Registering Scientific Information Sources for Semantic Mediation*

Amarnath Gupta[†], Bertram Ludäscher[†], and Maryann E. Martone[‡]

[†]San Diego Supercomputer Center [‡]Department of Neurosciences
University of California, San Diego
{gupta,ludaesch}@sdsc.edu mmartone@ucsd.edu

Abstract. In a conventional information mediation scenario it is assumed that all sources, including their schemas, are known before the integrated view is defined. We have found this assumption to be unrealistic for scientific information integration – new relevant sources are discovered quite frequently, and need to be integrated *incrementally* with an existing federation. In this paper, we address the issue of *source registration*, the mechanism by which a new information source "registers" its semantics with the mediator, such that not only new views can be defined with the newly joining source, but existing views can benefit from the source without any redefinition. We approach the problem in the framework of *semantic* (a.k.a. *knowledge-based* or *model-based*) *mediation*, a version of information integration where the sources cannot be integrated solely based on their own logical schema, but need additional domain knowledge at the mediator to "glue" them together. We solve the problem by introducing a process called *contextualization*, whereby a source specifies a set of axioms to express its own conceptual model relative to the mediator's knowledge base. To this end, we present a *context specification language* $\mathcal{CSL}$ that allows the user to specify this mapping, and illustrate how the mediator interprets a $\mathcal{CSL}$ specification to update its knowledge schema and preexisting views. The examples are derived from a real-world scenario involving an ongoing collaboration with several neuroscience groups.

1 Introduction

Information integration refers to the problem of combining multiple information sources such that they appear to a user as a single (virtually) integrated source over a single global schema. A mediator is a data integration software that allows one to define such an integrated schema over schemas of the individual sources. In doing so, it hides from the the various heterogeneities arising from differences in data source types, data models and query capabilities among different sources. Given a user query against the global schema, the mediator transparently decomposes it into constituent local subqueries against the appropriate the sources, collects partial query results from the sources, and after due post-processing, reports the combined results to the user. There are two predominant techniques to map the source schemas to the the global schema [FLM98]. In the *global-as-view* (GAV) model, the global schema is defined as

* work partially supported by NIH BIRN-CC 3 P41 RR08605-08S1, NSF/NPACI Neuroscience Thrust ASC-975249, and DOE SciDAC/SDM DE-FC02-01ER25486

a view over local schemas. Hence mediated data objects are "fused" together from parts obtained from data objects in one or more sources. In contrast, in the *local-as-view* (LAV) model, a global schema is defined first by modeling the application domain. Then the source schemas and their data objects are defined as views over the global schema. For query evaluation, the rules have to be "inverted" or "folded" (if possible) [LRO96,Hal01,GM02].

Recently, research has been devoted to the problem of *semantic* (also called knowledge-based or model-based) mediation [GLM00,LGM00,LGM01]. Semantic mediation adds a few additional considerations to the logical information integration problem as described above. In this scenario, sources not only export their logical schema, but also their *conceptual model* to the mediator, thus exposing their concepts, roles, classification hierarchies, and other high-level semantic constructs to the mediator. The mediator, in turn, exposes a conceptual schema to the user, based on the conceptual schemata of individual sources. Semantic mediation allows information sources to export their schema at an appropriate level of abstraction to the mediator. If the individual sources have different *local ontologies* or namespaces, the mediator needs to reconcile their differences, or establish some well-defined relationships among them. To enable this reconciliation, the mediator use a *global ontology* to correlate the terminology from different sources. The mediator's ontology includes general facts and rules that hold in the specific domain of application (or common-sense knowledge as for a dictionary [MWK00]), and some additional rules that explicitly specify the relationships among the conceptual models of the different sources.

In this paper, we continue our prior research [GLM00,LGM00,LGM01] on semantic integration of scientific information, and address the *source registration problem*. In a state-of-the-art information mediation scenario (semantic or otherwise), it is presumed that all sources, including their schemas, are known before the integrated view is defined. In our experience with scientific information sources, this is not a realistic assumption – new relevant sources are discovered quite frequently, and need to be integrated *incrementally* with an existing federation already operational among a number of previously known sources. The registration problem refers to the mechanism by which a new information source "semantically registers" with the mediator, such that not only new views can be defined with the newly joining source, but existing views can benefit from the source without any redefinition. We approach the problem by introducing a process called *contextualization*, whereby a source declares, in addition to its conceptual model, an extra set of axioms to express its own conceptual model in terms of the the mediator's knowledge base. The axioms are expressed in what [RTU01] calls an *interschema language*. The mediator, in turn, processes the contextualization axioms, by compiling them from the interschema language into its own internal formalism, and updating its knowledge base and views.

The organization and main contributions of the paper are as follows. In the remainder of this section, we present related work. In Section 2, we describe how an information source can specify its local semantics without explicitly linking to the mediator's ontology. In Section 3, we present $\mathcal{CSL}$, our *context specification language* that serves as the primary vehicle for source registration. Source registration is accomplished by specifying mappings between the source's ontology and the mediator's ontology using

$\mathcal{CSL}$ declarations. This is followed in Section 4 by a brief description of how $\mathcal{CSL}$ statements are processed in the mediator to complete the registration procedure. We conclude in Section 5, with a brief discussion and an outlook on future work.

Related Work

The general problem of defining a global schema over a set of local schema is not a new problem, see, *e.g.*, [SL90,PS98]. In [RR97] a seven-step methodology for schema integration based on semantic integrity constraints is presented; see [Tür99] for a comprehensive treatment of semantic integrity constraints in federated databases. For this paper, we focus on related research in three areas.

Conceptual Schema Integration. Research in schema integration techniques have been undertaken since the mid 80's [SL90]. The primary focus in information integration is in resolving relation and attribute conflicts between to-be-integrated schemas. A fraction of schema integration research has investigated the problem of conceptual schema integration. For example, the use of "integration operators" *copy*, *generalize*, *join*, and *simplify* has been proposed [CE93] to integrate conceptual schemas. In contrast, constraint based approaches have been proposed for operations like *generalization*, *type assignment*, and *exclusion constraints* to correlate two conceptual graphs [EJ95]. Benn et al. [BCG96] first transform the relational schema of each source to an object schema consisting of classes, attributes and semantic constraints. Schema integration is achieved using two groups of first-order rules. With the first group the integrator *identifies subgraph isomorphisms* between the schemas based on their semantic similarity. The second group of rules provide *merger rules*, including rules that state when two schemas, although similar, cannot be merged.

Semantic Mediation. Significant progress has been made in the general area of data mediation in recent years, and several prototype mediator architectures have been designed by projects like TSIMMIS [GMPQ$^+$95], SIMS [KMA$^+$98], Information Manifold [LRO96], Garlic [HKWY97], and MIX [BGL$^+$99]. While these approaches focus mostly on structural and schema aspects, the problem of *semantic mediation* has also been addressed:

In the DIKE system [PTU00], the focus is on automatic extraction of mappings between semantically analogous elements from different schemas. A global schema is defined in terms of a conceptual model (SDR network) where the nodes represent concepts and the (directed) edge labels represent their semantic distances and a score called *semantic relevance* that measures the number of instances of the target node that are also instances of the source node. The correspondence between objects are defined in terms of *synonymies*, *homonymies* and *sub-source similarities*, defined by finding maximal matching between the two graphs.

ODB-Tools [BB01] is a system developed on top of the MOMIS [BCV99] system for modeling and reasoning about the common knowledge between two to-be-integrated schemas. They present the object-oriented language ODL$_{I^3}$ derived from a description logic (OCDL). The language allows a user to create complex objects with finite nesting of values, union and intersection types, integrity constraints and quantified paths. These constructs are used to define a class in one schema as a *generalization*, *aggregation*, or *equivalent* with respect to another; *subsumption* of a class by another can be inferred. An integrated schema is obtained by clustering schema elements that are close to one another in terms of an affinity metric.

Calvanese et al. [CCG$^+$01] perform semantic information integration using an LAV approach by expressing the conceptual schema by a description logic language called $\mathcal{DLR}$, and subsequently defining non-recursive Datalog views to express source data elements in terms of the conceptual model. The language $\mathcal{DLR}$ represents concepts C, relations R, and a set of assertions of the form $C_1 \sqsubset C_2$ or $R_1 \subset R_2$, where R_1, R_2 are $\mathcal{DLR}$ relations with the same arity. Mediation is accomplished by defining "reconciliation correspondences", specifications that a query rewriter uses to match a conceptual level term to data in different sources.

Recently Peim et al. [PFPG02] have proposed to extend the well-known TAMBIS system [GSN$^+$01]. Their approach is similar to ours [GLM00,LGM01] in that a logic-based ontology (in their case the $\mathcal{ALCQI}$ description logic) interfaces with an "object-wrapped" source. Their work focuses on how a query on the ontology is transformed to monoid comprehensions for semantic query optimization. In contrast, this paper addresses the issue of how to *dynamically* register a new object source with pre-existing ontologies at the mediator.

Ontology Merging. The problem of ontology matching and merging stems from AI and KRDB research, and is now revisited by the Semantic Web community. Work on the Cyc Upper Ontology [Hov97], the Ontomorph system [Cha00], the Chimaera system [MFRW00], the PROMPT algorithm [NM00] and the FCA-MERGE algorithm [SM01] are all different techniques to represent and find term-matching relationships so that they can be put into a concept lattice.

The ONION system [MWK00] performs *algebraic composition of ontologies*. They use a special "semantic implication" relation $P \Rightarrow Q$ which relates graph patterns P and Q by making the assertion that the object Q *semantically belongs to* the class P. Bridge rules for semantic implications are typically expressed as simple Horn clauses, and are translated to graph operations. The system also admits functional rules, permitting simple functions to be executed as part of the ontology correlation process.

Summary. We note that most of the semantic information integration methods and systems described above essentially correlate schema elements from different sources with some common relationships including class-instance, class-subclass, relation-specialization, part-whole, class equivalence, class-subsumption, and algebraically composable classes. We posit that the source registration problem, introduced above, requires a more general *rule-definable* approach compared to current semantic correlation methods and should make the mappings between models "first-class citizens" [BHP00].

2 Modeling Source Semantics

To enable semantic mediation, an information source needs to be wrapped in such a way that it exports a *conceptual model* CM of the source rather than its logical schema. This

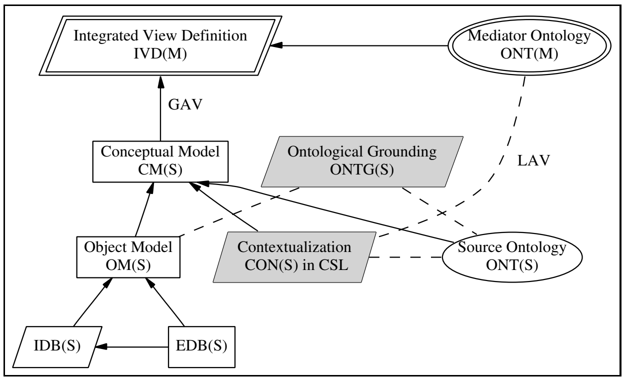

Fig. 1. Semantic Source Registration: logical components and their dependencies

requires the CM wrapper to provide the mediator with additional information about its object class structure and constraints that its logical schema may not provide. We call this the "lifting" of the source to a conceptual level. More details about our semantic mediation strategy can be found in [LGM01]. In the sequel, we describe in detail the conceptual model of the source after it has been "lifted" by the CM wrapper. In Section 3 we show how to specify the mapping between this lifted model and the ontology of the mediator.

2.1 Conceptual Model of the Source

The core components of the *conceptual model* $\mathsf{CM}(S)$ of a source S are:

$$\mathsf{CM}(S) = \mathsf{OM}(S) \cup \mathsf{ONT}(S) \cup \mathsf{CON}(S)$$

The logical components an their dependencies are depicted in Fig. 1:

- $\mathsf{OM}(S)$ is the *object model* of the source S and provides signatures for *classes*, *associations* between classes, and *functions*. $\mathsf{OM}(S)$ structures can be defined extensionally by facts (EDB), or intensionally via rules (IDB).
- $\mathsf{ONT}(S)$ is the *local ontology* of the source S, *i.e.*, defines *concepts* and their *relationships* from the source's perspective.
- $\mathsf{ONTG}(S)$ is the *ontological grounding* of $\mathsf{OM}(S)$ in $\mathsf{ONT}(S)$; it links the object model $\mathsf{OM}(S)$ (classes, attributes, associations) to the concepts and relationships of $\mathsf{ONT}(S)$.
- $\mathsf{CON}(S)$ is the *contextualization* of the local source ontology relative to a mediator ontology $\mathsf{ONT}(M)$.
- $\mathsf{IVD}(M)$ is the mediator's *integrated view definition* and comprises logic view definitions in terms of the sources' object models $\mathsf{OM}(S)$ and the mediator's ontology $\mathsf{ONT}(M)$. By posing queries against the mediator's $\mathsf{IVD}(M)$, the user has the illusion to interact with a single, semantically integrated source instead of interacting with independent, unrelated sources.

Classes in $\mathsf{OM}(\mathsf{CCDB})$

EXPERIMENT(id:id, date:date, cell_type:string, images:SET(image)).
IMAGE(id:id, instrument:ENUM{c_microscope, e_microscope}, resolution:float, size_x:int, size_y:int, depth:int, structures:SET(structure), regions:SET(deposit)).
STRUCTURE(id:id, name:string, length:float, surface_area:float, volume:float, bounding_box:Cube).
DEPOSIT(id:id, substance_name:string, deposit_type:string, relative_intesity:ENUM{dark,normal,bright}, amount:float, bounding_box:Cube).
...

Associations in $\mathsf{OM}(\mathsf{CCDB})$

co_localizes_with(DEPOSIT.substance_name, DEPOSIT.substance_name, STRUCTURE.name).
surrounds(s1:STRUCTURE, s2:STRUCTURE).
...

Functions in $\mathsf{OM}(\mathsf{CCDB})$

deposit_in_structure(DEPOSIT.id) $\rightarrow$ SET(STRUCTURE.name)
...

Source Ontology – $\mathsf{ONT}(\mathsf{CCDB})$

brain $\xrightarrow{has(co)}$ cerebellum $\xrightarrow{has(co)}$ cerebellar cortex $\xrightarrow{has(co)}$ vermis (ONT1)
dendrite $\xrightarrow{has(co)}$ spine process $\xrightarrow{has(pm)}$ spine (ONT2)
cell $\xrightarrow{projects\_to}$ brain_region
globus_pallidus $\xrightarrow{isa}$ brain_region. ... denaturation $\xrightarrow{isa}$ process. (ONT3)
$tc\_has(co)$:= **transitive_closure**($has(co)$). $tc\_has(pm)$:= **transitive_closure**($has(pm)$). (ONT4)
has_co_pm := **chain**($tc\_has(co), tc\_has(pm)$) (ONT5)
...

Ontological Grounding – $\mathsf{ONTG}(\mathsf{CCDB})$

domain(STRUCTURE.volume) **in** [0,300]
domain(STRUCTURE.name) **in** $tc\_has(co)$(cerebellum) (OG1)
domain(EXPERIMENT.cell_type) **in** $tc\_has(co)$(cerebellum) (OG2)
EXPERIMENT.cell_type $\xrightarrow{projects\_to}$ globus_pallidus (OG3)
DENATURED_PROTEIN $\xrightarrow{exhibits}$ denaturation. (OG4)
...

Fig. 2. Conceptual Model for Registering the Cell-Centered Database [MGW+02]

In the following, we present the local parts of $\mathsf{CM}(S)$, *i.e.*, $\mathsf{OM}(S)$, $\mathsf{ONT}(S)$, and $\mathsf{ONTG}(S)$ through a running example. The contextualization $\mathsf{CON}(S)$ is described in Section 3.

Example 1 (Cell-Centered Database: CCDB) Fig. 2 shows pieces of a simplified version of the conceptual model $\mathsf{CM}(\mathsf{CCDB})$ of a real-world scientific information source called the *Cell-Centered Database*, [MGW+02]. The database consists of a set of EXPERIMENT objects. Each experiment collects a number of cell IMAGES from one or more instruments. For each image, the scientists mark out cellular STRUCTURES in the image and perform measurements on them [MGW+02]. They also identify a second set of regions, called DEPOSITs, in images that show the deposition of molecules of proteins or genetic markers. In general, a region marked as deposit does not necessarily coincide with a region marked as a structure. □

Note that $\mathsf{OM}(\mathsf{CCDB})$ in Fig. 2 includes classes that are instantiated with observed data, *i.e.*, the extensional database $\mathsf{EDB}(\mathsf{CCDB})$. In addition to classes, $\mathsf{OM}(\mathsf{CCDB})$ stores *associations*, *i.e.*, n-ary relationships between object classes. The association co_localizes_with specifies which pairs of substances occur together in a specific structure. The object model also contains *functions*, such as the domain specific methods that can be invoked by a user as part of a query. For example, when the mediator or another client calls the function CCDB.deposit_in_structure(), and supplies the *id* of a deposit object, the function returns a set of STRUCTURE objects that spatially overlap with the specified deposit object.

Next, we describe the source's local ontology, $\mathsf{ONT}(\mathsf{CCDB})$. In our world, an ontology $\mathsf{ONT}(S)$ consists of a set of *concepts* and inter-concept *relationships*, possibly augmented with additional inference rules and constraints.[1] The ontological grounding $\mathsf{ONTG}(S)$ links the object model $\mathsf{OM}(S)$ to the source ontology $\mathsf{ONT}(S)$.

The source ontology serves a number of different purposes:

Creating a Terminological Frame of Reference. For defining the terminology of a specific scientific information source, the source declares its own controlled vocabulary through $\mathsf{ONT}(S)$. More precisely, $\mathsf{ONT}(S)$ comprises the terms (*i.e.*, *concepts*) of this vocabulary and the *relationships* among them. The concepts and relationships are often represented as nodes and edges of a directed graph, respectively. Two examples of interconcept relations are has(co) and has(pm) which are different kinds of part-whole relationships [2]. In Fig. 2, items ONT1 and ONT2 show fragments of such a concept graph. Once a concept graph is created for a source, one may use it to define additional constraints on object classes and associations.

Semantics of Relationships. The edges in the concept graph of the source ontology represent inter-concept relationships. Often these relationships have their own semantics that have to be specified within $\mathsf{ONT}(S)$. Item ONT4 declares two new relationships tc_has(co) and tc_has(pm). After registration, the mediator interprets this declaration and creates the new (possibly materialized) transitive relations on top of the base relations has(co) and has(pm) provided by the source S. Similarly, the item ONT5 is interpreted by the mediator using a higher-order rule for chaining binary relations:

- chain(R1,R2)(X,Y) **IF** R1(X,Z), R2(Z,Y)

With this, ONT5 creates a new relationship has_co_pm(X,Y) provided that there is a Z such that tc_has(co)(X,Z), *and* tc_has(pm)(Z,Y).

Ontological Grounding of $\mathsf{OM}(S)$. A local domain constraint specifies additional properties of the given extensional database, and thereby establishes an *ontological grounding* $\mathsf{ONTG}(S)$ between the local ontology $\mathsf{ONT}(S)$ and the object model $\mathsf{OM}(S)$ (Fig. 1). Items (OG1–OG2) in Fig. 2 refines the domains of the attributes EXPERIMENT.cell_type and STRUCTURE.name from the original type declaration (STRING). The refinement constrains them to take values from those nodes of the concept graph that are *descendants* of the concept cerebellum through the has(co) relationship.

[1] *e.g.*, ONT4, ONT5 in Fig. 2 define virtual relations such as *transitive closure* over the base relations

[2] By standards of meronyms, there are different kinds of the has relation: component-object has(co), portion-mass has(pm), member-collection has(mc), stuff-object has(so), place-area has(pa) etc. [AFGP96]

This constraint illustrates an important role of the local ontology in a "conceptually lifted" source. By constraining the domain of an attribute to be concept name C, the corresponding object instance o is "semantically about" C. In addition, this also implies that o is about any ancestor concept C' of C where ancestor is defined via has(co) edges only. Similarly, if a specific instance, STRUCTURE.name has the value 'spine process', it is also about 'dendrite' (ONT2 in Fig. 2).

In addition to linking attributes to concept names, a constraint may also involve inter-concept relationships. Let us assume projects_to(cell, brain_region) is a relationship in the source ontology ONT(CCDB). A constraint may assert that for all instances e of class EXPERIMENT, projects_to(e.cell_type, 'globus_pallidus') holds (OG3). The constraint thus *refines* the original relationship projects_to to suit the specific semantics of OM(CCDB). In Section 3, we will use these constraint-defined correspondences between OM(S) and ONT(S) in the contextualization process.

Intensional Definitions. In the CM wrapper of a source S, we can define virtual classes and associations that can be exported to the mediator as first-class, queriable items by means of an intensional database (view definition) IDB(S). For example, we can create a new virtual class called DENATURED_PROTEIN in IDB(CCDB) via the rule:

DENATURED_PROTEIN(ProtName) **IF**
DEPOSIT(ID, ProtName, protein, dark, _, _), deposit_in_structure(ID) $\neq \emptyset$

Thus, an instance of a DENATURED_PROTEIN is created when a "dark" protein deposit is recorded in an instance of DEPOSIT, and there is some structure in which this deposit is found. As a general principle of creating a CM wrapper, such a definition will be supplemented by additional constraints to connect it to the local ontology. For example, assume that ONT(CCDB) already contains a concept called process. Item (ONT3) defines denaturation as a specialization of process. We can now add the constraint (OG4) to complete the semantic specification about the new DENATURED_PROTEIN object.

Contextual References. It is a standard practice for scientific data sources to tag object instances with controlled vocabulary from a public standard. In (CON1) of Fig. 3, the source states the following mapping rule: The domain of the DEPOSIT.id field can be accessed through an internal method get_expasy_protein_id, which, given a protein name, gets its id from the SWISS-PROT database on the web.[3] How the source enforces this integrity constraint is internal to the source and not part of its conceptual export schema.

2.2 Mediator Information from the Source's Viewpoint

In order to address the source registration issue, we have to specify which components of an existing n-source federation can be "seen", *i.e.*, accessed by the new, $n+1^{st}$ source. A federation at the mediator consists of:

1. currently *registered conceptual models* CM(S) of each participating source S,

[3] http://www.expasy.ch

2. one or more *global ontologies* $\mathsf{ONT}(M)$ residing at the mediator that have been used in the federation, and
3. *integrated views* $\mathsf{IVD}(M)$ defined in a global-as-view (GAV) fashion.

Typical mediator ontologies $\mathsf{ONT}(M)$ are *public*, *i.e.*, serve as domain-specific expert knowledge and thus can be used to "glue" conceptual models from multiple sources. Examples of such ontologies are the Unified Medical Language System (UMLS) from the National Library of Medicine[4] and the Biological Process Ontology from the GeneOntology Consortium[5].

In the presence of multiple ontologies, *articulations*, *i.e.*, mappings between different source ontologies [MWK00] can be used to register with the mediator information about inter-source relationships.

Note that a source S usually cannot "see" all of the above components (1–3) when defining its conceptual model: While S sees the mediator's ontologies $\mathsf{ONT}(M)$ and thus can define its own conceptual model $\mathsf{CM}(M)$ relative to the mediator's ontology in a *local-as-view* (LAV) fashion, it cannot (i) directly employ *another* source's conceptual model $\mathsf{CM}(S')$, nor (ii) can it query the mediator's integrated view $\mathsf{IVD}(M)$ which is defined *global-as-view* (GAV) on top of the sources. The former is no restriction, since S' can register $\mathsf{CM}(S')$, in particular $\mathsf{ONT}(S')$ with the mediator, at which point S can indirectly refer to registered concepts of S' via $\mathsf{ONT}(M)$. The latter guarantees that query processing in this setting does not involve "recursion through the web", *i.e.*, between a source S and the mediator M (the dependency graph in Fig. 1 is acyclic).[6]

3 Context Specification Language $\mathcal{CSL}$

A contextualization $\mathsf{CON}(S)$ "situates" a source's conceptual model $\mathsf{CM}(S)$ in the context given by the mediator's ontology $\mathsf{ONT}(M)$. This is accomplished by mappings between the source ontology $\mathsf{ONT}(S)$ and the mediator ontology $\mathsf{ONT}(M)$. In the following, we present the context specification language $\mathcal{CSL}$ that allows us to express such mappings.

First, observe that a source's object model $\mathsf{OM}(S)$ can be described in terms of special "built-in" predicates C:$\mathsf{classes}(S)$ ("C is a class of S"), A:$\mathsf{assocs}(S)$ ("A is an association of S"), A:$\mathsf{attributes}(S,C)$ ("A is an attribute of class C in S"), and O:$\mathsf{instances}(S,C)$ ("O is an object instance of C in S"). Similarly, the local ontology $\mathsf{ONT}(S)$ can be described by $\mathsf{concepts}(S)$, $\mathsf{relationships}(S)$ and $\mathsf{constraints}(S)$, where the latter are first-order constraints over $\mathsf{concepts}(S)$ and $\mathsf{relationships}(S)$. Analogously, $\mathsf{ONT}(M)$ is described via $\mathsf{concepts}(M)$, $\mathsf{relationships}(M)$ and $\mathsf{constraints}(M)$. We call these special predicates the *model elements* of the source and mediator respectively, and use them to specify source-to-mediator mappings.

$\mathcal{CSL}$ allows one to specify element mappings and access mappings. An *element mapping* is a $\mathcal{CSL}$ expression that specifies how an element of the source's conceptual model relates to that of the mediator. An *access mapping* is a $\mathcal{CSL}$ expression that

[4] `http://www.nlm.nih.gov/research/umls/`
[5] `http://www.geneontology.org/process.ontology`
[6] At the cost of loss of efficiency, the restriction "no recursion through the web" could be lifted.

specifies how an element from source's conceptual model can be physically accessed from the mediator.[7] In this paper, we focus on the element mapping part of $\mathcal{CSL}$ and present the language through examples from our CCDB scenario (Fig. 3).

```
CONTEXTUALIZATION of CCDB – CON(CCDB)

domain(DEPOSIT.id) in get_expasy_protein_id(DEPOSIT.substance_name)
    IF DEPOSIT.deposit_type='protein'                                        (CON1)
map (equivalent)(X,Y)
    IF X:concepts(CCDB), Y:concepts(mediator), X.name = Y.name               (CON2)
map (subconcept)(brain, organ)
    IF brain:concepts(CCDB), organ:concepts(mediator)                        (CON3)
map (subconcept)(axon, compartment)
    IF axon:concepts(mediator), compartment:concepts(CCDB)                   (CON4)
map (concept_concept)(regulates('nejire','CREB'))
    IF 'nejire':concepts(mediator), 'CREB':concepts(CCDB)                    (CON5a)
map (concept_concept)(exists G, regulates('nejire', G), regulates(G, 'CREB'))
    IF 'nejire':concepts(mediator), 'CREB':concepts(CCDB)                    (CON5b)
map (concept_concept)(tc_regulates('nejire','CREB'))
    IF 'nejire':concepts(mediator), 'CREB':concepts(CCDB)                    (CON5c)
map (subrelation)(has(co), has_part)
    IF has(co):relationships(CCDB), has_part:relationships(mediator)         (CON6)
map (instance_concept)(X,ultrastructure)
    IF X:instances(STRUCTURE, CCDB), ultrastructure:concepts(mediator),
    has_part:relationships(mediator), dendrite:concepts(mediator),
    X.name in transitive_closure(part_of)(dendrite)                          (CON7)
map (assoc_rel)(surrounds(s1, s2), inverse(inside(s3,s4))
    IF surrounds(s1, s2):assoc(CCDB), inside(s3,s4):relationships(mediator),
    not has_part(s1, s2).                                                    (CON8)
map (concept_concept)(new evidence_of(regulates)('cfos','CREB'))
    IF 'cfos':concepts(mediator), 'CREB':concepts(CCDB).                     (CON9a)
map (holds) evidence_of(_)(X,Y)
    IF X:concepts(mediator), Y:concepts(CCDB), Z:concepts(mediator),
    evidence_of(_)(X,Z), not opposes(Z,Y).                                   (CON9b)
...
```

Fig. 3. Context Specification for the Cell-Centered Database [MGW+02]

Let us assume the CCDB source intends to inform the mediator that all concepts in ONT(CCDB) are identical to those concepts in the mediator that have the same name. This is expressed in $\mathcal{CSL}$ (CON2 in Fig. 3) as:

map (equivalent)(X, Y) **IF**
X:concepts(CCDB), Y:concepts(mediator),
X.name = Y.name

[7] *e.g.*, for an SQL source the access mapping is an SQL query

The general form of a $\mathcal{CSL}$ statement is

> **map** (*correspondence relation*)($X_1, \ldots, X_n$) **IF**
> *type declarations*,
> *body*

where the *correspondence relation* (*e.g.*, equivalent) specifies which kind of mapping is being defined, thereby instructing the mediator how to compile the statement into a logic program during registration. The *type declarations* specify the kind of model element each variable represents (*e.g.*, X above is of type concepts). The type declaration also specifies whether an X_i belongs to the source or to the mediator. The system ensures that the $X_1, \ldots, X_n$ include both source and mediator model elements (since **map** links *between* source and mediator model elements). Furthermore, *correspondence relations* are themselves typed, *e.g.*, equivalent expects its arguments to be either both concepts or both relationships. The *body* of the $\mathcal{CSL}$ statement is like the body of a logic rule and specifies additional conditions that the mapping must satisfy. All variables in the head of the statement are universally quantified, unless otherwise mentioned. In the following paragraphs we present informal examples of different forms of mapping relations that can be described in $\mathcal{CSL}$.

Subconcept Mapping. Consider C_1:concepts(source) and C_2:concepts(mediator). The correspondence relation "subconcept" defines an isa relation between them. (CON3) in Fig. 3 states that brain, a concept defined in ONT(CCDB) isa organ, defined at the mediator. As discussed in Section 4, after registration, pre-existing integrated views will "see" the CCDB's isa relation through the subconcept mapping established via **map**(subconcept) declarations. In this example, the source concept brain *specializes* the mediator concept organ. Similarly, a source can also *generalize* a mediator concept. Assume, *e.g.*, that ONT(CCDB) has a concept called compartment (not shown in Fig. 2), and the mediator has the concept axon. Item (CON4) states that axons *are* compartments. The mediator has translation rules for both uses of the subconcept mapping.

Concept-Concept Mapping. Subconcept mapping is a special case of inter-concept mapping across the source and the mediator. In general, a concept of the source will be related to a concept at the mediator through a user-specified relationship R. For example, assume that ONT(M) contains the information that 'nejire' isa gene, and CCDB contains the relation 'CREB' isa protein (not shown in Fig. 2). Item (CON5a) shows declaration that states that 'nejire' bears the relationship *regulates* with 'CREB'. Since the relation *regulates* is known to the mediator, it translates the above mapping to enable any integrated view that accesses 'nejire' via the *regulates* relationship, to have access to 'CREB' in CCDB.

The concept-concept mapping allows a number of variations:
Often in the domain of scientific information, direct relationships between two concepts are not known. Assume for simplicity, that 'nejire' regulates exactly one unknown gene G, which in turn regulates 'CREB'. To express this, the $\mathcal{CSL}$ expression in (CON5a) will be modified to (CON5b), with an existential quantifier in the head.

If there were an *unknown number* of intermediate genes in the regulation path between 'nejire' and 'CREB', we would express this fact in $\mathcal{CSL}$ by placing 'CREB' in

$tc\_regulates$ of 'nejire', where $tc\_regulates$ is the transitive closure relation built on $regulates$ as in item (CON5c).

Subrelation Mapping. The "subrelation" mapping declares a relation in $\mathsf{ONT}(S)$ to be a special case of a relationship in $\mathsf{ONT}(M)$ (or vice versa). Consider that the mediator uses a relationship called has_part. CCDB uses more refined relationships $has(co)$ and $has(pm)$. Item (CON6) declares $has(co)$ to be a specialization of the mediator's has_part relationship. We omit the arguments of the relationships if the arguments of one relationship corresponds exactly to the positionally identical element of the second. The mediator processes this mapping by declaring $has(co)$ as one possible substitution of has_part for the source CCDB.

Concept-Instance Mapping. In the last section we showed how a **domain** declaration is used to connect a concept in the local ontology to instances of a local object class. Our idea there was to make the statement that the qualified instances of the object class were "semantically about" the concept. The concept-instance mapping is a similar idea to connect the instances of a local object class to a concept at the mediator. Let ultrastructure be a concept defined at the mediator. Let us also assume that has_part is a relationship defined at the mediator. We use item (CON7) to state that every instance of the class STRUCTURE in CCDB whose name has a value that can be found in the has_part tree of the mediator's ontology is "semantically about" the concept ultrastructure. So, if the mediator's ontology has the fragment:

$$\mathsf{dendrite} \overset{has\_part}{\longrightarrow} \mathsf{SER}$$

and CCDB had an object instance STRUCTURE(50, 'SER', 20.2, 45.5, ...), then this instance is "about" an ultrastructure.

Relation-Association Mapping. The "assoc_rel" mapping relates an inter-object association A in $\mathsf{OM}(S)$ to an inter-concept relationship in R in $\mathsf{ONT}(M)$. Let us assume $\mathsf{A}(\mathsf{X}_1, \mathsf{X}_2)$ and $R(Y_1, Y_2)$ are both binary. For the "assoc_rel" mapping to hold, X_1 and X_2 are *implicitly considered to be* "semantically about" Y_1 and Y_2 respectively. If $\mathsf{A}(\mathsf{a}_1, \mathsf{a}_2)$ is an instance of the association in the extension of $\mathsf{OM}(S)$, then one can construct a relation $R(a_1, a_2)$ at the mediator. Assume, *e.g.*, $\mathsf{ONT}(M)$ contains the spatial relationship $inside(s_3, s_4)$ meaning that structure s_3 is physically inside s_4. Now consider the association $\mathsf{surrounds}(s_1 : \text{STRUCTURE}, s_2 : \text{STRUCTURE})$ in $\mathsf{OM}(\mathsf{CCDB})$. In (CON8) of Fig. 3, we use the reserved word **inverse** to associate $\mathsf{surrounds}.s_1$ with $inside.s_4$ and $\mathsf{surrounds}.s_2$ with $inside.s_3$. If we find the instance $\mathsf{surrounds}(\text{'caudate\_putamen'}, \text{'fiber\_bundle'})$, the mediator can create a new relationship $inside(\mathsf{fiber\_bundle}, \mathsf{caudate\_putamen})$.

New Relationship Mapping. We repeat the $\mathcal{CSL}$ expression in item (CON5a):

map (concept_concept)($regulates$('nejire', 'CREB'))
  IF 'nejire':concepts(mediator), 'CREB':concepts(CCDB)

where the relationship $regulates$ is declared as part of a concept-concept mapping. $\mathcal{CSL}$ assumes that the name of the relationship is known to the mediator, otherwise allows one to declare unknown relationships via the reserved word **new**. For example, consider the statement of item (CON9a):

map (concept_concept)(**new** $evidence\_of(regulates)$(‘cfos′, ‘CREB′))
    IF ′cfos′:concepts(mediator), ′CREB′:concepts(CCDB)

where $evidence\_of(regulates)$ is a new relationship. Typically, the declaration of a new relationship will be accompanied by additional constraints that specify its properties.

Mapping Constraints. Constraints are specified in $\mathcal{CSL}$ using the "*holds*" mapping element. Item (CON9b) shows an axiom about the relation $evidence\_of(\_)$ for any parameter. The axiom assumes that the mediator knows the relation $opposes(X, Y)$ (*i.e.*, X contradicts Y), and states that no concept of CCDB can be an evidence of two opposing concepts of the mediator.

4 Registration Process

In the following, we outline how $\mathcal{CSL}$ specifications are handled by the mediator to complete the source registration process. The registration process involves the following steps:

- *Store*: At runtime, the mediator receives $\mathcal{CSL}$ statements sent by the source and stores them in a global registry.
- *Index*: Based on $\mathsf{CON}(S)$, the mediator updates $\mathsf{ONT}(M)$ to include new local concepts and relationships introduced by $\mathsf{ONT}(S)$. Then mediator updates its global concept index to keep track of which concepts have been used and referred to by the registered sources.
- *Assimilate Local Semantics*: The ontological grounding $\mathsf{ONTG}(S)$ and local integrity constraints of a source S are translated into an executable specification at the mediator. For example, the following statement from Fig. 2
    domain(STRUCTURE.volume) **in** [0,300]
  is translated into a logic rule encoding an integrity constraint in the form of a denial:
    false :– X:structure[volume→V], $\neg(0 \leq$V$\leq 300)$
  Similarly, the ontological grounding rule (OG1)
    domain(STRUCTURE.name) **in** $tc\_has(co)$(cerebellum)
  is translated into the logic rule
    false :– X:structure[name→N], ¬tc_has(co)(cerebellum)
- *Assimilate Context*: The contextualization $\mathsf{CON}(S)$ is assimilated at the mediator. Consider, for example, item (CON6) which states that the CCDB relation $has(co)$ is a "subrelation" of the mediator's relation has_part:
    map (subrelation)($has(co)$, has_part) (CON6)
      IF $has(co)$:relationships(CCDB), has_part:relationships(mediator)
  This is translated into the logic rules
    has_part(X,Y) :– CCDB.has(co)(X,Y) (*derive*)
    false :– CCDB.has(co)(X,Y), ¬has_part(X,Y) (*denial*)
  The first rule is used to *populate* and *query* the has_part relation at the mediator, while the second, logically equivalent rule specifies the integrity constraint as a denial and is used for *reasoning* about contextualizations.[8]

[8] This is similar to subsumption testing in description logics; the details of this are beyond the scope of this paper.

- IVD *extension*: The final step is to augment the view definitions $\mathsf{IVD}(M)$ to reflect model elements such as equivalent and subconcept. For example, the declaration (CON2) states that source and mediator concepts should be considered equivalent if they are syntactically equal; (CON3) states that what CCDB calls brain is a subconcept of what the mediator calls organ. Logically, this corresponds to extending the concept hierarchy by asserting the equivalence or subconcept relationship between the respective terms. Note that the logic view definitions $\mathsf{IVD}(M)$ do not have to be rewritten but can automatically access the newly asserted concepts (equivalent or subconcepts), provided that inheritance rules have been asserted.[9]

5 Conclusion

In this paper we have investigated the problem of source registration in the context of semantic information mediation. We have shown how a source can export its schema and information semantics to the mediator by specifying its *object model*, *local ontology*, and *ontological grounding* that relates the local ontology with elements of the object model. This explicit modeling of the source's semantics to facilitate mediation is a novel contribution of our work. Further, we have developed $\mathcal{CSL}$ a context specification language by which the source maps its local ontology in the context of the mediator's ontology. The language allows a mediation engineer to perform fine-grained mapping between the modeling constructs of the source and those of the mediator. We are currently in the process of implementing a more complete version of the language.

We have outlined how the mediator can interpret the source's declarations and internalize these definitions to complete the process of registration. However, there are several difficult and unresolved problems in assimilation. For example, how should the mediator deal with contradictions between its own ontological definitions and the source's local ontology? Also, how does the mediator's query engine evaluate the views that have been affected by the newly joining source? We plan to address these issues in the future.

References

[AFGP96] A. Artale, E. Franconi, N. Guarino, and L. Pazzi. Part-whole Relations in Object-Centered Systems: An Overview. *Data & Knowledge Engineering*, 20:347–383, 1996.

[BB01] D. Beneventano and S. Bergamaschi. Extensional Knowledge for semantic query optimization in a mediator based system. In *Int. Workshop on Foundations of Models for Info. Integ. (FMII-2001)*, 2001.

[BCG96] B. Benn, Y. Chen, and I. Gringer. A rule-based strategy for schema integration in a heterogeneous information environment, 1996.

[BCV99] S. Bergamaschi, S. Castano, and M. Vincini. Semantic Integration of Semistructured and Structured Data Sources. *SIGMOD Record*, 28(1):54–59, 1999.

[9] See [KLW95] for monotonic inheritance rules in F-Logic: the implementation language of our semantic mediation prototype.

[BGL+99] C. Baru, A. Gupta, B. Ludäscher, R. Marciano, Y. Papakonstantinou, P. Velikhov, and V. Chu. XML-Based Information Mediation with MIX. In *Intl. Conf. on Management of Data (SIGMOD)*, pp. 597–599, 1999.

[BHP00] P. A. Bernstein, A. Y. Halevy, and R. A. Pottinger. A vision for management of complex models. *SIGMOD Record*, 29(4):55–63, 2000.

[CCG+01] D. Calvanese, S. Castano, F. Guerra, D. Lembo, M. Melchiori, G. Terracina, D. Ursino, M. Vincini. Towards a Comprehensive Methodological Framework for Semantic Integration of Heterogeneous Data Sources. *Intl. Workshop on Knowledge Representation meets Databases (KRDB)*, 2001.

[CE93] P. N. Creasy and G. Ellis. A Conceptual Graph Approach to Conceptual Schema Integration. In *Conceptual Graphs for Knowledge Representation: ICCS*, pp. 126–141, Quebec, Canada, 1993.

[Cha00] H. Chalupsky. OntoMorph: A Translation System for Symbolic Knowledge. In *Principles of Knowledge Representation and Reasoning*, 2000.

[EJ95] L. Ekenberg and P. Johannesson. Conflictfreeness as a Basis for Schema Integration. In *Conference on Information Systems and Management of Data (CISMOD)*, pp. 1–13, 1995.

[FLM98] D. Florescu, A. Levy, and A. Mendelzon. Database Techniques for the World-Wide Web: A Survey. *SIGMOD Record*, 27(3), September 1998.

[GLM00] A. Gupta, B. Ludäscher, and M. E. Martone. Knowledge-Based Integration of Neuroscience Data Sources. In *Intl. Conference on Scientific and Statistical Database Management (SSDBM)*, 2000.

[GM02] J. Grant and J. Minker. A Logic-Based Approach to Data Integration. *Theory and Practice of Logic Programming (TPLP)*, 2(3):323–368, 2002.

[GMPQ+95] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. Ullman, and J. Widom. The TSIMMIS Approach to Mediation: Data Models and Languages. In *Next Generation Information Technologies and Systems*, 1995.

[GSN+01] C. Goble, R. Stevens, G. Ng, S. Bechhofer, N. Paton, P. Baker, M. Peim, and A. Brass. Transparent Access to Multiple Bioinformatics Information Sources. *IBM Systems Journal*, 40(2):534–551, 2001.

[Hal01] A. Y. Halevy. Answering Queries Using Views: A Survey. *VLDB Journal*, 10(4):270–294, 2001.

[HKWY97] L. M. Haas, D. Kossmann, E. L. Wimmers, and J. Yang. Optimizing Queries Across Diverse Data Sources. In *Intl. Conf. on Very Large Databases (VLDB)*, pp. 276–285, Athens, Greece, 1997.

[Hov97] E. Hovy. A Standard for Large Ontologies. In *Workshop on Research & Development Opportunities in Federal Information Services*, 1997.

[KLW95] M. Kifer, G. Lausen, and J. Wu. Logical Foundations of Object-Oriented and Frame-Based Languages. *Journal of the ACM*, 42(4):741–843, 1995.

[KMA+98] C. A. Knoblock, S. Minton, J. L. Ambite, P. J. M. N. Ashish, I. Muslea, A. G. Philpot, and S. Tejada. Modeling Web Sources for Information Integration. In *15^{th} National Conference on Artificial Intelligence*, 1998.

[LGM00] B. Ludäscher, A. Gupta, and M. E. Martone. Model-Based Information Integration in a Neuroscience Mediator System. In *Intl. Conf. on Very Large Data Bases (VLDB)*, pp. 639–642, Cairo, Egypt, 2000.

[LGM01] B. Ludäscher, A. Gupta, and M. E. Martone. Model-Based Mediation with Domain Maps. In *17th Intl. Conf. on Data Engineering (ICDE)*, Heidelberg, Germany, 2001.

[LRO96] A. Y. Levy, A. Rajaraman, and J. J. Ordille. Querying Heterogeneous Information Sources Using Source Descriptions. In *Intl. Conference on Very Large Data Bases (VLDB)*, pp. 251–262, 1996.

[MFRW00] D. L. McGuinness, R. Fikes, J. Rice, S. Wilder. The Chimaera Ontology Environment. *17^{th} Natl. Conf. on Artificial Intelligence (AAAI)*, 2000.

[MGW+02] M. E. Martone, A. Gupta, M. Wong, X. Qian, G. Sosinsky, S. Lamont, B. Ludäscher, and M. H. Ellisman. A Cell-Centered Database for Electron Tomographic Data. *Journal of Structural Biology*, 2002. to appear; see also `http://ncmir.ucsd.edu/CCDB/`.

[MWK00] P. Mitra, G. Wiederhold, and M. L. Kersten. A Graph-Oriented Model for Articulation of Ontology Interdependencies. In *Extending Database Technology*, pp. 86–100, 2000.

[NM00] N. F. Noy and M. A. Musen. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In *17th National Conference on Artificial Intelligence (AAAI)*, pp. 450–455, 2000.

[PFPG02] M. Peim, E. Franconi, N. Paton, and C. Goble. Query Processing with Description Logic Ontologies Over Object-Wrapped Databases. In *Intl. Conf. on Scientific and Statistical Database Management (SSDBM)*, 2002.

[PS98] C. Parent and S. Spaccapietra. Issues and Approaches of Database Integration. *Communications of the ACM*, 41(5):166–178, 1998.

[PTU00] L. Palopoli, G. Terracina, and D. Ursino. The System DIKE: Towards the Semi-Automatic Synthesis of Cooperative Information Systems and Data Warehouses. In *Proc. ADBIS-DASFAA Symposium*, pp. 108–117, 2000.

[RR97] V. Ramesh and S. Ram. Integrity Constraint Integration in Heterogeneous Databases: An Enhanced Methodology for Schema Integration. *Information Systems*, 22(8):423–446, 1997.

[RTU01] D. Rosaci, G. Terracina, and D. Ursino. A Semi-automatic Technique for Constructing a Global Representation of Information Sources Having Different Formats and Structure. In *DEXA*, pp. 734–743, 2001.

[SL90] A. P. Sheth and J. A. Larson. Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. *ACM Computing Surveys*, 22(3):183–236, 1990.

[SM01] G. Stumme and A. Maedche. FCA-MERGE: Bottom-Up Merging of Ontologies. In *IJCAI*, pp. 225–234, 2001.

[Tür99] C. Türker. *Semantic Integrity Constraints in Federated Database Schemata*. DISDBIS 63, infix-Verlag, 1999. Ph.D. thesis, Fakultät für Informatik, Universität Magdeburg.