Mediated Information Gain

Joachim Biskup
Universität Dortmund
D-44221 Dortmund, Germany
biskup@ls6.cs.uni-dortmund.de

David W. Embley
Brigham Young University
Provo, Utah, USA
embley@cs.byu.edu

Abstract

We deal with the problem how to gain information from many heterogeneous data sources. As a solution we propose a general framework for mediated information gain: it minimizes and supports the human burden, clearly identifies the algorithmically tractable subtasks, and assures a well-understood level of quality. We identify a "mediation task" as an extended amalgamation of searching, querying and updating in traditional information systems, and we introduce the notion of a "formal information state" which unifies and extends the classical notions of a schema, an instance, a query with additional specifications and a returned answer. The quality is concerned with correctness, precision and completeness. Correctness is guaranteed by maintaining internal consistency. Precision is achieved by embeddings identifying only relevant parts of the sources. And completeness is reached by user interactions with revisions and by iteration. The whole framework is characterized by eighteen rules which capture the essentials of mediated information gain and suggest a high-level architecture for implementations.

1. Introduction: Rules for an Established Procedure are Wanted

How do users gain information from many heterogeneous data sources? Using the worldwide "information infrastructure" with "data super-highways" and a potentially unlimited number of "providers," any member of the evolving "information society" is faced with this challenge. Surely, we should not expect to see anything like a global answer, a universal "information supply program," or an always-applicable "information methodology." There is a simple reason for this position. The "information society" is characterized by the perspective that digitally available "information" and "data" are becoming as manifold and rich as "society" itself. The variety of life is increasingly reflected in digitized representations, and, additionally, digitized items are considered as an integral part of life in their own right, not just as an image of an "outside world." Hence, we had better refrain from dreaming of "omnipotent technologies," and avoid thinking of an "all-encompassing view" of information.

Nevertheless, the challenge remains: how do users obtain information in a concrete situation? There are many different situations with different individuals involved having their specific thoughts and feelings, interests and goals, traditions and trainings. For instance, sometimes a person may wish to perform "net-based brain storming" by browsing the World-Wide Web following sudden inspirations and spontaneous associations—why not? In other occasions, however, a person may want to follow an established procedure for *mediated information gain* which

- minimizes and supports the human burden, as far as actually possible,
- clearly identifies the algorithmically tractable subtasks and executes them automatically, and
- assures a well-understood level of quality.

The rules of such a procedure and the overall architecture of implementing mediating computing systems are the topic of this paper. The main contributions can be summarized as follows:

- We identify a *mediation task* as an extended amalgamation of searching for matches (like we do in information retrieval) and logic-oriented querying and constraint preserving updating (like we do in data and knowledge bases).
- We introduce the notion of a *formal information state* which unifies and extends the classical notions of a schema, an instance, as well as a query with additional specifications and a returned answer.
- We emphasize *internal consistency* of formal information states as the main criteria for correctness.

- We identify *embeddings* as the basic tool for homomorphic reduction of source data and for merging it with target data.
- We clarify the *role for user interaction* in revising formal information states.
- Finally, we characterize the scenario and the overall procedure by stating eighteen rules which capture the essentials of mediated information gain and suggest a *high-level architecture* for implementations.

The rules to be discussed are rooted in some basic human philosophical quests. First, recall fundamental questions in philosophy: how can we "recognize" and "mentally represent" what we are "experiencing," how can we assign "meaning" to sensations, how can we claim to "know" anything? Second, recall that computer science has dealt with the topic right from the beginning, resulting in specialized and overlapping research areas such as information systems, knowledge engineering, artificial intelligence, and computational intelligence. Finally, recall that with the advent of distributed interoperable computing, our ability to dispense and manipulate information globally has exploded dramatically. It is far beyond the scope of this paper to survey all these roots and report on the specific research and development that has been done by many scientists around the world. Thus, only selected recent technical contributions in mediation will be mentioned.

The rest of this paper is structured as follows. Section 2 briefly reviews selected previous work in order to identify the basic features. Section 3 sketches an anticipated scenario, explicates fundamental terms, presents and justifies the rules for "mediated information gain," and derives the high-level architecture for implementing mediating computing systems. Section 4 discusses some selected refinements and more advanced features.

2. Basic Features of Mediated Information Gain

In this section we review some previous work in order to identify some basic features. Moreover, this review supplies us with a collection of examples that can be used to illustrate and to substantiate the more abstract considerations in subsequent sections.

Basically, there are two traditional streams of research and development for information systems, data- and knowledge-base technology and information retrieval. Data- and knowledge-base technology is based on schema declaration, strongly formatted instances satisfying declared semantic constraints under updates, and declarative well-typed query languages implementing the notion of formal implication for fragments of first order logic. Information retrieval is based on search key declarations, semi-structured instances with explicitly or implicitly provided descriptors, and query languages implementing the search for instance items, the descriptors of which match logical combinations of key values. Search machines for the Web can be seen as a recent variant. Surely, there is a substantial overlap between the two streams, and many researchers advocate a unified view. See for instance [13, 14].

Mediated information gain takes ideas from both streams. ¿From database technologies, it adapts the declarative style of updates and queries, and from information retrieval, it adapts the search for matches between key values and descriptors. More precisely, we are interested here in the following features:

- A database query is explicitly expressed in terms of the identifiers declared in the database schema, and it has itself a *well-defined target schema*, which can also be understood as a view schema. Thus a query result can be *further processed* like persistently stored database items. The semantics of queries are based on deriving *logical implications* from the actual database instance.
- A database update is explicitly expressed in terms of the declared database schema and of extensional data to be inserted, deleted or used for updates. The *semantic constraints* declared in the schema are an invariant to be maintained under updates.
- An information retrieval request is explicitly expressed in terms of *search key values* supposed to be appropriate for the considered *sources*. The semantics of requests are based on a declared notion of *matches* between key values and descriptors.

In Section 3 we will present our notion of a task for mediated information gain, a *mediation task* for short. This notion amalgamates the itemized features of data- and knowledge-base technology and information retrieval in a specific way, and it substantially extends them. The amalgamation is achieved by an iterative process that converts an initial formal information state into a final result state, as fully explained in Section 3:

- A mediation task starts with an initial state that is a self-description of the desired result comprising, among others, a *well-defined target schema* including *semantic constraints* as well as *search key values*.
- A mediation task exploits possibly heterogeneous sources for generalized *matches* between parts of the self-description and parts of the source states. Such generalized matches are called *embeddings* in order to emphasize two points: the asymmetry between the mediation task and the sources, and the unavoidable

coarsening of information when data flows from the sources to the target. See for instance [9, 17].

- A mediation task reduces *logical implications* derived from matching parts of the sources such that the implications can be *combined* with the task's current state.

- A mediation task maintains a part of its initial self-description, the *semantic constraints* in particular, as invariant.

Summarizing, a *mediation task* can be understood as an extended amalgamation of searching for matches, logic-oriented querying and constraint preserving updating.

Starting with the seminal proposal [23], a lot of work has already been been done in mediated information systems, see for instance [1, 2, 5, 8, 16, 18, 20, 21, 24, 25]. In the following we use the pioneering and most advanced project, TSIMMIS [15, 19, 22], and our own developments DE [10, 11], MMM [3, 4, 6] and IEI [7] to discuss some of the achievements so far and to exemplify our innovative approach to mediated information gain.

In any approach to gain information we have to deal with three aspects of information. (1) Users have to syntactically provide some information about their needs. (2) The underlying information system manages some kind of stored information. (3) A returned query answer constitutes information gain. We unify all three aspects and introduce for them the generic term *formal information state* (fis). Then, *gaining information* means to appropriately relate the occurring formal information states according to the specific purposes under consideration.

- In classical database technology, we have the following kinds of formal information states: a database is a formal information state comprising the intensional schema and the extensional instance; a query is a formal information state specifying the view schema; a returned query answer transforms the formal information state of the query by adding extensional data taken from the database instance. Afterwards, the query answer is a formal information state which comprises a schema and data, similar to a stored database.

- TSIMMIS generally refrains from separating schemas and instances. As a consequence, a TSIMMIS database is a formal information state that gathers a collection of heterogeneous data items each of which carries its individual schema-like self-description. A query is a formal information state specifying structural properties of the desired data, mostly expressed by path expressions. A returned query answer is a formal information state that again has heterogeneous self-describing data items each of which matches the specified structural properties. Therefore the returned formal information state can be used like the stored database. In order to prepare for retrieving matching data from heterogeneous sources, specific wrappers are constructed in advance to formally capture the potential for matches on a schema-like level.

- Seen from the perspective of a user, the MMM maintains a formal information state which is an application-oriented database schema without a persistent instance. A user query is a formal information state as for classical databases. Basically, query evaluation results in adding a partial instance to the formal information state of the application-oriented schema. This partial instance is dynamically generated from the instances of connected sources. For each source a human administrator has identified and declared a "type embedding" between items of the application-oriented database schema and items of the virtual or existing schemas of the sources. More generally, a type embedding is among tree like structures built from atomic types using several type constructors. These type embeddings are automatically exploited for decomposing and translating a global query addressed to the mediator into a set of local queries to the sources. Thus any source is interpreted as a formal information state consisting of a schema (used to identify type embeddings at source connection time) and an instance (used to generate partial instances at user query time). Wrappers serve for resolving heterogeneity.

- DE and IEI are initialized with a target formal information state denoted as an "ontologically specified target view" which comprises an OSM schema [12] (for types, object sets, relationship sets, semantic constraints,...) and additional specifications for expected or possible data values, context keywords, a type hierarchy and coercion routines, or unit conversion routines. The primary goal is to populate such a formal information state with data from heterogeneous sources. For DE, the objective is to automatically extract data from unstructured sources. For IEI, any source is seen as a formal information state comprising again an OSM schema and, additionally, a population. Then a suitable "embedding function" between the target formal information state and the schema part of the source is semi-algorithmically constructed, by interaction with a human administrator. Such an embedding function has to satisfy several formal requirements that serve two purposes: (1) they homomorphically reduce data of the source to a task conforming representation and load the reduced data into the target; and (2) they guarantee that the loaded data conforms to all semantic constraints of the target. Moreover, data from several sources is appropriately merged with the ultimate goal of obtaining a constraint satisfying result.

Besides the basic features of formal information states and embeddings and the corresponding data extractions, there are additional important features: (1) *revision* of the target formal information state, (2) *control* of the overall mediation task, (3) the user's *confidence* for situations with incomplete or vague specifications and with ambiguities, and (4) the assured *quality* of the final result. We only give some brief guidelines:

- *Revision* is well-known from information retrieval where usually a user interactively determines a key-value expression that yields a manageable amount of information. IEI proposes revision of the target formal information state by weakening semantic constraints if sources appear to be intuitively appropriate but cannot otherwise deliver suitably restricted data.

- *Control* in traditional approaches as well as in TSIMMIS and MMM is split in a precomputation phase performed by a human administrator, and a subsequent phase for fully automatically evaluating user queries or updates. IEI attempts to partially automate and to extend the precomputation phase. It uses a built-in search strategy with user interaction for algorithmically determining the pertinent embeddings and for subsequent data loading.

- The user's *confidence* is a minor problem in databases, relating mostly to null values and assumptions concerning the use of negation operators. However, in information retrieval the user's confidence is a major topic, since descriptors have always been subject to imprecision, incompleteness and related problems. IEI organizes the search for an embedding with the required formal properties by tentatively mapping single task items to single source items and memorizing annotations about the issues considered. At the end, the tentative single mappings have to be combined into a consistent embedding function. The decisions are taken according to a ranking on the annotations. In a still rudimentary approach, this ranking is interpreted to reflect the user's confidence.

- Assured *quality* is solved for databases only once by proving the query language to be correct and complete with respect to the underlying notion of logical implication. However, in information retrieval, quality in terms of recall and precision ratios is the most crucial concern for individual queries. For TSIMMIS, the quality of a query answer essentially relies on the query containing just the attribute identifiers known in the sources, or on a preceding user-driven exploration of these identifiers. For MMM, the quality is determined by the type embeddings identified by the administrator in advance. For DE and IEI, the quality is first of all dependent from the initial "ontologically specified target view" formally capturing what the user has in mind. In particular the discriminating power of the target formal information state with respect to the considered sources is crucial. The experiments conducted with DE show very promising outcomes with respect to recall and precision ratios. Finally, IEI lets users specify semantic constraints for the initial formal information state. This allows satisfaction of semantic constraints to be an important quality factor. Indeed, this is one of the primary goals of IEI, and it is shown to be achievable.

3. The Rules and the Architecture

We now propose a list rules for mediated information gain from which we will derive a more or less canonical architecture. Before doing so, we sketch an anticipated scenario thereby explicating the terms in the desired goal.

3.1. Scenario and Explication of Terms

An individual (or a group of individuals), called a *user*, is willing or entitled to perform a specific informational activity in which there is a need to gain and use information from a variety of sources. For this purpose a user sets up a mediation task in the following restricted sense: the task is *initialized* with a formal information state that captures the user's informational *requirements*; and the task finally results in a formal information state that (hopefully) *implements* the user's informational needs. In most cases, the final state is an appropriate enrichment of the initial one. The final information state is considered to be worthwhile if it allows the user to perform the specific informational activity.

The formal information states act as a strict boundary in the following ways: (1) the informational activities can only be achieved by means of the states; (2) anything beyond the states is seen to be inaccessible or even nonexistent; and (3) any specific detail of interest for the activity must be somehow included in the formal information states. These limiting properties, however, hold only for the *current* information state. The current formal information state may be enriched by a mediation task, and it can be *revised* by the user.

The information state is *formal* in the sense that it is only syntactic. Surely, since the activities involve interactions with external components, say people or mechanical devices operating in real space and time, the formally constructed information state should have a suitable correspondence to the external concepts. Within our present framework, however, this problem is *not* our primary concern.

Some parts of a formal information state are declarative or *intensional* in nature, i.e., they are interpreted procedurally. Other parts of the state are purely *extensional* in nature, i.e., they are used as rough atomic data. As usual, this distinction is strongly context dependent. Typically, within our framework, intensionally understood parts of a formal information state originate from the user, and extensionally understood parts originate in data sources. If used this way, intensional parts can be interpreted as a specification for a desired extensional part, and the intensional part can guide the automatic extraction of the extensional part. However, we may also employ the two parts the other way round: given extensionally understood parts, whether originating from the user or from data sources, can be examined for intensional content. If used this way, the extensional part is interpreted as a specification for a desired intensional part.

To construct a formal information state, we need *syntactic material* as input. In all cases some *initial input* must necessarily be supplied by a user: only a user knows the informational activities, and only a user can judge the appropriateness with respect to external concepts. In most cases the initial input has been and will be intensional, and it typically comprises something like a *schema*.

Occasionally—in fact, traditionally as a rule—the data is also added by the user. Typically, this data somehow reflects an external concept. For instance the birth of a child is reflected by several data items including family name, given name, date and location of birth, sex, names of the mother and the father. Adding new data, however, is usually monitored by already available data by checking semantic constraints, understood as intensional, against the enlarged extensional part. In case of a violation, semantic constraints can sometimes be exploited to somehow automatically "correct" enlarged extensional part. Surely, a user should be careful in allowing such facilities, in particular when starting with external concepts. Users need confidence that an automatic "correction" does not blur assumed correspondences with external concepts.

Now, the new challenge of the information society potentially puts users in a *completely new situation*: they are connected to a network of sources that actually appear as formal information states. Thus they can no longer mentally convert external concepts into data parts of a state using human insight. Instead they now have to *gain information* solely from the sources, relying in the extreme only on their specific initial information state, and the formal information states of the sources. Indeed, this situation entails a strong restriction on the one hand, but it also offers fascinating undreamed opportunities on the other hand—users can be assisted or even partially replaced by computing systems. Our rules explore this extreme scenario in order to understand how to design the desired computing systems, which are called *mediators*.

3.2. Summarized Scenario

The first set of rules summarizes and extends the key features of the scenario as presented above. The rules are roughly visualized by Figure 1.

Rule 1 (Information as a formal information state) *Information is defined as a formal information state that is considered worthwhile for a well-defined activity by a specific user.*

Rule 2 (Principle of syntax) *A formal information state is built from syntactic material.*

Rule 3 (Overloading) *Any syntactic material can be understood as intensional or as extensional.*

Rule 4 (Correctness as consistency) *A correct formal information state should satisfy a formal notion of (internal) consistency which can be algorithmically checked.*

Rule 5 (Closedness) *Any syntactic material is either given by the user or extracted from other formal information states.*

Rule 6 (User-system interaction) *Building a formal information state is an interactive process: initialization and revision are done by a user who has the informational activity in mind; data extraction is performed purely algorithmically based on an already available partial formal information state called the target.*

Rule 7 (Consistency as invariant) *The main invariant of the interactive process is the consistency of the formal information state under construction.*

Rule 8 (Completeness and precision as a goal) *The goal of the interactive process is to achieve a certain degree of completeness (all relevant parts) and precision (only relevant parts).*

Rule 9 (Role of the mediating system) *A mediating computing system manages a formal information state as a target, controls the interaction with the user, and executes the algorithmic extraction from accessible sources, which are formal information states.*

3.3. Outline of Procedure

Roughly outlined, with reference to Figure 2, the procedure for mediated information gain starts with a user-specified target, which is a formal information state fis_{init}, and then the procedure iteratively approximates a specification for the desired information:

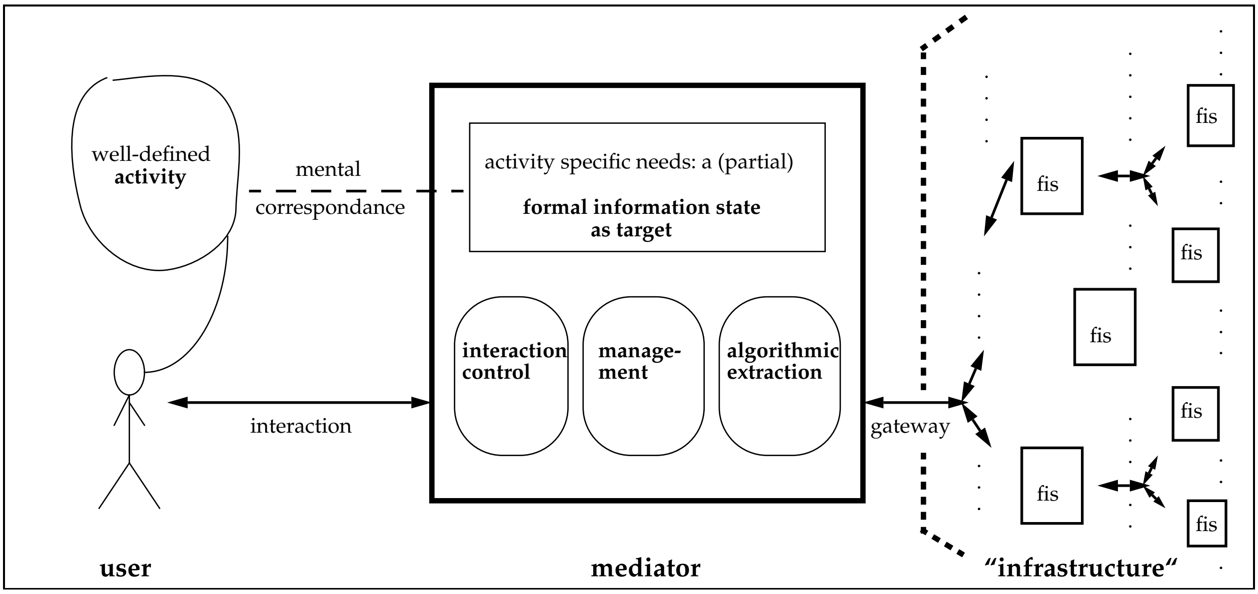

Figure 1. Visualization of the scenario with a specific user, a mediator, and an “information infrastructure” appearing as a collection of formal information states (fis)

initialize the target $fis_{current}$ as fis_{init};
control
  select an accessible fis_{source} from the infrastructure;
  exploit fis_{source} with respect to $fis_{current}$:
    `if` an *embedding* is found
    `then` **reduce** fis_{source} and
      combine *consistently* with $fis_{current}$
      yielding a new candidate fis_i
    `else` determine *violations* and
      interactively **locally revise**
      the current target $fis_{current}$
||
  merge candidates fis_i's:
    `if` the fis_i's can be *consistently merged*
    `then` replace $fis_{current}$ by the
      consistent merge of all fis_i's
    `else` determine *violations* and
      interactively **globally revise** the
      current target $fis_{current}$;
until desired information has been
      sufficiently approximated by $fis_{current}$;
return the final $fis_{current}$.

The main requirements are roughly met as follows, although the details are more extensive and subtle [7]:

- *Correctness* is guaranteed by maintaining internal *consistency* as an invariant.
- *Precision* is achieved by limiting *embeddings* to only relevant parts of the sources.
- *Completeness* is achieved by appropriately allowing users to interactively *control* the procedure and to *revise* the current target.

3.4. Rules for the high-level architecture

The procedure outlined in Section 3.3 is described in more detail by the following rules and the additional explanations. These rules suggest the basic conceptual components and the high-level architecture of a mediator, which we show visually in Figure 2.

Rule 10 (User driven initialization) *In the initialization step, the user tentatively specifies a (partial) formal information state fis_{init} as a target which presumably captures the user's current mental understanding of the information needs.*

Rule 10 suggests that the overall procedure works declaratively and top-down. Accordingly, the expressiveness of the declaration language for formal information states essentially determines the potential power of the procedure: a computing system can only process what the user can express.

Rule 11 (Main algorithmic phases) *Algorithmic extraction works in four main phases: (1) exploitation, (2) source reduction, (3) target revision, and (4) candidate merging. These phases are suitably repeated and interleaved, and even backtracking is possible.*

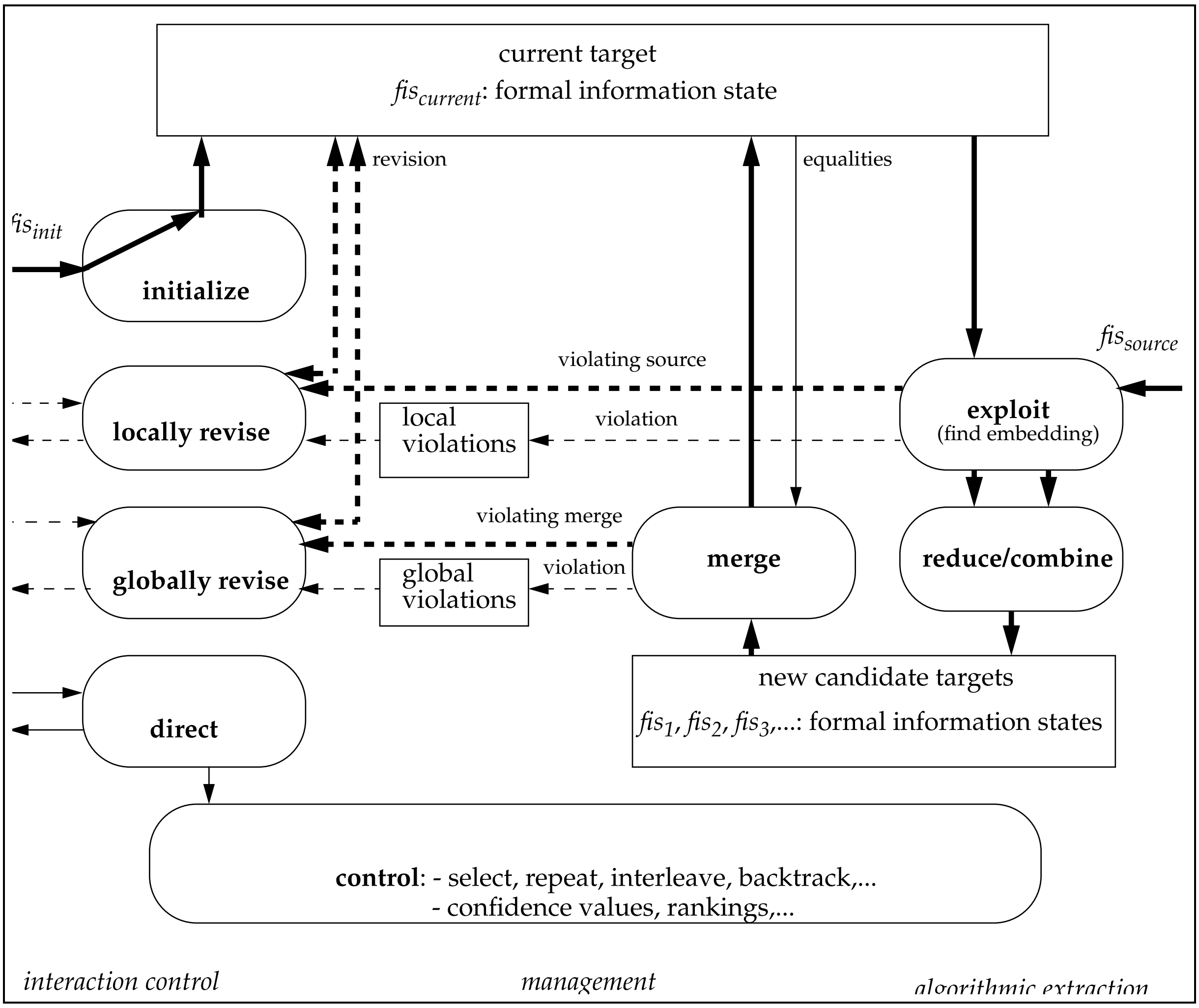

Figure 2. Basic components and high level architecture of a mediator: bold arrows indicate major flows of formal information states; thin arrows denote major other flows; dotted cases apply only in case of violations

As usual, the overall power of a computing system is the result of combining basic functionalities and control structures. The basic functionalities are considered specific for our topic. We argue that the control structures can be adapted from well-understood techniques in computer science, though, in fact, the final performance of actual implementations is expected to heavily depend on clever adaptations.

Rule 12 (Intensional exploitation) *In the exploitation phase, the current target $fis_{current}$ is used purely intensionally: $fis_{current}$, or any suitable part of it, is understood as a declarative specification of the set of source formal information states such that $fis_{current}$ can be embedded in each of them.*

Intuitively, exploitation searches for relevant data contained in sources. The activity-specific notion of relevance guiding the search must be syntactically declared. The current formal information state serves as the intensional declaration and is understood as a search parameter.

Rule 13 (Embeddings define homomorphic reductions) *Embedding of one formal information state fis_a in another one fis_b essentially means that there exists a homomorphic reduction of fis_b to fis_a: any informational activity (e.g., a query) in fis_a can be simulated in fis_b (but not necessarily conversely) and, additionally, fis_a can be consistently combined with the homomorphic reduction of fis_b.*

When searching for relevant data in sources, embeddings formally define the correspondences between a source (fis_{source}) and the target ($fis_{current}$). Intuitively, a correspondence implies that some informational activity (e.g., a query) could be performed directly in the source. In order to ensure this property, the source must be rich enough to deliver any desired information, and we must know how to discard details that are not needed and how to convert source-specific representations to target representations.

Rule 14 (Exploitation) *In an exploitation phase, any accessible fis_{source} is checked to see whether the target $fis_{current}$, or any suitable part of it, can be embedded in fis_{source}: basically every check returns either an embedding function and the corresponding homomorphic reduction mapping $RED_{source\_to\_current}$ or one or more violating subparts of $fis_{current}$.*

If we find a correspondence, we have to derive an executable specification stating how to algorithmically simulate performing the informational activity in the source.

If we cannot, we have inspected a source that does not qualify for simulation as it stands. Since sources are supposed to be autonomous, the user is not authorized to change any feature of a source. Therefore, if the user sees an advantage to including a source (or part of a source) for which there is not a derivable specification for simulating an informational activity, the user must revise the target specification based on a detected inadequacy of the target.

Rule 15 (Source reduction) *In a follow-up source-reduction phase, a homomorphic reduction mapping $RED_{source\_to\_current}$ is used to extract the corresponding part of fis_{source}, to reduce the corresponding part to the abstraction layer of $fis_{current}$, and to tentatively combine the reduction obtained with $fis_{current}$ to a new candidate target fis_i.*

Relevant parts of sources have to be made accessible. Conceptually, as literally stated by Rule 15, we can always achieve accessibility by producing a copy and managing the copy by the mediating system under the control of the user. However, we can also explore the option of managing only references to the original source version. In any case, at this stage of the procedure, the new candidate target still remains tentative because only a single source has been investigated in isolation, whereas the overall goal is to merge the best of all sources.

Rule 16 (Local target revision) *In a follow-up local-target-revision phase, a collection of violating subparts of $fis_{current}$ is presented to the user offering options for interaction: for each case either rejection or some minimal revisions of the target $fis_{current}$ such that the violation disappears. (We could also devise more advanced options that treat several violations as a whole.) Then the user selects any of the options which presumably captures the user's improved mental understanding of the information needs of the informational activity.*

An inspected source might suggest a target revision. The goal of a revision is to remove a detected inadequacy between the target and the source. The basic approach is to minimally weaken the requirements of the target such that, afterwards, the source qualifies as being relevant. However, the user could also investigate the violating source more deeply in order to learn how to better specify the target. This possibility is further discussed in Section 4.4.

Rule 17 (Candidate merging) *In a candidate merging phase, several previously generated new candidate targets are checked to see whether they can be consistently merged. Basically every check either replaces the current target by merging examined new candidate targets, or it returns one or more violating subparts of the merge.*

Intuitively, the target should specify which sources can be put together to serve the specific informational activity. Thus we have to investigate both the relevance of each individual source and the possible mixtures of relevant sources. We propose to layer the investigation, at least conceptually: we first treat sources in isolation; subsequently, we try to merge them. Merging means to take the union while using specified equalities among identified items. If no equalities are specified in the target, then the merge is just the set theoretic union. The impact of equalities is further discussed in Section 4.3. Although all contributing candidates have already been evaluated as being relevant during the exploitation phase, the resulting merge might still happen to be inadequate. We argue that, at this stage of the procedure, inadequacy means (internal) inconsistency. This appears to be a consequence of Rule 12 which considers the embedding relationship between the target and each single source but does not state any requirement on relationships among the sources, except possibly the equalities.

Rule 18 (Global target revision) *In a follow-up global-target-revision phase, a collection of violating subparts of a merge is presented to the user offering options for interaction: for each case either rejection or a collection of minimal revisions of the merge such that the violation disappears. Then the user selects any of the options which presumably captures the user's improved mental understanding of the information needs of the informational activity.*

An inconsistent merge of several new candidate targets demands a revision. The goal of the revision is to remove a detected inconsistency. The basic approach is to minimally change the merge.

4. Some Refinements and Advanced Features

In this section we briefly give some ideas about how to refine or to add more advanced features.

4.1. Confidence values

The notion of an embedding is *formal*, and it is intended to ensure desirable formal properties of a source reduction. There is the *hope* that the formal properties relate to worthwhile external concepts, but there is no guarantee. For any concrete case we can try to gather evidence for such a hope and to somehow rate the evidence. Moreover, the formal information states under considerations might be incomplete or subject to uncertainties. We also need ratings for these situations. Thus, when trying to make final decisions whether two formal information states are related by an embedding, we cannot be completely sure whether the external concepts are treated appropriately. Based on the gathered and rated evidence, however, we can calculate some confidence value, and we can compare confidence values for several candidates in order to select one that seems to be most promising. Surely, there is still only a hope that internal concepts correspond to external concepts, but we may be able to approximate some kind of guarantee.

4.2. Content-based embeddings

Since everything is built from syntactic material, a user has to convert any mental understanding of external concepts into syntactic form. This restriction also applies for expressing a suitable notion of embeddings. A first approach basically relies on a manifold type system with semantic constraints and activity-specific value sets. This approach can be easily enriched by adding additional constraints, such as mandatory keywords or related features. A more advanced application, however, demands more sophisticated notions that capture an intuition of "content" which is beyond simple structural properties and explicit occurrences of some strings. In particular, this problem arises in multimedia applications, where the "content" is only loosely coupled with the formal structure, if at all, and we have to analyse images rather than text.

4.3. Equalities

Dealing with several autonomous sources requires a recognition of which syntactic items represent which external concepts. There are two opposite problems. First, do two different syntactic items stemming from different sources represent the same external concept? Second, do two occurrences of the same syntactic item in two different sources represent two different external concepts? Both questions are likely to be difficult to answer. Clearly, there is a simple default—that syntactic equality is equivalent to conceptual equality. Unfortunately, however, in real applications this default is sometimes misleading. So we need specialized ways to resolve both problems.

4.4. Revision by data mining

It is commonly thought to be good practice to layer a formal information state into (at least) an intensional part, called a *schema*, and an extensional part, called an *instance*. Then the appropriate notion of (internal) consistency expresses that the (extensional) instance satisfies all the constraints and requirements of the (intensional) schema. If an inconsistent state arises, then the intension and the extension do not correspond. An intension/extension conflict can be resolved in two completely different ways, and in any of their combinations of course. First, the user could trust the intension, since it presumably captures the *a priori mental* understanding of the informational activity, and believe that the violation is in the extension. Second, a user could trust the extension as some sort of empirical data underlying the activity, and try to "mine" for an appropriate intension, i.e., try to find meaningful properties of the extension in order to use them as *posteriori empirically* found semantic constraints.

4.5. Materialization and refreshment

For some applications it appears reasonable to build and maintain the final formal information state over a long period of time. In this case the results of earlier steps can be *materialized*. However, then we somehow have to be informed about updates in the sources and to *refresh* the materialization appropriately.

4.6. Privacy protection

Privacy, understood as informational self-determination, basically requires that individuals keep control over their personal data. In particular, individuals should be able to control the linking of several parts of their personal data where each part has been purposely revealed to a different partner. Surely, in "real life" unlinkability cannot always be achieved, if some adversary or an authority is investigating a single case. However, unlinkability should be the rule, rather than the exception, and many statutes explicitly support unlinkability by obligating and protecting partners like lawyers, physicians, or journalists, not to reveal confidential personal data about others. Since information gain and data extraction are purposely designed to link data from several sources, there could obviously be a potential conflict between the functionality of a mediating computing system and the right of privacy. Thus, appropriate countermeasures must be taken.

5. Conclusions

People want to know. Computer science has to provide tools for assisting them. In this paper we have analyzed and structured a class of potential tools devoted to activity-oriented mediated information gain from heterogeneous sources. This class is characterized by eighteen rules suggesting a high-level architecture for implementation. Existing systems can be seen as partial implementations. Our own design [7] would deliver a specific realization leaving room for many further enhancements. Thus, for the time being, we have mainly proposed a framework for future research and development, and we identified the basic features and most important issues.

References

[1] S. Abiteboul, S. Cluet, V. Christophides, T. Milo, G. Moerkotte, J. Simon. Querying documents in object databases. *Journal of Digital Libraries*, (1997) 1:1, pp. 5 - 19.

[2] S. Abiteboul, D. Quass, J. McHugh, J. Widom, J.L. Wiener. The Lorel query language for semistructured data. *Journal of Digital Libraries* (1997) 1:1, pp. 68 - 88.

[3] C. Altenschmidt, J. Biskup. Entwurf und Realisierung eines Multimedia-Mediators (in German). *Proceedings NRW-Verbund "Die Virtuelle Wissensfabrik"*, Birlinghoven, 1999.

[4] C. Altenschmidt, J. Biskup, J. Freitag, B. Sprick. Weakly constraining multimedia types based on a type embedding ordering. *Proceedings 4th Workshop on Multimedia Information Systems*, MIS 98, Istanbul, Sept. 1998, Lecture Notes in Computer Science 1508, Springer, Berlin etc., 1998, pp. 121 - 129.

[5] Y. Arens, C.Y. Chee, C.-N. Hsu, C.A. Knoblock. Retrieving and integrating data from multiple information sources. *International Journal of Intelligent and Cooperative Information Systems* (1993) 2:2, pp. 127 - 158.

[6] J. Biskup, J. Freitag, Y. Karabulut, B. Sprick. Query evaluation in an object-oriented multimedia mediator. *Proceedings 4th Int. Conference on Object-Oriented Information Systems*, Brisbane, Australia, Nov. 1997, Springer, London etc., 1997, pp. 31 - 43.

[7] J. Biskup, D.W. Embley. Extracting information from heterogeneous sources using ontologically specified target views. submitted for publication, available under http://ls6-www.informatik.uni-dortmund.de/publications/1999.html.en, 1999.

[8] P. Buneman, S. Davidson, M. Fernandez, D. Suciu. Adding structure to unstructured data. *Proceedings 6th International Conference on Database Theory*, ICDT 97, Delphi, Greece, Lecture Notes in Computer Science 1186, Springer, Berlin etc., 1997, pp. 336 - 350.

[9] K.B. Bruce, G. Longo. A modest model of records, inheritance, and bounded quantification. In [17], pp. 151 - 195.

[10] D.W. Embley, D.M. Campbell, Y.S. Jiang, S.W. Little, D.W. Lonsdale, Y.-K. Ng, R.D. Smith. Conceptual-model-based data extraction from multiple-record Web pages. *Data & Knowledge Engineering* (1999) 31:3, pp. 227 - 251.

[11] D.W. Embley, D.M. Campbell, S.W. Little, R.D. Smith. Ontology-based extraction and structuring of information from data-rich unstructured documents. *Proceedings of the 7th International Conference on Information and Knowledge Management*, CIKM 98, Washington D.C., Nov. 1998, pp. 52 - 59.

[12] D.W. Embley. *Object Database Development: Concepts and Principles*. Addison Wesley, Reading, Massachusetts, 1998.

[13] N. Fuhr. Object-oriented and database concepts for the design of networked information retrieval systems. *Proceedings of the 5th International Conference on Information and Knowledge Management*, 1996, pp. 164 - 172.

[14] N. Fuhr. Models for integrated information retrieval and database systems. *IEEE Data Engineering Bulletin* (1996) 19:1, pp. 3 - 13.

[15] H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, J. Widom. Integrating and accessing heterogeneous information sources in TSIMMIS. *Proceedings of the AAAI Symposium on Information Gathering*, Stanford, March 1995, pp. 61 - 64.

[16] M.R. Genereseth, A.M. Keller, O.M. Duschka. Infomaster: an information integration system. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Tuscan, Arizona, May 1997, pp. 195 - 206.

[17] C.A. Gunter, J.C. Mitchell (eds.). *Theoretical Aspects of Object-Oriented Programming*. MIT Press, Cambridge -London, 1994.

[18] R. Hull. Managing semantic heterogeneity in databases: a theoretical perspective. *Proceedings of the 16th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, Tuscan, Arizona, May 1997, pp. 51 - 61.

[19] Y. Papakonstantinou, H. Garcia-Molina, J. Widom. Object exchange across heterogeneous information sources. *Proceedings 11th International Conference on Data Engineering*, Taipeh, Taiwan, March 1995, pp. 251 - 260.

[20] V.S. Subrahmanian, S. Adali, A.Brink, R. Emery, J.L.Lu, A.Rajput, T.L. Rogers, R.Ross, C.Ward. HERMES: a heterogeneous reasoning and mediator system. Submitted for publication, 1994 (see http//www.cs.umd.edu/hermes).

[21] A. Tomasic, L. Raschid, P. Valduriez. Scaling heterogeneous databases and the design of DISCO. *Proceedings of the 16th International Conference on Distributed Computing Systems*, May 1996, pp. 449 - 457.

[22] J.D. Ullman. Information integration using logical views. *Proceedings of the 6th International Conference on Database Theory*, ICDT 97, Delphi, Greece, Lecture Notes in Computer Science 1186, Springer, Berlin etc., 1997, pp. 19 - 40.

[23] G. Wiederhold. Mediators in the architecture of future information systems. *IEEE Computer* (1992) 25:3, pp. 38 - 49.

[24] G. Wiederhold, M. Genesereth. The conceptual basis for mediation. *IEEE Expert, Intelligent Systems and their Applications* (1997) 12:5, pp. 38-47.

[25] L.-L. Yan, M.T. Ozsu, L. Lin. Accessing heterogeneous data through homogenization and integration mediators. *Proceedings of the 2nd IFCIS International Conference on Cooperative Information Systems*, Kiawah Island, South Carolina, June 1997, pp. 130 - 139.