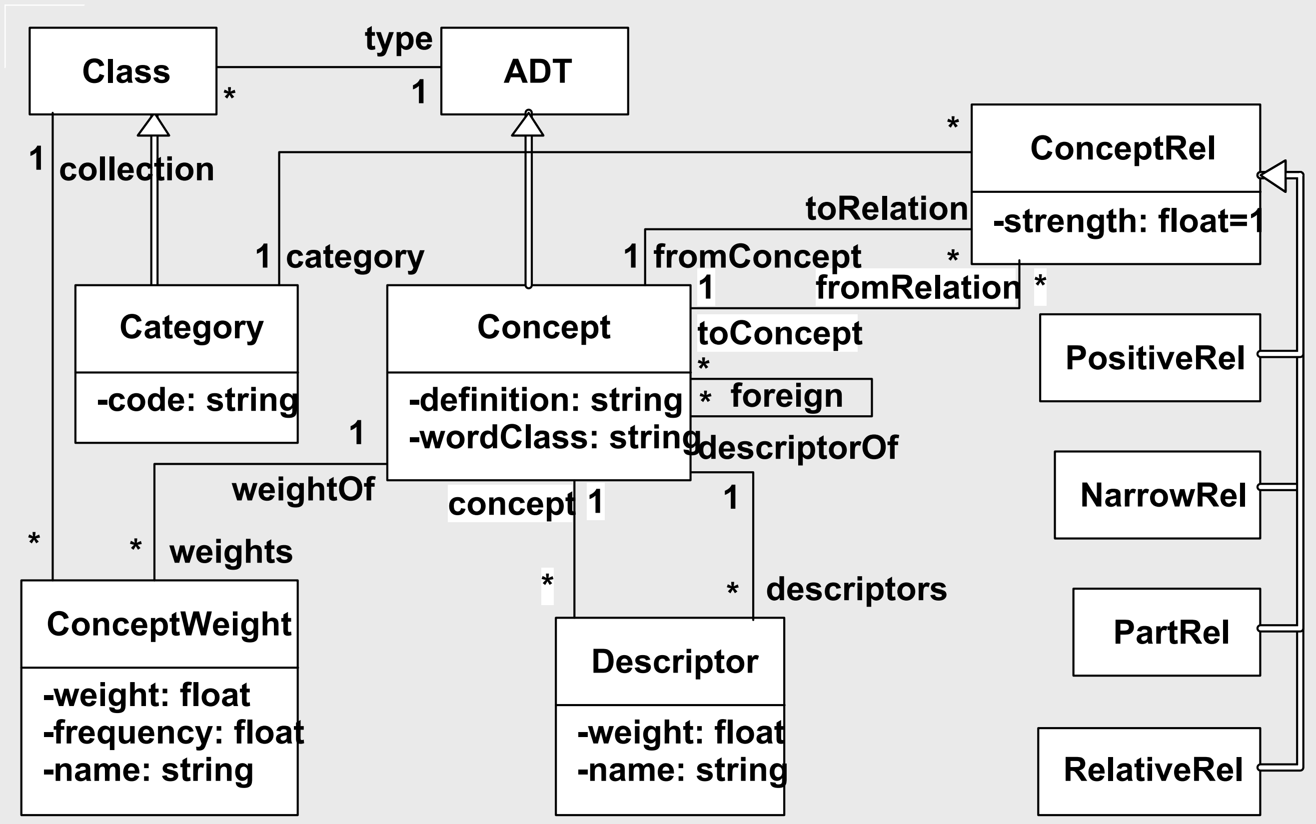
$$W_{Xk} = \frac{f_k \log \frac{N}{n_k}}{\sqrt{\sum_{i \in V_X} \left(f_i \log \frac{N}{n_i} \right)^2}}$$

establishing intercontext relationships between concepts

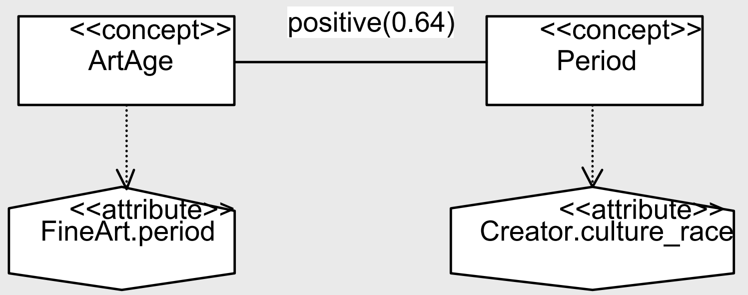
$$sim(X, Y) = \frac{\sum_{k \in V_X \cup V_Y}^{t} (W_{Xk} \cdot W_{Yk})}{\sqrt{\sum_{k \in V_X}^{t} (W_{Xk})^2 \cdot \sum_{k \in V_Y}^{t} (W_{Yk})^2}}$$

$$r(X, Y) = \frac{\sum_{k \in V_X \cup V_Y}^{t} \min(W_{Xk}, W_{Yk})}{\sqrt{\sum_{k \in V_X}^{t} (W_{Xk})^2}} \qquad r(Y, X) = \frac{\sum_{k \in V_X \cup V_Y}^{t} \min(W_{Xk}, W_{Yk})}{\sqrt{\sum_{k \in V_Y}^{t} (W_{Yk})^2}}$$

Ontological Metainformation



Identification of Ontologically Relevant Elements



Collection registration at a mediator as a process of compositional development

Process of an Information Source Registration

For each *source* class the following steps (of the **compositional development** process) are required [LNCS 2151]:

1. **relevant federated classes identification**

   - Find federated classes that ontologically can be used for defining source class extent in terms of federated classes. To a source class several federated classes may correspond covering with their instance types different reducts of an instance type of the source class. On another hand, several source classes may correspond to one federated class.

2. **most common reducts construction**

   For an instance type of each identified *federated* class do:

   - Construct most common reducts for instance type of this federated class and source class instance type to concretize (partially) such federated instance type. Most common reduct may include also additional attributes corresponding to those federated type attributes that can be derived from the source type instances to support them.

   - In this process for each attribute type of the common reduct a concretizing type, concretizing function or their combination should be constructed (this step should be recursively applied).

Process of an Information Source Registration

For each *source* class the following steps are required:

3. **partial source view construction**

   - For each relevant federated class construct a partial source view expressing a constraints in terms of the federated class that should be satisfied by values of respective most common reducts of source class instances. Thus partial views over all relevant federated classes will be obtained.

4. **partial views composition**

   - Construct compositions of the source type most common reducts obtained for instance types of all federated classes involved.

   - Construct a source view as a composition of partial views obtained above. This is an expression of a materialized view of an information source in terms of federated classes. An instance type of this view is determined by the most common reducts composition constructed above.

Subject Mediator. Cultural Heritage Collections. Collections Registration

Federated Level Metainformation

Local into Federated Level Mapping

<u>CIMI Profile of z39.50</u>

museum_object	**creator_c**
created_by* date_collected* description* object_id* relation* … content_general collection mrObject	nationality works

<u>Louvre Museum Web Site</u>

department	**author**
name description sections	name nationality works

<u>Uffizi Museum Web Site</u>

artist	**canvas**
name biography paint_list	title painter date history description to_image

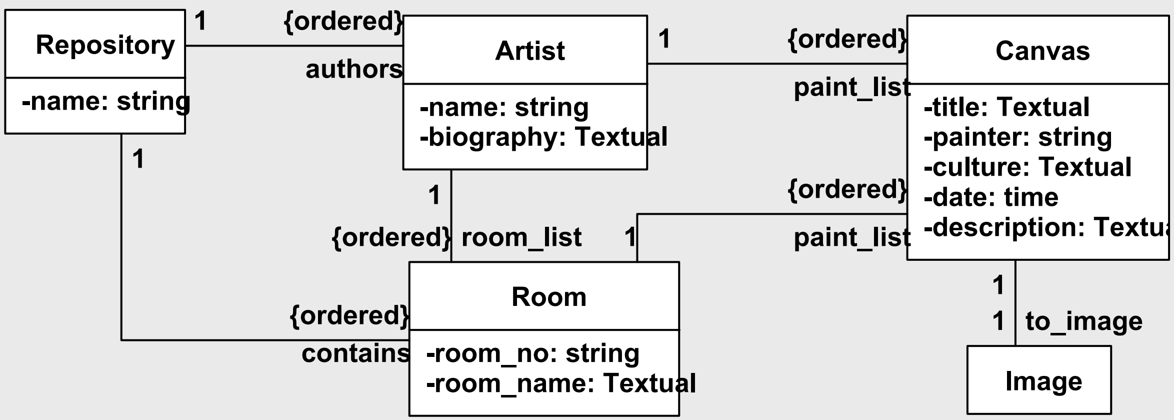
Local Views in Terms of Federated Classes

creator_c(c/Creator_Creator_Info [name, nationality, date_of_birth, date_of_death, works/{set_of:Heritage_Entity_Museum_Object}]) ⊆ creator(c[name, nationality, date_of_birth, date_of death, works])

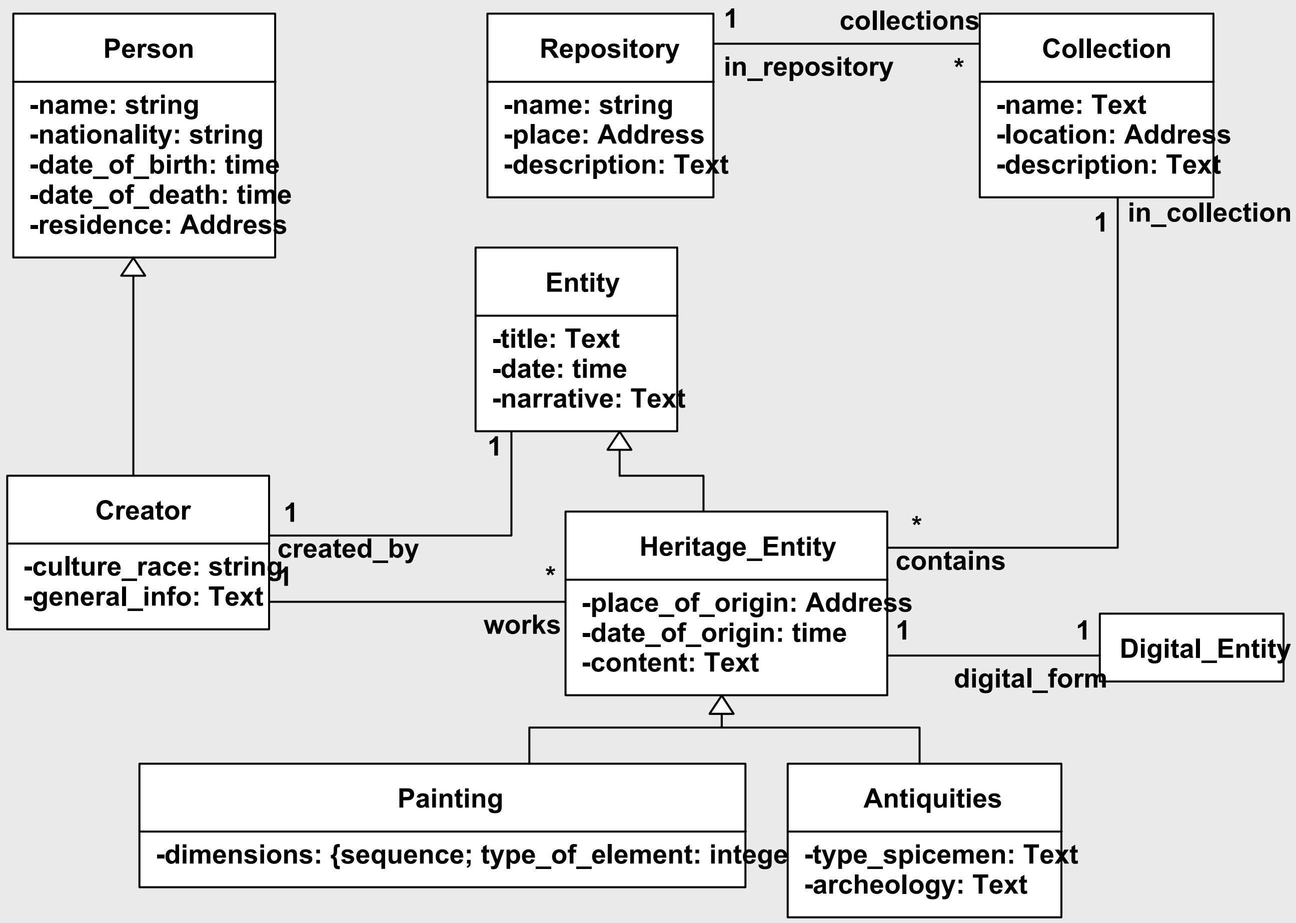
author (a/Creator_Author[name/fname, nationality, works/{set_of:Heritage_Entity_Work}]) ⊆ creator(name, nationality, works (w)) & ∃ c,s (repository (c/Collection [contains(s/Section)]) & repository.name = ‘Louvre’ & in (w, s.contains))

artist(a/Creator_Artist[name, nationality, general_info/Text_Textual, works/{set_of: Painting_Canvas}]) ⊆ creator(a[name, nationality, general_info, works]) & repository (n/name, collection) & n = ‘Uffizi’ & ∃ col/Collection (¬ isempty (intersect (collection(col/ Collection).contains, works)))

Specifications of Types of the Uffizi Site Schema



Specifications of Types of the Mediator Schema



Most Common Reduct (Example)

```
{CR_Painting_Canvas;
  in: c_reduct;
  metaslot
      of: Canvas;
      taking: {title, painter, date, description, to_image};
      reduct: R_Painting_Canvas
  end;
  simulating: {
    R_Painting_Canvas.title ~ CR_Painting_Canvas.title;
    R_Painting_Canvas.created_by ~
                CR_Painting_Canvas.get_created_by;
    R_Painting_Canvas.date_of_origin ~ CR_Painting_Canvas.date;
          ... };
  get_created_by: {in: function;
      params: {+ext/CR_Painting_Canvas, -returns/Creator};
      predicative: {ex c/Canvas ((c/CR_Painting_Canvas = ext) &
          ex a/Artist ((c.painter = a.name) &
          returns = a/CR_Creator_Artist)))}}
         ...
}
```

Partial Source View Construction (Example)

The formula expressing the local class *canvas* is terms of the federated class *painting* is defined as:

canvas(p/CR_Painting_Canvas) $\subseteq$ *painting(p/R_Painting_Canvas) &*
p.in_collection.in_repository = 'Uffizi‘

Specification of a class (actually, this is *local as view* class) containing this formula is:

```
{v_canvas_painting;
 in: class;
 class_section: {
  key: invariant, {unique; {title}};
  lav: invariant, {subseteq (v_canvas_painting(p),
   painting(p/R_Painting_Canvas) &
   p.in_collection.in_repository = 'Uffizi')}
           };
 instance_section: CR_Painting_Canvas
}
```

Source View Composition (Example)

A final formula for a local class *canvas* in terms of the federated classes *painting* and *creator* is:

canvas(p/CR_Painting_Creator_Canvas) $\subseteq$

painting(p/R_Painting_Canvas) & p.in_collection.in_repository = 'Uffizi' &

creator(c/R_Creator_Canvas) & ∃ w/Painting (in(w, c.works) &

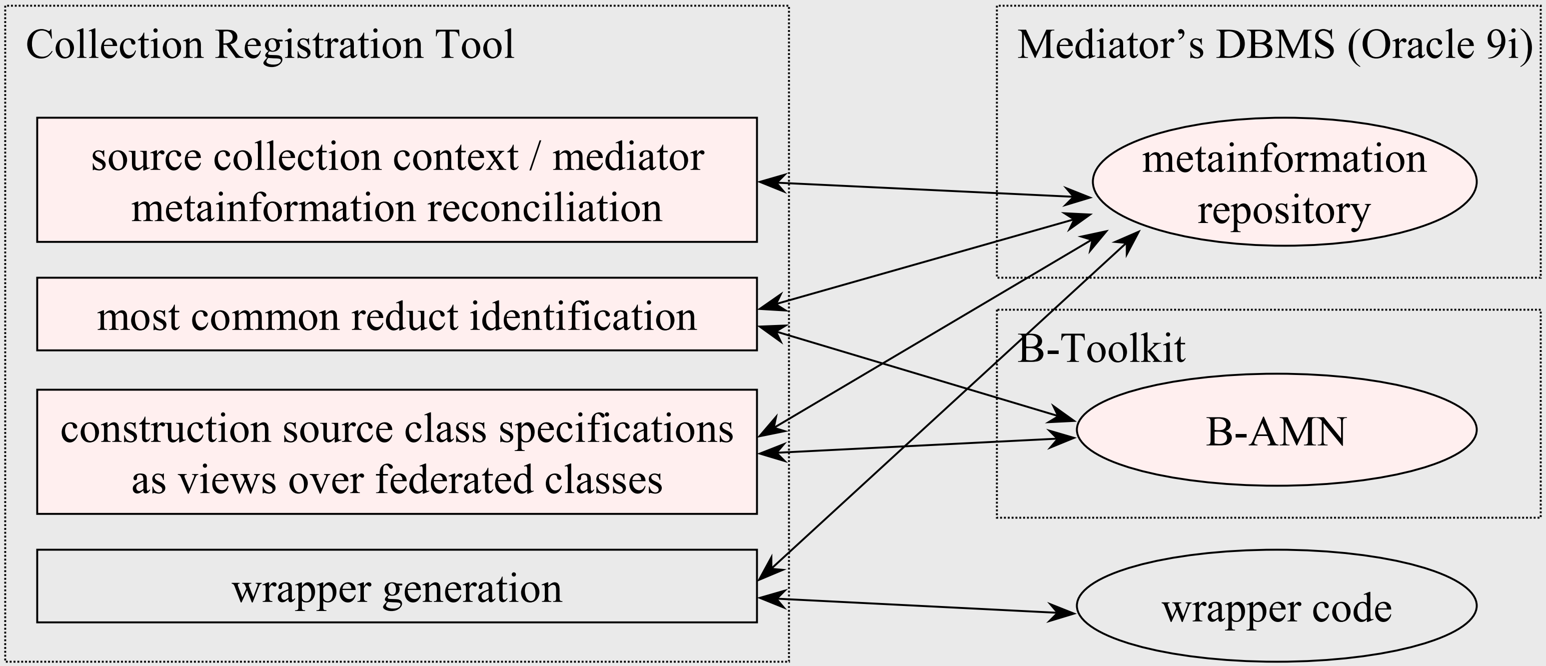
w.in_collection.in_repository = 'Uffizi')

Complete definition of source view looks as follows:

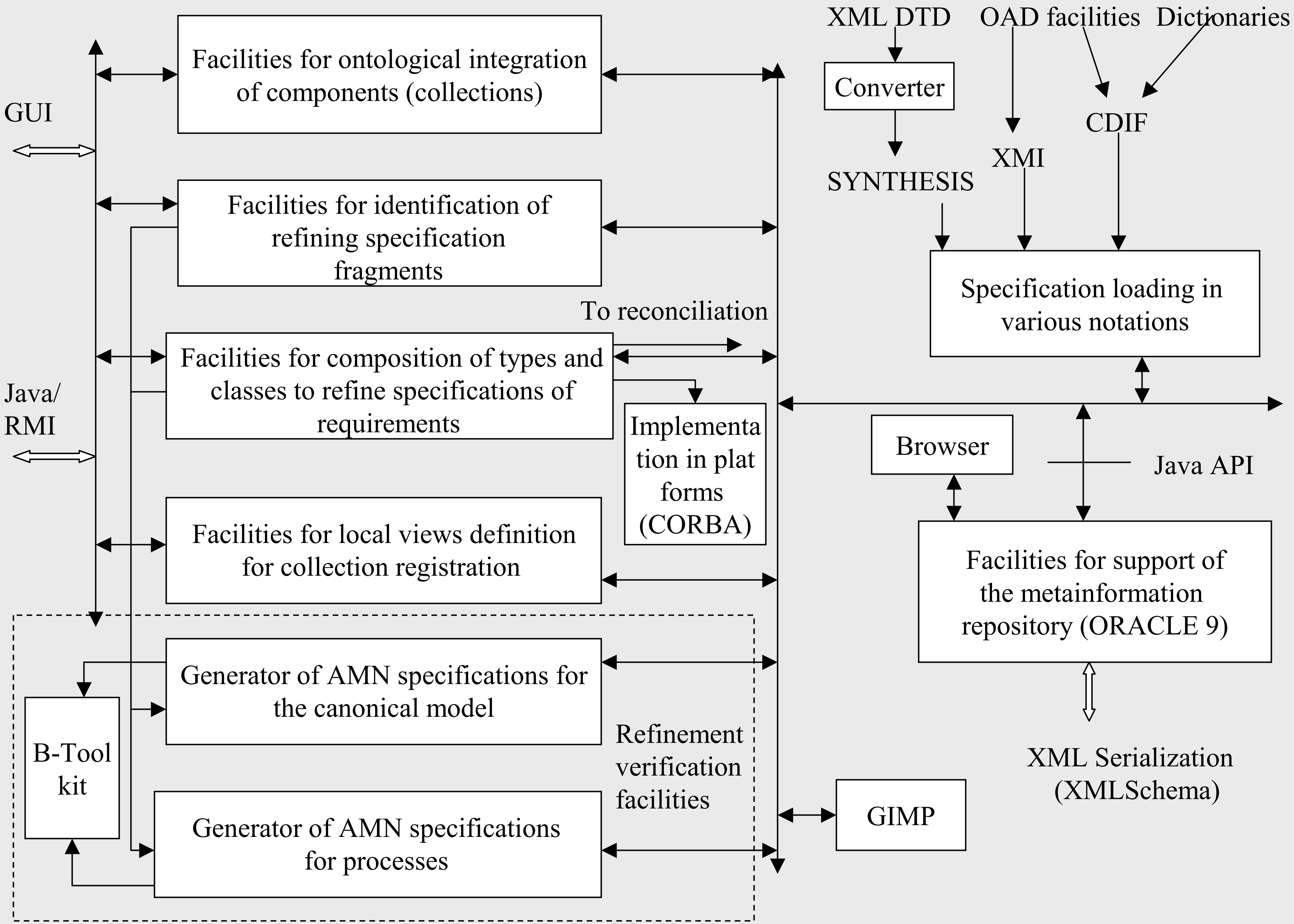
```
{v_canvas;
 in: class;
 class_section: {
  key: invariant, {unique; {title}};
  lav: invariant, {subseteq(v_canvas,
   painting(p/R_Painting_Canvas) &
   p.in_collection.in_repository = 'Uffizi' &
   creator(c/R_Creator_Canvas) & ex w/Painting
   in(w,c.works)& w.in_collection.in_repository = 'Uffizi')})
            };
 instance_section: CR_Painting_Creator_Canvas
}
```

CR_Painting_Creator_Canvas = CR_Painting_Canvas $\sqcup$ CR_Creator_Canvas

Structure of the Collection Registration Tool



COMPOSITIONAL DEVELOPMENT and REGISTRATION



Intermediator Protocol

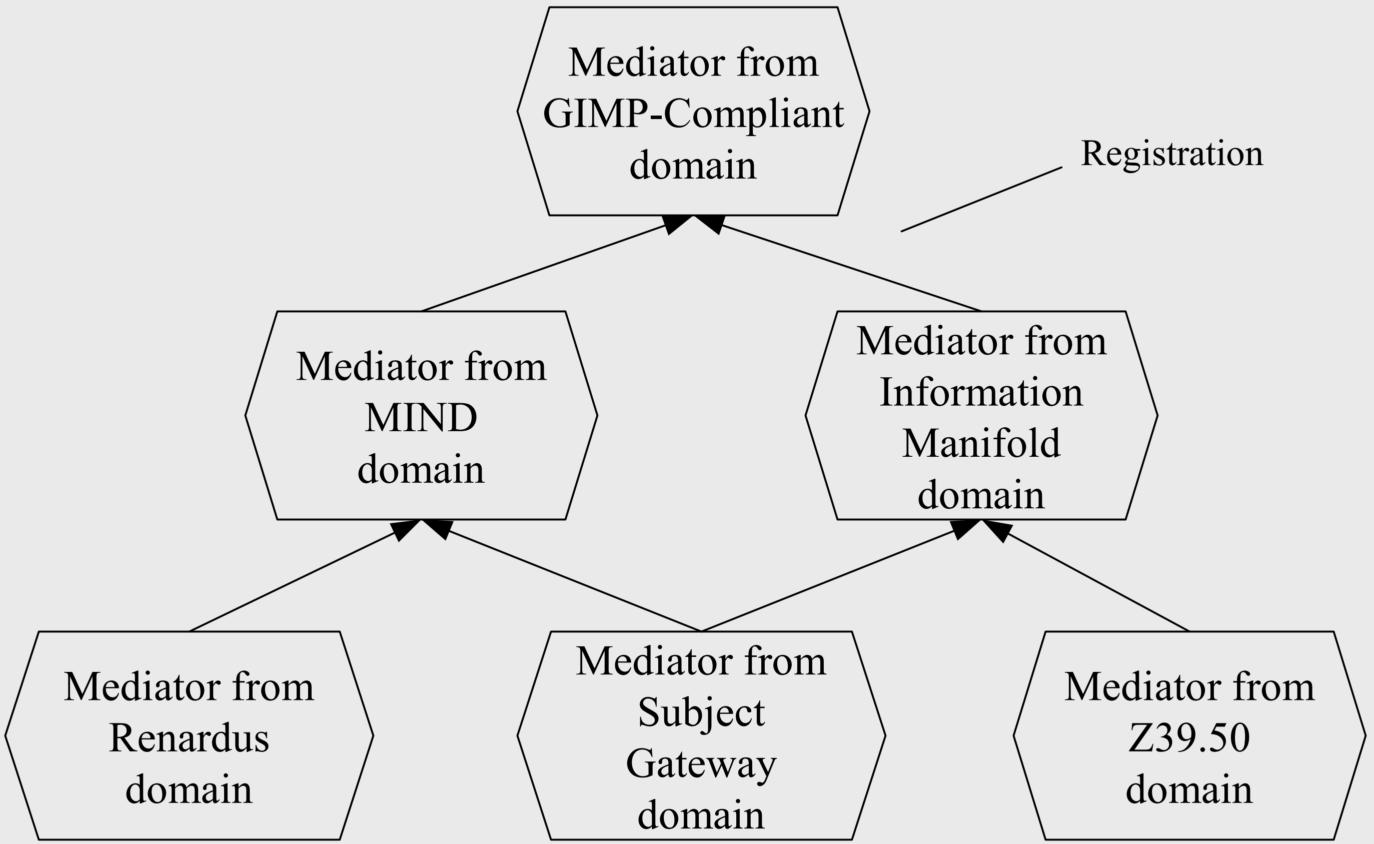
Diversity of Existing Architectures

The diversity of architectures exhibiting certain features of mediation platforms can be observed in distributed digital libraries (DL) area:

- z39.50 architecture and profiles;
- wide networks of DL which enhance connectivity across document repositories and provide a quasi standardized access (e.g., NCSTRL approach to Open Architecture for the Confederated Digital Library);
- community-oriented DL which appears as a common need, subject gateways and broker-based architectures (e.g., Renardus project);
- federated architectures and mediators (MIND project, subject mediation at RAS)
- mediators over ontology-based sources, CARMEN project (ECDL'2001)

The intermediator framework is required to support heterogeneous mediators interoperability with Generalized Intermediator Protocol (GIMP)

Subject Mediators Registration Structure



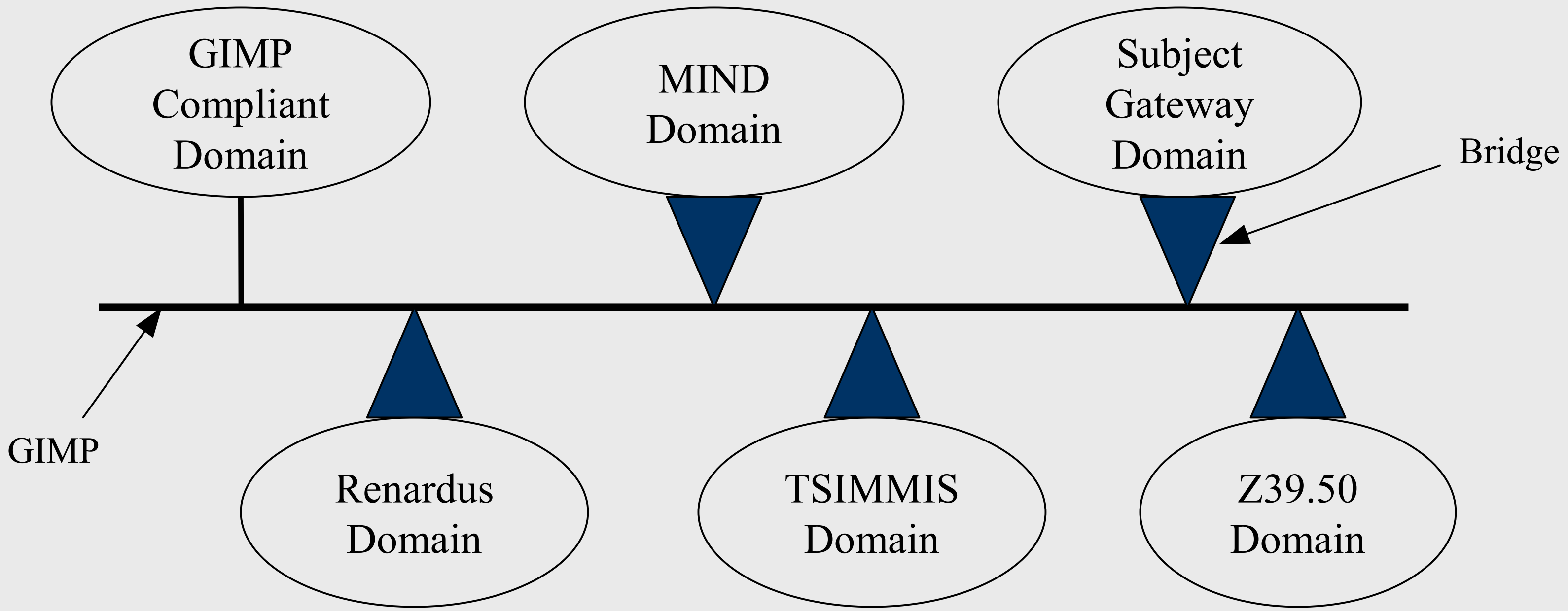
Intermediator Framework and Normative Registration Procedure

- Mediator platform-neutral (canonical) information model leads to unifying models and metainformation representation for various mediator platforms.
- Such modeling, its respective metainformation and representation is called *normative*.
- Each new mediator (source) platform can be abstracted in the common middleware concepts applying specific mapping approaches and tools.
- Mediator services and facilities can be specified as normative, platform independent in frame of the canonical model.
- A registration procedure considered here is such normative procedure independent on the models of participating mediators. Only one such registration procedure is required for the intermediator framework.

Generalized Intermediator Protocol (GIMP)

- GIMP is investigated as a combination of two protocols – extended Simple Digital Library Interoperability Protocol (SDLIP) providing facilities for data access in the intermediator environment and a subset of Open Archive Initiative Protocol (OAIP) providing facilities for mediator metainformation exchange during the registration process.
- *Domain* is a distinct mediator scope, within which certain common mediator features defined by a common mediator platform are exhibited.
- A mapping or *bridging* resides at the boundary between the domains providing the required transformations.

Intermediator Framework



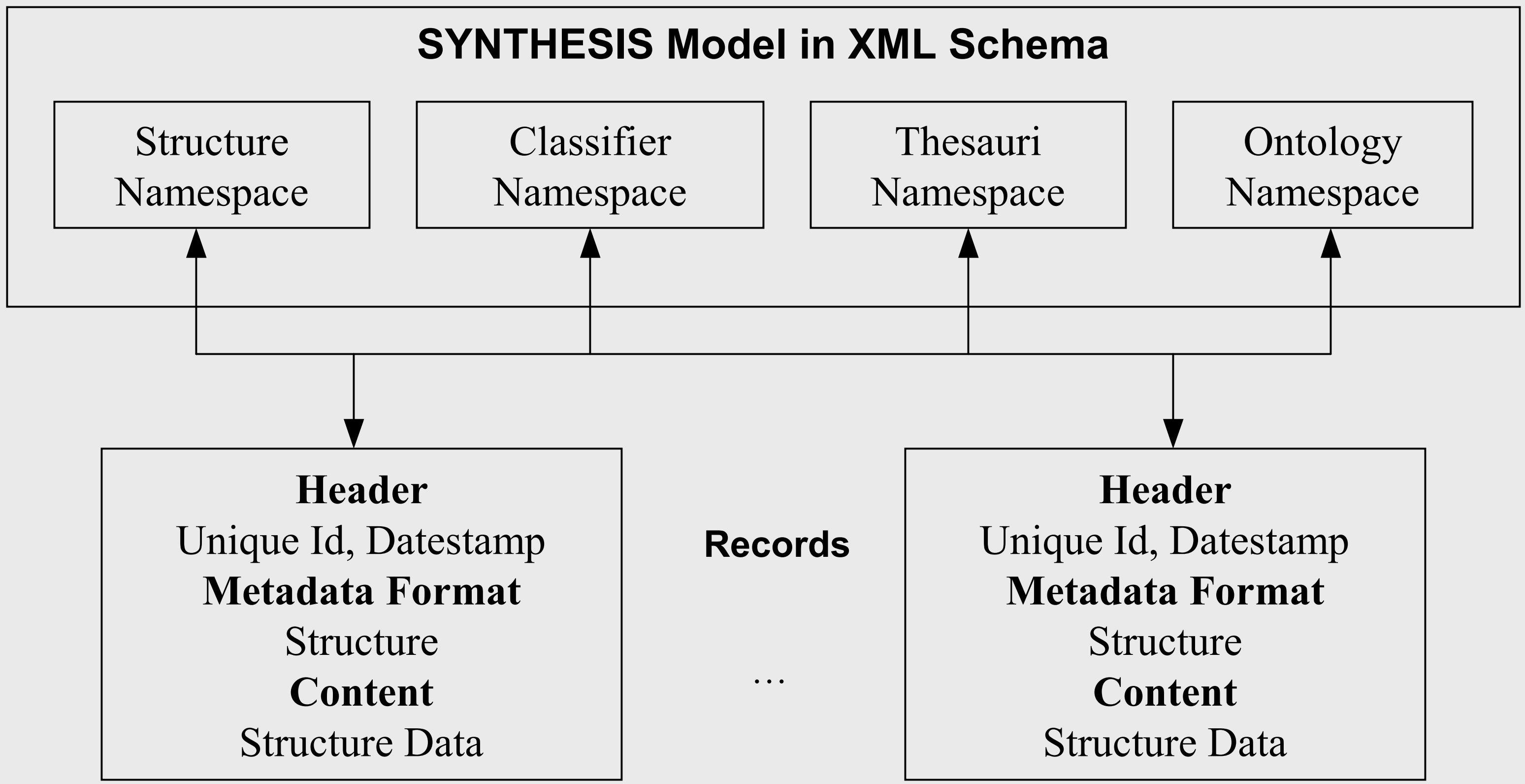
Representation of Metainformation in GIMP

- *Mediator metainformation repository* is a network accessible server to which GIMP protocol requests, embedded in HTTP, can be submitted.
- Metainformation is output in the form of a *record* (XML-encoded byte stream)
- A particular subject mediator (source) is considered a constituent of the repository (OAI *item*).
- Repositories are assumed to be GIMP-compliant (conformant to the canonical model).
- GIMP records are structured exactly as the records of the OAIP.
- The canonical model in GIMP is syntactically represented in the XML Schema language.

Representation of Metainformation in GIMP (2)

- XML Schema definitions providing namespaces for the canonical model are contained in *http://www.ipi.ac.ru/synthesis/projects/XMLBIS/synxmls/*.
- In GIMP, a canonical schema may be organized in modules containing all subject mediator specifications of one of the kinds: structural specifications, ontological specifications, thesaurus specifications, classifier specifications, etc.
- For each kind of a metainformation format an XML Schema definition and a respective namespace are provided.
- A subject mediator can disseminate metainformation in multiple formats mentioned (according to OAIP).
- Each record disseminated by a *GetRecord* protocol request is identified by the combination of the unique identifier of a subject mediator and a metadata prefix, which identifies the metadata format.

GIMP Items and Metadata Formats



Heterogeneous Information Sources Registration Protocol (OAIP based)

- A mediator (source) provider starts the registration by locating a mediator defining a subject domain to be registered in. For a known mediator and a known subject domain, a provider applies an OAIP request:

  ListMetadataFormats <identifier of a subject mediator>

- To get all metainformation specifications of the given kind apply:

  GetRecord <identifier of a subject mediator> <metadataPrefix>

- After obtaining the required specifications from the mediator, the provider contextualizes its source metainformation in the mediator's context.
- After the availability of the local source schema becomes known to the subject mediator, in its turn it applies:

  ListMetadataFormats <identifier of a local source> and after that

  GetRecord <identifier of a local source> <metadataPrefix>

- Mediator takes registration information from the provider, processes and checks the results to complete the registration. The success or failures are reported to the provider.

Extensible Ontological Modeling Framework

Extensible Ontological Modeling Framework for Subject Mediation

- A mediator ontological language (MOL) may depend on a subject domain and is to be defined at the mediator consolidation phase
- For different information sources different ontological models (languages) can be used to define their ontologies
- Reversible mapping of the source ontological models into MOL is needed for information sources registration at the mediator
- A subset of the canonical mediator data model is used for the core of MOL
- MOL can be constructed as a union of extensions of the MOL core equivalent to each of the models in a set *M* of ontological models that are used for the sources potentially relevant to the mediator subject domain
- During operational phase, to make tight ontological integration possible, each source ontological model should be mapped into the MOL
- According to GIMP, registration is done by providers in their native environments. The mediator’s ontology should be mapped into the native model, contextualization is to be done on that level and the result is to be mapped back into MOL.

The Core of the Mediator Ontology Language

- A subset of the mediator's canonical model is used as the MOL Core
- ADT specifications are included with state-based attributes and invariants
- Invariants are expressed using closed formulae of the SYNTHESIS object calculus
- Predicates in formulae correspond to types or their Boolean combinations (expressed with & (intersection), | (union), ^ (complement))
- Association metatypes establishing properties of type attributes are used

If an association *R* is defined on a domain C_1 and a range C_2 then the bounds (pairs of positive integers) define the following. The first bound gives for any object c_1 of C_1 an admissible range (minimal and maximal value) of a number of different objects c_2 of C_2 such that $\{<c_1,c_2>\}$ belongs to *R*. The second bound for any c_2 of C_2 gives a minimal and maximal value of a number of objects c_1 of C_1 such that $\{<c_2,c_1>\}$ belongs to an association inverse to *R*.

MOL core extension equivalent to DAML+OIL model

- Classes and restrictions are mapped into types, properties are mapped into type attributes (that may belong to certain association metatypes), individuals are mapped into frames.
- New built-in SYNTHESIS metatypes (induced by DAML+OIL) are introduced: daml_oil, restriction, disjointWith, disjointUnionOf, onProperty, …
- Class elements, Property restrictions, Cardinality constraints have respective interpretations in MOL core extension
- Classes Thing, Nothing of DAML+OIL are mapped into Taval, Tnone types of MOL respectively. subClassOf <class-expression list> is mapped into supertype <type name list>.
- A type with name T in the type name list results of mapping of a class expression CE (this mapping is denoted for short CE>T)

Class Elements Interpretation

- disjointWith element asserting that C is disjoint with the class-expression CE in the element is mapped into the invariant:

  I: {in: invariant, disjointWith, {{(C>T1 & CE>T2 = {}) }}}

- disjointUnionOf element asserts that C has the same instances as the disjoint union of the class-expressions element (all of the classes defined by the class-expressions of a disjointUnionOf element must be pairwise disjoint). For two class expressions CE1 and CE2 the resulting invariant looks as:

  I: {in:predicate, invariant, disjointUnionOf, {{(C>T = (CE1>T1 | CE2>T2)) & (CE1>T1 & CE2>T2 = {}) }}}

```
{Person;
     in: type, daml_oil;
    …
     dunion: {in: invariant, disjointUnionOf, {{(Person(p/Person) =
                    (Man(p/Person) | Woman(p/Person))) & (Man(p/Person) &
                    Woman(p/Person) = {}) }}}
};
```

Property Restrictions Interpretation

- daml:toClass element defines the class of object x for which it holds that if the pair (x,y) is an instance of P, then y is an instance of the class-expression CE>T or datatype.

  I: {in: invariant, onProperty, toClass,{{all x (ST(x) & (P(x) $\subseteq$ CE>T))}}}

{R_Person;
    in: type, daml_oil, restriction; supertype: Animal;
    hp: {in: invariant, onProperty, toClass, {{all ap (R_Person(ap) &
        hasParent(ap) <= Person)}}};
    hasFather: Male;
    metaslot in: HasFatherCard end };

Here:

{HasFatherCard;
    in: metatype, association, daml_oil; superclass: HasFather;
    instance_section:
    {domain: Animal; range:Male;
     association_type: {{1,1},{0,inf}};
        metaslot in: onProperty, cardinality, UniqueProperty end}};

Cardinality Constraints Interpretation

- daml:maxCardinality element. This defines the class of all objects that have at most N distinct values for the property P.

```
{HasSpouse;
      in: metatype, association, daml_oil;
      instance_section:
      {domain: Person;
        range: Person;
        association_type: {{0,1},{0,1}}
            metaslot
               in: onProperty, maxCardinality
            end}
     };
```

Cardinality Constraints Interpretation (2)

- daml:maxCardinalityQ element. This defines the class C of all objects that have at most N distinct values for the property P that are instances of the class expression CE or datatype (and possibly other values not belonging to the class expression or datatype).

```
{domain: C>T1 ;
     range: CE>T2;
     association_type: {{0,N},{?,?}};
       metaslot     in: onProperty, Cardinality
       end }
{HasOccupation;
    in: metatype, association, daml_oil;
     instance_section:
     {domain: Person;
       range: FullTimeOccupation;
       association_type: {{0,1},{0,inf}}
          metaslot
              in: onProperty, CardinalityQ
          end
 };
```

Reverse Mapping of the MOL obtained into DAML+OIL

General rules for the reverse mapping are the following:

- MOL type that does not belong to restriction metatype is mapped into DAML+OIL class (subclass). Type invariants belonging to disjointWith, disjointUnionOf, sameClassAs metatypes are mapped into the respective elements of the resulting class definition.
- MOL type attributes together with the semantics introduced by the related association metatypes are mapped into DAML+OIL properties (subproperties).
- MOL type that belongs to restriction metatype is mapped into DAML+OIL restriction. In such cases invariants of MOL type belonging to metatypes toClass, hasValue, hasClass are mapped into the respective elements of the resulting restriction.
- MOL type attributes P of a type C>T defined in an association metatype having association types related to onProperty metatype are mapped into cardinality or cardinalityQ property restriction of a class C.

Example of the reverse mapping

For R_Person type the following restrictions will be generated (toClass and cardinality restrictions):

```
<daml:Class rdf:about="#Person">
    <rdfs:subClassOf>
        <daml:Restriction>
            <daml:onProperty rdf:resource="#hasParent"/>
            <daml:toClass rdf:resource="#Person"/>
        </daml:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <daml:Restriction daml:cardinality="1">
            <daml:onProperty rdf:resource="#hasFather"/>
        </daml:Restriction>
    </rdfs:subClassOf>
</daml:Class>
```

Query rewriting

Steps of query rewriting

- Produce inverse rules out of the mediator's views definitions

  $v(X) \text{ :- } p1(X1), \ldots, pn(Xn), C$ as

  $\rho(pi(Xi)) \text{ :- } v(X)$ (for $i = 1, \ldots, n$),

  ρ maps the non-distinguished variable to the corresponding Skolem functions
- Find sequence of IR heads having the same *class names* as subgoals in a query. Several destinations can be produced.
- Define mapping of free variables in destinations to variables in respective subgoals of Q For each destination and variable mappings defined, construct a target formula T: $q(X) \text{ :- } f1(p1(Y1)), \ldots, fn(pn(Yn)), \delta(C)$
- Replace $fi(pi(Yi))$ with view atoms. If the constraint δ (C) and the inferred constraints of the view atoms in p-formula are consistent, then output the formula.
- If there are Skolem functions in formulae, try to eliminate them.
- Discriminate source and mediator (post-processing) parts of query interpretation
- Establish executable ordering of subqueries

Example: Inverse Rules

- canvas(p/Painting_Canvas) ⊆ painting(p/Painting_Canvas) &
  p.in_collection.in_repository.name = 'Uffizi'
- canvas(c/Creator_Canvas) ⊆ creator(c/Creator_Canvas) &
  in(c.works, w/Painting) & w/Painting.in_collection.in_repository.name = 'Uffizi'
- Painting_Creator_Canvas = Painting_Canvas ⊔ Creator_Canvas
- canvas(p/Painting_Creator_Canvas) ⊆ painting(p/Painting_Canvas) &
  p.in_collection.in_repository.name = 'Uffizi' & creator(c/Creator_Canvas) &
  in(c.works, w/Painting) & w/Painting.in_collection.in_repository.name = 'Uffizi'

Inverse rules for the view def above IRs are produced from partial views.

- painting(p/Painting_Canvas) :- canvas(p/Painting_Canvas)
- creator(c/Creator_Canvas) :- canvas(c/Creator_Canvas)
- in(c/Creator_Canvas.works, p/Painting_Canvas) :-
  get_in(c/Creator_Canvas.works, p/Painting_Canvas)

Query Rewriting Example

q(p):- creator(c/Creator), c.name = ‘Rafael’, painting(p/Painting), in(c.works, p), p.title = ‘Madonna’, p.in_collection.in_repository.name = ‘Uffizi’

- **Destination:**

  creator(c/Creator_Canvas), painting(p/Painting_Canvas), in(c/Creator_Canvas.works, p/Painting_Canvas)

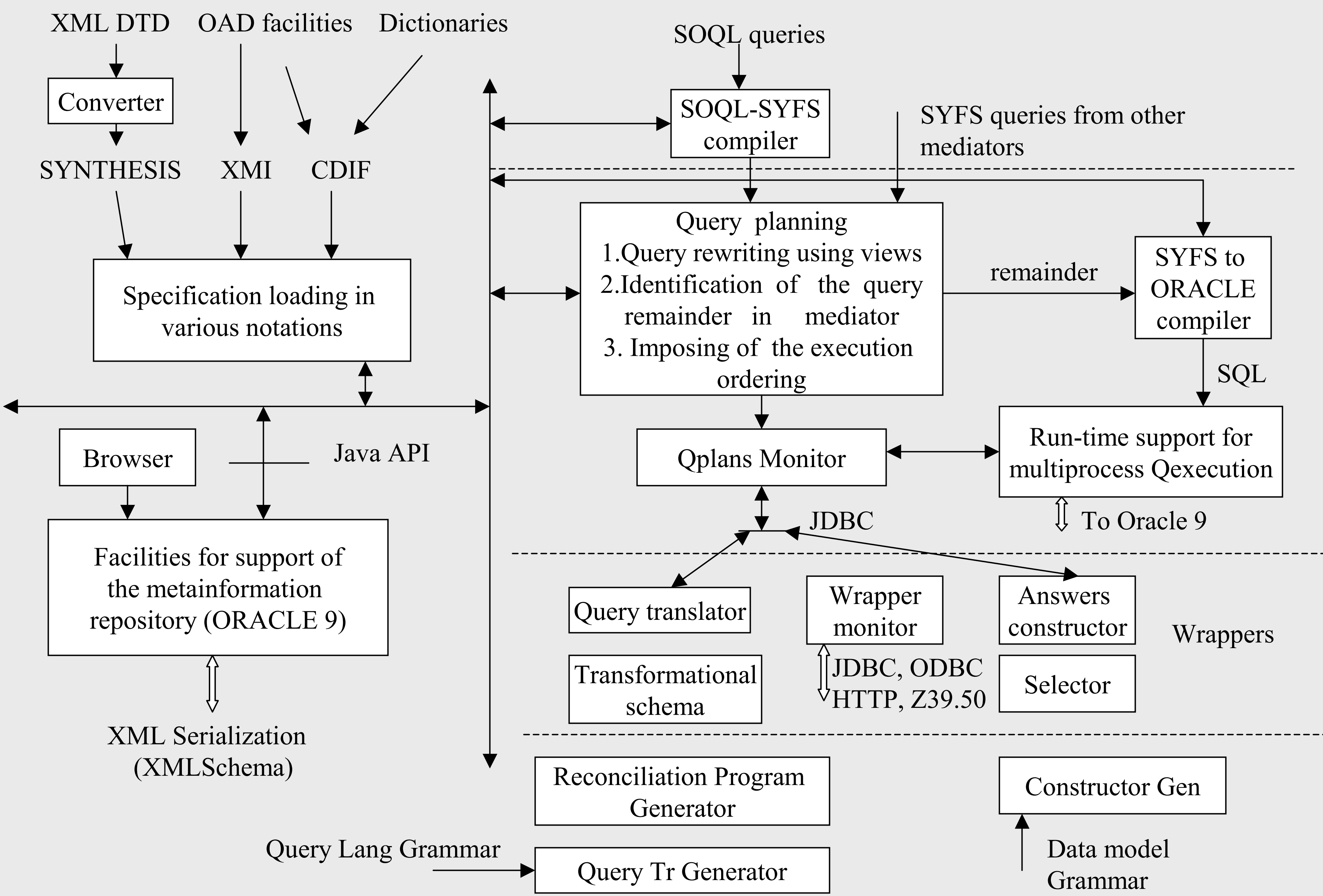
- **Target**

  creator(c/Creator_Canvas), painting(p/Painting_Canvas), in(c/Creator_Canvas.works, p/Painting_Canvas), c.name = ‘Rafael’, p.title = ‘Madonna’, p.in_collection.in_repository.name = ‘Uffizi’

- **Replace with views applying inverse rules:**

  canvas(c/Creator_Canvas), canvas(p/Painting_Canvas), get_in(c/Creator_Canvas.works, p/Painting_Canvas), c.name = ‘Rafael’, p.title = ‘Madonna’, p.in_collection.in_repository.name = ‘Uffizi’

MEDIATOR INSTRUMENTATION



Mediator for the gene expression regulation domain

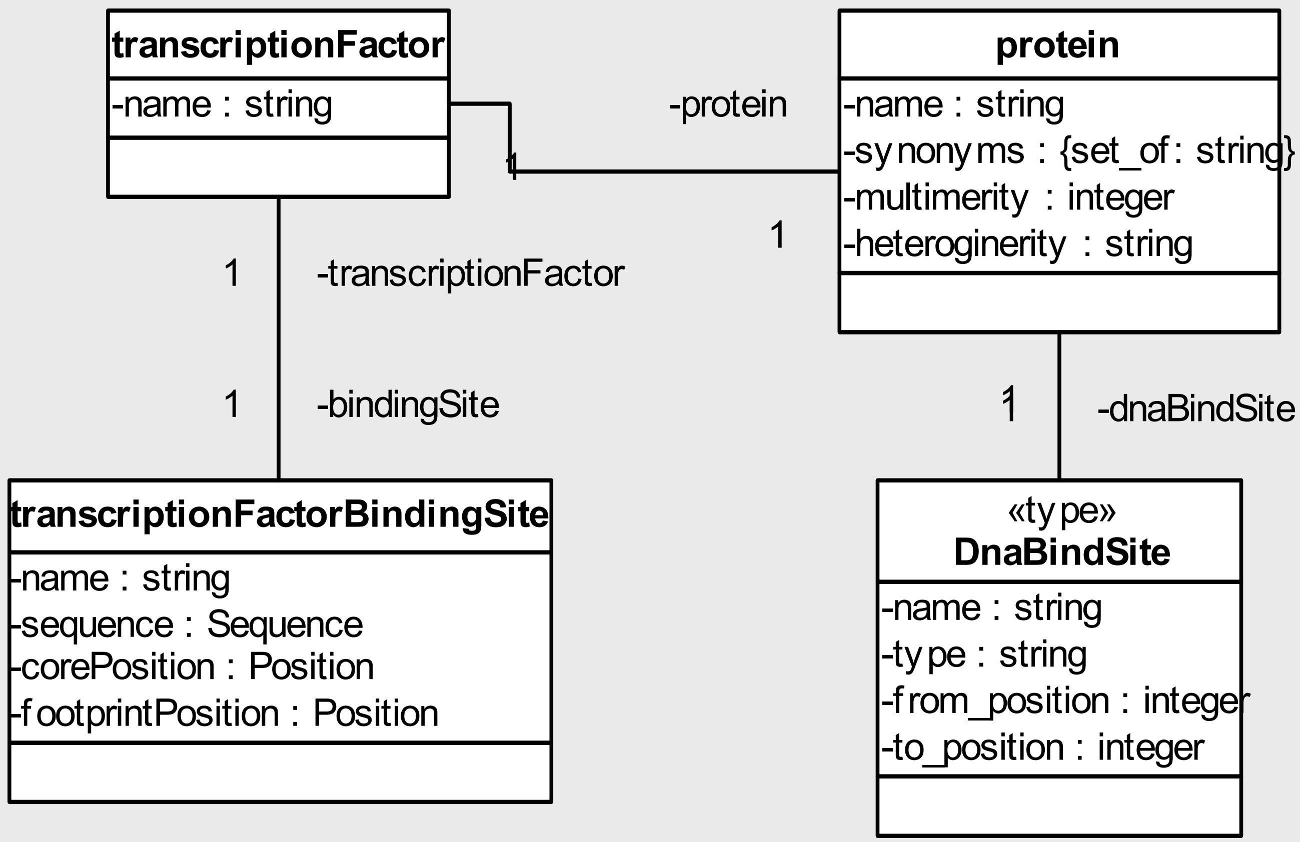
- The model of the subject domain (gene expression regulation) has been developed. The model has a multilevel structure and includes ontological definition of the related concepts and thesauri, definition of information structuring, types of experiments, data analysis methods, as well as the related models of the respective theory.
- The mediator is oriented on a broad class of problems. The intuition behind them can be provided by an example sequence of interrelated queries to the mediator that are intended for preparation of the training samples of regulatory regions, which may be used further by recognition programs: to output the set of transcription factor binding sites sequences, which have a definite type of DNA-binding domain, search for transcription factors corresponding to the proteins found, search for transcription factor binding sites; search for the sequences of pre-ordered length including relevant transcription factor binding sites.

Ontological definition examples

- **Name** "protein"
- **Definition** "A large molecule composed of one or more chains of amino acids in a specific order; the order is determined by the base sequence of nucleotides in the gene coding for the protein. Proteins are required for the structure, function, and regulation of the body's cells, tissues, and organs, and each protein has unique functions. Examples are hormones, enzymes, and antibodies."

- **`Name`** "transcription factor"
- **`Definition`** "A protein that regulates transcription after nuclear translocation by specific binding with DNA or by stoichiometric interaction with a protein that can be assembled into a sequence-specific DNA-protein complex."
- **`Part-of`** "transcription complex"
- **`Subclass-of`** "regulatory protein"
- **`Subclass-of`** "protein"

Fragment of the mediator schema specification in UML



Information sources

Initial set of information sources to be registered at the mediator includes:

- The database TRRD developed at the Institute of Cytology and Genetics that contains information about structural and functional organization of extended transcription regulating regions of eukaryotic genes and their expression.
- The database SWISSPROT contains an information about the structure and functions of genes, about their domain structure, sequences, etc.
- The databases EMBL/GenBank accumulate information about the sequences of DNA, RNA, their exon-intron structure, and other functional layout.
- The database Medline/PubMed stores bibliography that is required for supporting and verifying the data presented.

Sources Registration Example

Views are expressed by means of the inverse rules expressing classes of the mediator through classes of sources (first rule relates protein to a class in SWISSPROT, two other rules relate *transcriptionFactor* and *transcriptionFactorBindingSite* to the respective classes in TRRD):

- *protein(p/Protein_SProtein) :- sprotein(p/Protein_SProtein)*
- *transcriptionFactor(t/TranscriptionFactor_FACTORS) :- factors(t/TranscriptionFactor_FACTORS)*
- *transcriptionFactorBindingSite(s/TranscriptionFactorBindingSite_SITES) :- sites(s/TranscriptionFactorBindingSite_SITES)*

Query rewriting and planning

- Query: *Display the transcription factor binding sites with the definite types of DNA binding domain*
- In the mediator's canonical model this query is expressed as:

transcriptionFactorBindingSite(s) & transcriptionFactor(t) & protein(p) & s.transcriptionFactor = t & t.protein = p & p.structure.type = "HOMEBOX"

- After query rewriting applying the inverse rules above, we get the query:

sites(s/TranscriptionFactorBindingSite_SITES) & factors(t/TranscriptionFactor_FACTORS) & sprotein(p/Protein_SProtein) & s.transcriptionFactor = t & t.protein = p & p.structure.type = "HOMEBOX"

- This query is planned as a subquery SQ1 to TRRD and a subquery SQ2 to SWISSPROT with the remaining postprocessing in the mediator SQ3:

SQ1(s,t):- FACTORS(t/TranscriptionFactor_FACTORS) & SITES(s/TranscriptionFactorBindingSite_SITES) & s.transcriptionFactor = t

SQ2(p):- sprotein(p/Protein_SProtein) & p.structure.type = "HOMEBOX"

SQ3(s,t,p) :- SQ1(s,t) & SQ2(p) & t.protein = p

Applications of subject mediators

- Subject domain mediation has good perspectives for heterogeneous information sources integration in process of formation of professional communities around the Internet
- The approach looks promising for the worlds of multiple dynamically changing sources (content, availability) providing also for mediator's scalability
- Provides for interoperability of DL metadata registries to cross the boundaries between different information contexts
- Reaching more flexibility in developing geo-referenced mediators (instead of rigid "bucket" schema any geo-referencing mediator can be defined and implemented)
- Introducing subject domains (educational disciplines) contexts above heterogeneous collections, digital libraries for sharing and managing semantic information
- Subject domains semantic interoperability through mediators composition

References

- Kalinichenko L.A. Methods and tools for equivalent data model mapping construction In Proceedings of the International Conference on Extending Database Technology (EDBT'90), Venice, 1990
- Kalinichenko L.A. Workflow Reuse and Semantic Interoperation Issues. Advances in workflow management systems and interoperability. A.Dogac, L.Kalinichenko, M.T. Ozsu, A.Sheth (Eds.),. NATO Advanced Study Institute, Istanbul, August 1997
- Kalinichenko L.A. Method for data models integration in the common paradigm. In Proceedings of the First East European Conference “Advances in Databases and Information Systems”, St. Petersburg, September 1997
- D. O. Briukhov, L. A. Kalinichenko.Component-Based Information Systems Development Tool Supporting the SYNTHESIS Design Method. Proc. of the East European Conference on "Advances in Databases and Information Systems", Poland, Springer, LNCS No.1475
- L. A. Kalinichenko. Compositional Specification Calculus for Information Systems Development. In Proc. of the Third East European Conference on Advances in Databases and Information Systems (ADBIS'99), Maribor, Slovenia, Lect. Notes Comput. Sci., 1999
- L. A. Kalinichenko. Integration of heterogeneous semistructured data models in the canonical one. In Proc. of the First Russian National Conference on "Digital Libraries: Advanced Methods and Technologies, Digital Collections", Saint-Petersburg, October 1999

References

- Kalinichenko L.A., Briukhov D.O., Skvortsov N.A., Zakharov V.N. Infrastructure of the subject mediating environment aiming at semantic interoperability of heterogeneous digital library collections. Proc. of the Second Russian National Conference on "Digital Llibraries: Advanced Methods and Technologies, Digital Collections, Sep. 26-28, 2000, Protvino
- N. A. Skvortsov, L. A. Kalinichenko. An Approach to Ontological Modeling and Establishing Intercontext Correlation in the Semistructured Environment. Proc. of the Second Russian National Conference on "Digital Llibraries: Advanced Methods and Technologies, Digital Collections, Sep. 26-28, 2000, Protvino
- Briukhov D.O., Kalinichenko L.A., Skvortsov N.A. Information sources registration at a subject mediator as compositional development. Proceedings of the Fifth East European Symposium on Advances in Databases and Information Systems (ADBIS'01), Springer, LNCS, 2001
- Kalinichenko, L.A., Briukhov D.O., Tyurin I.N., Skvortsov N.A. Intermediator framework protocol for information sources registration at heterogeneous mediators In Proceedings of the DELOS Workshop on Interoperability in Digital Libraries, September 8-9, 2001, GMD-IPSI, Darmstadt, Germany
- D.O. Briukhov, L.A. Kalinichenko, N.L .Podkolodny, V.N . Zakharov, Institute for Problems of Informatics RAS, Moscow, Institute of Cytology and Genetics SB RAS, Novosibirsk, Mediation of heterogeneous information resources in the gene expression regulation domain. To be published in Proceedings of the BGRS’2002 Conference, August 2002

References

- D. O. Briukhov, L. A. Kalinichenko, N. A. Skvortsov, S.A.Stupnikov Value Reconciliation in Mediatorsof Heterogeneous Information Collections Applying Well-Structured Context Specifications. In Proc. of the BalticDB’2002 Conference, June 2002
- L. A. Kalinichenko, N. A. Skvortsov. Extensible Ontological Modeling Framework for Subject Mediation. Submitted to ECDL’2002