Transparent access to multiple bioinformatics information sources

by C. A. Goble
R. Stevens
G. Ng
S. Bechhofer
N. W. Paton
P. G. Baker
M. Peim
A. Brass

This paper describes the Transparent Access to Multiple Bioinformatics Information Sources project, known as TAMBIS, in which a domain ontology for molecular biology and bioinformatics is used in a retrieval-based information integration system for biologists. The ontology, represented using a description logic and managed by a terminology server, is used both to drive a visual query interface and as a global schema against which complex intersource queries are expressed. These source-independent declarative queries are then rewritten into collections of ordered source-dependent queries for execution by a middleware layer. In bioinformatics, the majority of data sources are not databases but tools with limited accessible interfaces. The ontology helps manage the interoperation between these resources. The paper emphasizes the central role that is played by the ontology in the system. The project distinguishes itself from others in the following ways: the ontology, developed by a biologist, is substantial; the retrieval interface is sophisticated; the description logic is managed by a sophisticated terminology server. A full pilot application is available as a Java™ applet integrating five sources concerned with proteins. This pilot is currently undergoing field trials with working biologists and is being used to answer real questions in biology, one of which is used as a case study throughout the paper.

The molecular biology community is a distributed one with a culture of sharing substantial quantities of rapidly evolving information. However, the development of a global informatics infrastructure to support this community has been piecemeal. Each area of molecular biology generates its own databases, and a wide range of specialized interrogation and analysis tools are commonly used over these resources. The Molecular Biology Database Collection,[1] for example, currently holds over 500 information resources, excluding the tools that analyze the information contained therein. The most popular resources include those concerned with protein sequences (SWISS-PROT, an annotated protein sequence database that originated in Switzerland, and PIR, the Protein Information Resource), genome data (ACeDB, a *Caenorhabditis elegans* database), DNA (deoxyribonucleic acid) sequences (EMBL—the European Molecular Biology Laboratory, and GenBank), protein structure (PDB, the Protein Data Bank), motifs (PROSITE, a database of protein families and domains, and PRINTS, a compendium of protein fingerprints), and sequence matching (BLAST, Basic Local Alignment Search Tool). Others are more specialized; for example, Yeast Proteome Database (YPD) and FlyBase are species-specific.

This network of information services forms a loose federation of autonomous, distributed, heterogeneous data repositories, ripe for information integration.[2] A number of approaches of varying sophistication have been adopted, from Web-based browsers to data warehouses.[3] The characteristics of this collection of resources are worth expressing in order to set the context for the rest of the paper, and our approach in particular.

0018-8670/01/$5.00 © 2001 IBM

Figure 1 Running example Query 1: A typical bioinformatics query over multiple data resources

Query 1: Select motifs for antigenic human proteins that participate in apoptosis and are homologous to the lymphocyte associated receptor of death (also known as lard).

Translation: Select patterns in the proteins that invoke an immunological response and participate in programmed cell death that are similar in their sequence of amino acids to the protein that is associated with triggering cell death in the white cells of the immune system.

(A) Concept expression in GRAIL:

```
Motif which
<isComponentOf (Protein which
  <hasOrganismClassification Species
    FunctionsInProcess Apoptosis
    HasFunction Antigen isHomologousTo
    Protein which <hasName
                        ProteinName>)>)>
```

Species: Is instantiated by value “human”
ProteinName: Is instantiated by value “lard”

(B) Equivalent expression in ALC standard Description Logic notation:

$$
\begin{aligned}
A &\equiv \text{Protein} \sqcap \exists\ \text{hasName.ProteinName}\\
B &\equiv \text{Protein} \sqcap \exists\ \text{isHomologousTo.A}\\
&\quad \sqcap \exists\ \text{hasFunction.Antigen}\\
&\quad \sqcap \exists\ \text{functionsInProcess.Apoptosis}\\
&\quad \sqcap \exists\ \text{hasOrganismClassification Species}
\end{aligned}
$$

$$\text{Motif} \sqcap \exists\ \text{isComponentOf.B}$$

(C) Informal query plan:

- Select proteins with protein name “lard” from SWISS-PROT
- Execute a BLAST sequence alignment process against SWISS-PROT results
- Check the entries for apoptosis process and antigen function
- Pass the resultant sequences to PROSITE to scan for their motifs

(D) CPL expression:

```
set-unique {(#motif1:motif1)|
\protein3 <- get-sp-entries-by-de("lard"), \protein2 <- do-blastp-by-sq-in-entry(protein3),
Check-sp-entries-by-kwd("apoptosis",protein2), check-sp-entries-by-de("antigen",protein2),
Check-sp-entry-for-species("human",protein2), \motif1 <- do-ps-scan-by-sq-in-entry(protein2)}
```

The data resources are frequently not databases in the conventional sense in that they do not have a separate schema containing their meta-data (or if they do, it is not freely accessible), and they do not have a declarative query language such as SQL (Structured Query Language). Most are tools, processes (e.g., sequence alignment), or proprietary flat file structures containing embedded meta-data, with a limited set of parameterizable services accessed through a call-based interface. Little distinction is made between databases (e.g., SWISS-PROT) and tools (e.g., BLAST). The sources have complete autonomy, continually extending their intensional and extensional coverage.

The resources are poorly integrated and difficult to use together (partially a consequence of the previous point). This condition is a drawback if we consider the complex retrieval tasks that biologists working in this environment are typically required to undertake. For example, consider the query in Figure 1, a typical query in drug target detection. Biologists must:

- Construct their own view of the meta-data in each source (the intension) and the instances covered by that source (the extension), resolving any semantic heterogeneities between the sources (e.g., SWISS-PROT covers some information on proteins, PROSITE covers motifs on protein sequences, and BLAST is a tool for matching sequences)
- Construct the various parts of the request in the different formats and terms required by the different sources, taking care to resolve structural and

semantic differences between seemingly similar information (e.g., the global unique identifier of a protein is its accession number, but these are inconsistent between sources)
- Locate and communicate with the sources, and process intermediate results into appropriate input formats for successive stages
- Interoperate between resources, unprotected from the vagaries of the various services provided, planning a series of requests that pick from each resource the information relevant to the query, and tracking and linking related instances through the sources. The requests usually incorporate some transformation processes that link together strictly ordered chains of retrieval, filtering, and processing procedures. There are often many alternative ways to resolve a request, which have varying efficiency, and the user has to choose among these options.

Heavy reliance is made on biologists' knowledge of molecular biology and bioinformatics, and their interpretation of each source whose intensional and extensional coverage is dynamic. If biologists wish to go beyond the standard provision offered by predefined query systems such as SRS (Sequence Retrieval System),[4] they must resort to developing their own analysis programs.

In this paper we present a prototype mediation system called TAMBIS (Transparent Access to Multiple Bioinformatics Information Sources), designed to relieve biologists of the task of choosing, combining, and interacting with the resources required to answer their queries. The TAMBIS architecture is service-oriented, centered on an extensive source-independent global ontology of molecular biology and bioinformatics, represented in a Description Logic, and managed by a terminology server. The ontological services are used extensively by the user interface component and during the query transformation process. The emphasis in TAMBIS is on the following:

- High transparency. TAMBIS aims to provide the illusion of a single query language, a single data model, and a single location. Biologists express requests independently of any knowledge of the sources.
- Read-only access. The autonomy of the sources is a basic requirement, and performing updates on the integrated view of the data resources is inappropriate and undesirable.
- Complex queries expressed over multiple diverse data sources. This characteristic implies a retrieval-oriented architecture in which there is a requirement for the production of efficient and correct query plans over multiple sources; the emphasis is on the coordinated interoperation of diverse sources.
- Heterogeneity management. As one would expect, there is considerable syntactic heterogeneity between the various bioinformatics sources, of the kind traditionally classified in the integration literature. There is some semantic heterogeneity, although the sources have limited overlap in their intensions or extensions.
- A visual query interface. Many researchers[2,5] stress that visual query interfaces are very important for facilitating the interaction of scientists with component databases, and that users are not prepared to handle query languages such as SQL. Current graphical user interfaces to biological sources generally only support the specification of predefined queries or have limited query languages (e.g., SRS[4]).

Many other researchers have proposed the use of ontologies for integrating heterogeneous sources,[6] although the approach has serious limitations.[7,8] The scope of the TAMBIS project is unusually wide, covering not only the ontological representation of a complex domain, but also the expression of queries in a visual interface and the execution of those queries over highly heterogeneous sources. We believe that TAMBIS has the following distinctive features:

- An unusually rich domain ontology, which currently contains around 1800 biological concepts and their relationships and is capable of inferring many more by way of compositional constraints encompassed in the ontology, known as *sanctioning*. Its coverage includes proteins and nucleic acids, their motifs, protein structure and structural classification, biological processes, and functions.
- A Web-based dynamic, compositional query formulation and ontology browsing interface entirely driven by the ontology service. The user interface ensures that only biologically coherent queries can be expressed and acts as an effective tutorial on the services available to the biologist.
- A query translation and planning process that identifies appropriate sources, plans an efficient way of executing a query, and generates an execution plan for use with a middleware layer. Sources accessed through TAMBIS need not themselves provide query language interfaces.

The TAMBIS pilot is currently in the field for evaluation trials by biologists. The pilot uses a subset of the ontology (250 asserted concepts covering proteins), with complete mappings to five of the most popular bioinformatics resources: SWISS-PROT (protein sequences); PROSITE (protein motifs); BLAST (sequence homology); ENZYME (data bank of enzyme classes); and CATH (a hierarchical classification of protein domain structures, clustering proteins at four levels: class, architecture, topology, and homologous superfamily, thus structural classification). With just this collection, the TAMBIS pilot can pose a significant number of the queries desired by biologists.[9] Note that, in line with the points above, the sources cover connected but complementary and barely overlapping information.

In the next section of this paper we present the TAMBIS Ontology (TaO) and the terminology services, and discuss the utility of combining a description logic with a compositional constraint system. In the subsequent section, we describe the TAMBIS architecture and its use of the ontology and wrapper services. Then we give an idea of TAMBIS in use, drawn from the pilot implementation. In the fifth section we present related work in bioinformatics and information integration, concluding the paper with a summary.

TaO: The TAMBIS global domain ontology

Biologists' knowledge of molecular biology and bioinformatics, and their interpretation of the resources with respect to this knowledge, is essential to the task of combining resources to answer queries. Practical exploitation of knowledge-based information integration systems has often been hindered in the past by the lack of suitable ontologies in challenging domains. Bioinformatics researchers have recognized that semantic schema and data matching could be aided by a comprehensive thesaurus of terms or a reusable reference ontology of biological concepts.[2,10]

TAMBIS uses a global, source-independent domain ontology to provide a unified conceptual level representation of its registered component resources. The ontology is described fully by Baker et al.[11] Bioinformatics is mainly concerned with the study of the protein and nucleic acid of biological macromolecules. Hence, these form the core of the ontology. The databases and the tools that TAMBIS covers describe the principal concepts of the molecular biology and bioinformatics part of the ontology: macromolecules and their motifs, their structure, function, cellular location, and the processes in which they act.

The ontology was developed by a biologist and a bioinformatician over a period of two years, using a range of modeling tools[12] and feedback from the TAMBIS interface itself. A mixed top-down and bottom-up iterative methodology was employed. The top-down component extended the upper levels of an ontology previously developed for a medical application[13] and reused the taxonomy of CATH. This top-down part of the ontology construction was complemented with a bottom-up approach that added further concepts to these general concepts such as specific motifs, kinds of secondary and tertiary structure of molecules, and cellular components. These concepts were gathered from the coverage of the databases themselves, their schemas, and their keyword collections. This bottom-up part of the construction helps ensure that the ontology covers the information in the sources themselves. Part of the future work of TAMBIS is to develop tools to facilitate this process. The emphasis is on the information coverage of these popular data sources and was specifically developed for a retrieval task. Only passing attention was paid to reusability and encoding bias. The ontology is organized into subdomains around protein structure, function, homology, location, and process, and is organized into those concepts concerned with molecular biology and those concerned with bioinformatics (for example, accession numbers of protein database entries). For a detailed description see Baker et al.[11] No other biological ontologies were reused at the time—the TAMBIS ontology predated the Gene Ontology[14] by some years, for example.

TAMBIS uses a global, source-independent domain ontology.

The ontology is described using the Description Logic GRAIL (GALEN Representation and Integration Language),[15] which is a formal and declarative representation. Descendants of the KL-ONE knowledge representation system, Description Logics[16] (DLs) describe the domain in terms of a limited set of primitive *concepts* that denote a set of *individuals* (instances) and *roles* that denote a set of binary relationships between individuals. An individual in the denotation of a concept is said to be an *instance* of that concept. Recursive *term constructors* associate concepts and roles to define new complex compositional concepts. A concept filling the value of role is a *role filler*. Using this Description Logic, the ontology comprises:

- A vocabulary for representing and communicating knowledge about molecular biology and a set of relationships that hold among the terms in that vocabulary
- A logic-based framework for reasoning about concepts and inter-relationships; for example, to infer that a concept is either more specialized than another or that a new concept is inconsistent with the rest of the ontology
- A constraint system, referred to as *sanctioning*, to control the combination of terms
- Constants represented as values and some individuals represented as nominals. GRAIL, in common with many DLs, has only reasoning over the terminology itself. Nominals are instances in the ontology. For example, Heam could be argued to be an instance of prosthetic group and not a class in its own right.
- Instantiation directives that indicate if a role filler is instantiable with a value (e.g., Protein Name or Species)
- Visibility directives that indicate whether an abstract concept should be visible to a browser

Reasoning services for description logics. The semantics of the term constructors is sufficiently well-defined to support reasoning about the concept descriptions. Consequently, Description Logics provide a variety of reasoning services[16] that make them attractive as models for describing complex and incomplete information, including:

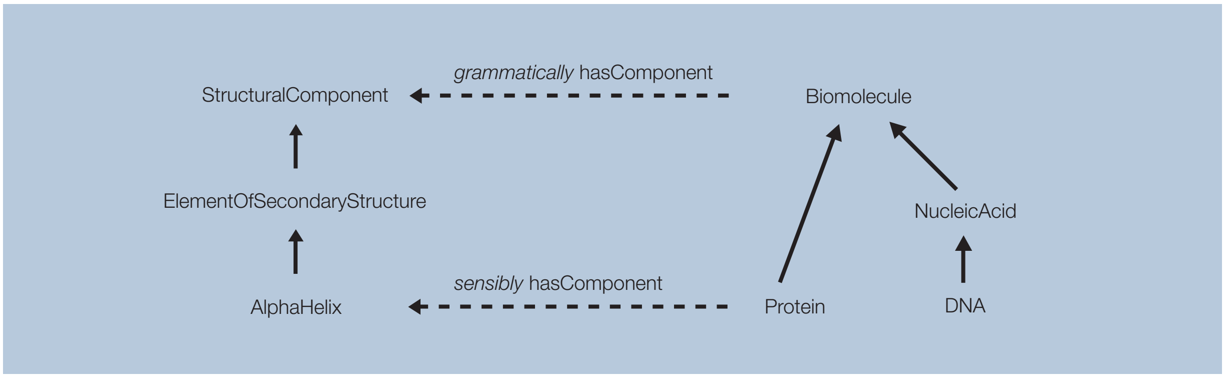
- *Subsumption*: One concept is said to subsume another when its extension must be a superset of the subsumed concepts' extension, as a logical consequence of their descriptions. *Primitive* concepts are just described as inclusion assertions ($\text{concept}_{\text{subsumee}} \subseteq \text{concept}_{\text{subsumer}}$). For *defined* compositional concepts, subsumption is *automatically inferable* using a suitable algorithm that is provably complete and decidable.[17]
- *Classification*: By using subsumption tests, a collection of conceptual definitions can be organized into a partial order. Newly defined concepts introduced to a pre-existing lattice are positioned into their correct place, dynamically evolving the classification structure.
- *Satisfiability*: Given a concept definition, we can determine whether the concept is satisfiable with respect to the subsumption lattice, i.e., if we cannot classify the concept, then it is unsatisfiable.
- *Retrieval*: Given a concept definition, we can retrieve all the instances of that concept, which includes all instances of subsumed concepts.

Description Logics balance expressivity, completeness, and tractability. In the past, results concerning the intractability of reasoning within DLs have been used to dismiss their use in real-world applications. Although, in the worst case, DL languages are generally known to be intractable, there have been tremendous advances in the last five years in the DL community in implementations of optimized reasoning engines, which can deliver realistic performance for practical applications. This is even true of highly expressive DLs such as FaCT (Fast Classification of Terminologies) and the SHIQ[18] reasoner logic. Thus, the old arguments of empirical intractability are outdated. GRAIL is less expressive than many other Description Logics and has a tractable subsumption test. GRAIL has only one concept-forming operator, which, and its language restricts expressions to be *conjunctive* ones with existential role quantification of the form (keywords are in italics):

BaseConcept *which* <role_1 Concept_1, . . . ,
role_n Concept_n> *name* ConceptName

Although GRAIL has a simple language and is comparatively inexpressive, we had a number of technical and pragmatic reasons for adopting its use initially. GRAIL compensates for its limited expressivity by supporting transitive roles, role hierarchies, a powerful set of concept assertion axioms, and a novel multilayered sanctioning mechanism for roles. Defining the role partOf to be transitive allows the query processor, for instance, to reason that a protein that is part of the inner mitochondrial membrane is also part of the mitochondrion. This can be useful when the sources only allow the larger grained query to be asked, but local processing can achieve the finer grained request. Both transitivity and role hierarchies allow sophisticated descriptions to be made for concepts that allow the reasoner to automatically infer subsumption (for example, a cytosolPart is a kind of cellularPart, so anything defined as a kind of cytosolPart is also a kind of cellularPart). The sanctioning

Figure 2 Sanctioned roles



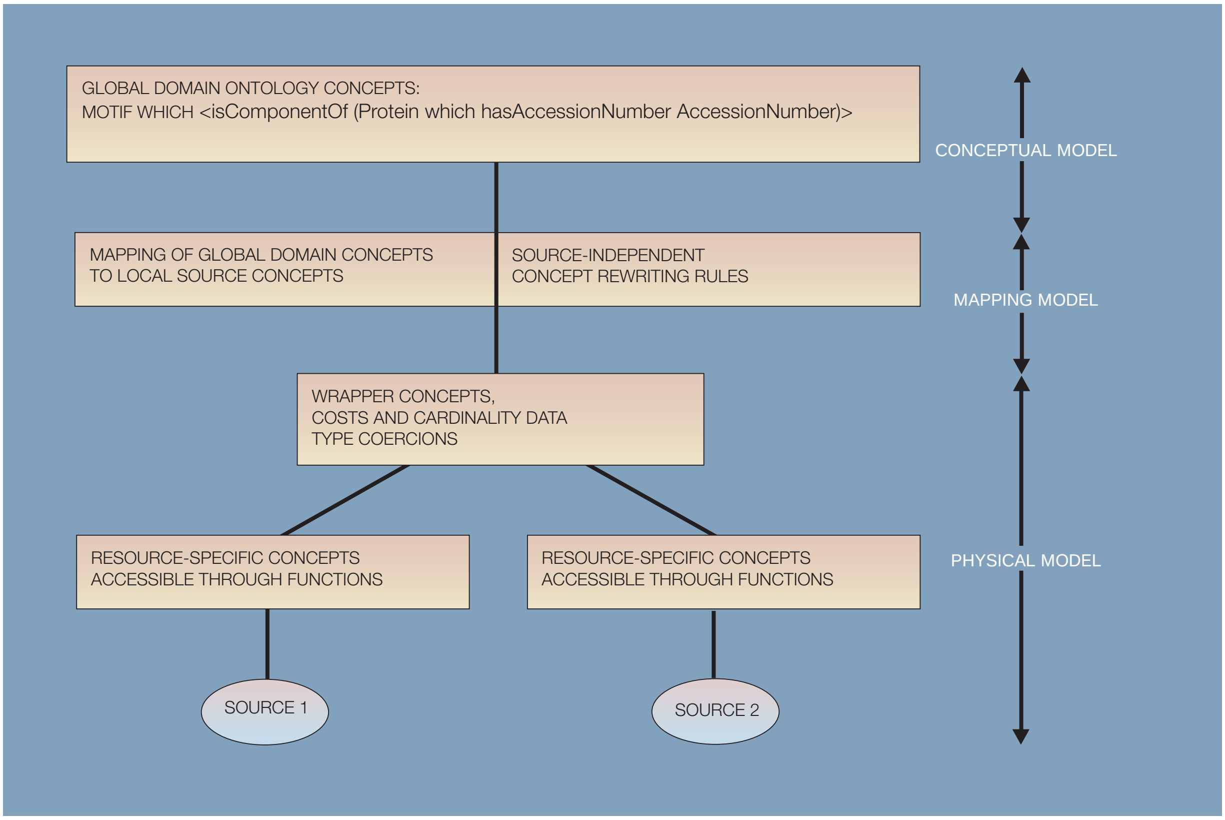
mechanism is crucial to the TAMBIS query formulation interface, since it constrains what concepts users may form, thus guiding them to form meaningful queries.

Sanctioned term construction. GRAIL operates a kind of "closed world" model. To restrict the construction of complex concepts to only those that are semantically meaningful (in terms of the ontology), GRAIL provides rules or sanctions that dictate which roles may legitimately be applied to which concepts. This model is quite different from the usual role restriction approach normally found in Description Logics. Two concepts can only be combined by a role if they have been explicitly sanctioned to do so. Sanctions are inherited. The sanctioning of concepts and their relationships allows a large number of complex concepts to be inferred (or generated) from a relatively sparsely populated model. Two levels of sanctioning are provided: *grammatical* and *sensible*. Grammatical sanctions express abstract or general relationships between classes of things, whereas sensible sanctions indicate that instantiable compositions can be built. A grammatical sanction must be in place before a sensible sanction can be made.

Figure 2 shows the sanctioning of the relationship hasComponent at the grammatical and sensible levels. The relationship between the concepts Biomolecule and StructuralComponent is sanctioned at the grammatical level because it is grammatically permissible to speak of biomolecules having structural components, but not all kinds of biomolecule can legitimately have any kind of structural component. The solid arrows in Figure 2 show explicit subsumption relationships. Thus, Protein is subsumed by Biomolecule and an AlphaHelix is subsumed by StructuralComponent. The hasComponent relationship between the concepts Protein and AlphaHelix is sanctioned at the sensible level because any kind of protein could legitimately have an alpha helix. However, not all biomolecules will have alpha helices (DNA, for instance)—sanctioning is about representing the *possibility* of composition, not its necessity. Although grammatical sanctions on their own do not permit the construction of instantiable composite definitions, they do represent valid queries that may be formed. In the example, asking for all biomolecules that have some structural component is a valid question. Given a set of concepts and suitably sanctioned roles, we can systematically combine them to create all possible legitimate composite concepts. For more on the ontology itself and the sanctioning mechanism, see Baker et al.[11]

Querying. Description Logics are naturally suited for expressing queries and defining views.[16] A concept formed as a query is resolved when its extension is retrieved. For example, the concept Protein which hasFunction Receptor describes the class of receptor proteins; retrieving the instances of that concept answers the query "find all proteins that act as receptors." The subsumption hierarchy is effectively a query inclusion hierarchy. We can navigate the lattice to find concepts that are the direct subsumers and subsumees of a concept. Such query generalization or specialization and imprecise querying is supported by: generalizing or specializing concept or role terms (through role hierarchies) and role fillers; relaxing or restricting constraints on roles; and adding or removing terms. Abstract or intensional answers are possible as well as the enumeration of

Figure 3 The TAMBIS models


GLOBAL DOMAIN ONTOLOGY CONCEPTS:
MOTIF WHICH <isComponentOf (Protein which hasAccessionNumber AccessionNumber)>
CONCEPTUAL MODEL
MAPPING OF GLOBAL DOMAIN CONCEPTS TO LOCAL SOURCE CONCEPTS
SOURCE-INDEPENDENT CONCEPT REWRITING RULES
MAPPING MODEL
WRAPPER CONCEPTS,
COSTS AND CARDINALITY DATA
TYPE COERCIONS
RESOURCE-SPECIFIC CONCEPTS ACCESSIBLE THROUGH FUNCTIONS
RESOURCE-SPECIFIC CONCEPTS ACCESSIBLE THROUGH FUNCTIONS
PHYSICAL MODEL
SOURCE 1
SOURCE 2

instances, for example, “What kinds of cellular processes are there?” or “What things can a motif be a component of?”

Terminology services. The reasoning services associated with GRAIL are encapsulated within a Terminology Server (TeS). The TeS supports concept reasoning, role sanctioning, thesauri, and extrinsics[19] services. The TeS supports a range of ASK and TELL interactions.[16] For example, given concepts A and B: Is A satisfiable? Does A subsume B? What are the direct and indirect subsumers or subsumees of A? What are the sanctioned roles and role fillers for A? What is the English language expression for A? Is the role filler for role r on A instantiable? Is role r_1 the inverse of role r_2? For further discussion see Bechhofer and Goble.[20] In the next section we show how the global ontology and the sources relate, and we outline how the TeS services are used in the TAMBIS components.

Architecture

The TAMBIS project adopts a top-down approach to integrating information sources, with three layers given in Figure 3, using a mixture of procedural (wrappers) and declarative (ontology) interoperation.[21] The domain ontology was constructed first, with mappings from the model to the underlying data source schemas being determined subsequently.

The physical model. The data resources are encapsulated by wrappers, described in the functional multidatabase language CPL, the Collection Programming Language.[22] The underlying data structures and file formats are converted to the nested, value-based model of CPL (see part D of Figure 1). CPL models complex data types such as lists, sets, and variants, with drivers (wrappers) that execute requests over data sources. Type coercions between functions of the different wrappers are handled at the wrapper level, as is location and format transparency. Consequently the wrappers are quite substantial. There is no local ontology in the sense of OBSERVER;[23] instead there are collections of functions that form the application programming interface (API) for the source. These functions provide a physical, rather than a logical, level of mapping since aspects such as alternative access paths are presented by functions. No cross-model assertions or representations exist; all intersource mapping is channeled through the global ontology. In effect the ontology coordinates intersource management.

The conceptual model. The global ontology is a unified conceptual level representation of its registered component resources, encompassing the concepts made accessible through their CPL wrapper functions. Source-independent queries are formulated in the same language as the conceptual description, hiding the sources from the user. However, the global ontology is more than the union of the schemas of its sources in that it provides an abstract framework for relating, reconciling, and coordinating the concepts of the sources. There are also queries (or subqueries) that can be answered intensionally based on the ontology alone.

The mapping model. The role of the query processor is to convert a query phrased only in terms of the conceptual layer into an executable plan in terms of the classes and methods of the physical layer. To do this, a range of mappings is required. These mappings are used either during query planning or by queries at run time. The mappings constitute the TAMBIS Sources and Services Model (SSM), which relates the wrapper services in the sources with their conceptual counterparts in the domain ontology. Currently, the sources and services data are constructed manually. Use of the same information used in the bottom-up part of the ontology construction helps to ensure that the SSM is complete—each concept is systematically checked to see whether it has a commensurate mapping. This task is, however, difficult to perform by hand, especially when sources or concepts change. Part of the future work on TAMBIS will be the development of tools to manage this mapping of concepts and relationships to wrapper classes and values in the sources.

Mapping concepts and roles to functions. Seven categories of mapping capture the relationships between the conceptual and physical representations of queries.[24] For example, the iteration mapping indicates that the instances of a concept can be obtained by iterating over values obtained by evaluating some function. An iteration mapping is thus a pair, consisting of a concept description and the description of the function that can be used to retrieve the instances of the concept. Each function carries CPL type, cardinality, and cost information. For example:

```
< concept: protein,
    function: < name: "get-all-sp-entries",
                arguments: [],
                resultType: "protein_record",
                cardinality: 80000,
                cost: 1000,
                source: "SwissProt"
              >
    >
```

In the above display, the CPL function, get-all-sp-entries, is declared to be able to return all proteins in the source SwissProt, in the form of values of CPL type protein_record. A more specialized concept may provide an alternative approach to iteration. For example, instances of the concept Protein which hasFunction CatalysisProtein which catalyzes Reaction can be obtained from the source Enzyme:

```
< concept: Protein which hasFunction
        CatalysisProtein which catalyzes Reaction,
    function: < name: "get-all-enzyme-entries",
                arguments: [],
                resultType: "enzyme_record",
                cardinality: 5000,
                cost: 200,
                source: "Enzyme"
              >
    >
```

The ontology is used during the query transformation process as a semantic index to the wrapper methods, and the subsumption mechanism is used to select the most specialized mapping available. For example, there is no entry for the concept Protein which hasFunction Receptor, but there is an SSM entry for the concept Protein which hasFunction biologicalFunction. The latter concept subsumes the former concept and the class of sources and services rules. Another entry indicates that a mapping can be used for the role filler Receptor as an argument to the method indicated by the subsuming concept. Thus, the ontology guides the choice of an appropriate method for the query concept component. Source-independent rewrite rules govern the choice of mappings and how the query components are combined.[24]

Mapping values. It is sometimes the case that a concept in the ontology maps onto a scalar value in a CPL query. For example, in Figure 1, the concept name Apoptosis maps (rather directly) to the string "apoptosis." These mappings can, however, be much less obvious. For example, the concept Kinase maps to "2.7.-.-" in a classification scheme used by the source Enzyme. This would be represented by the triple:

< Kinase, "Enzyme", "2.7.-.-" >

Architectural components. To realize the above model, TAMBIS has five major components organized into a classical mediator-wrapper three-layer model: a presentation layer, a mediation layer (dealing with mappings), and a wrapper layer (dealing with the physical models), as shown in Figure 4.

The Terminology Server, discussed earlier, is extensively used in the query formulation and transformation and in the Sources and Services Model. The components are written in, or interfaced to, the Java** programming language.

1. Query formulation interface: This is a graphical and forms-based interface in which the user browses the ontology and forms complex conceptual requests without having to memorize terms or be aware of the relevant information sources. The result is a declarative GRAIL concept expression. The interface has a desirable side effect of being a tutorial for biologists new to bioinformatics. More details on the interface are provided in the next section.
2. Query transformation processor. The input to the query processor is a GRAIL query, and the output is a CPL program. Because the CPL program represents an ordered collection of function calls, the query processor must not only generate a valid execution plan, it must also take into account the likely performance of alternative implementations for a query. Briefly, the process resembles a traditional database query planner: the conceptual query is decomposed into a collection of query components, and, using a cost model, alternative evaluation orders are identified and ranked. The ways in which each query component can be evaluated are generated with reference to the mappings described as outlined in the subsection on the mapping model. The subsumption service of the TeS is used in the selection of the most specialized mapping available at each point, so the mappings are effectively indexed by the concept model. The query processor is described in detail in Paton et al.[24]
3. Wrapper service. The wrapper service is provided by BioKleisli, a low-level mediation system developed for the biological community[22] that offers format and location transparency, but does not hide the sources from the user and does not offer schema or data reconciliation. BioKleisli is built on CPL. The service coordinates and dispatches the execution plans generated by TAMBIS to the appropriate wrapped component information services. Results are returned in HyperText Markup Language (HTML) as a Web page in a local Web browser.

A flavor of TAMBIS

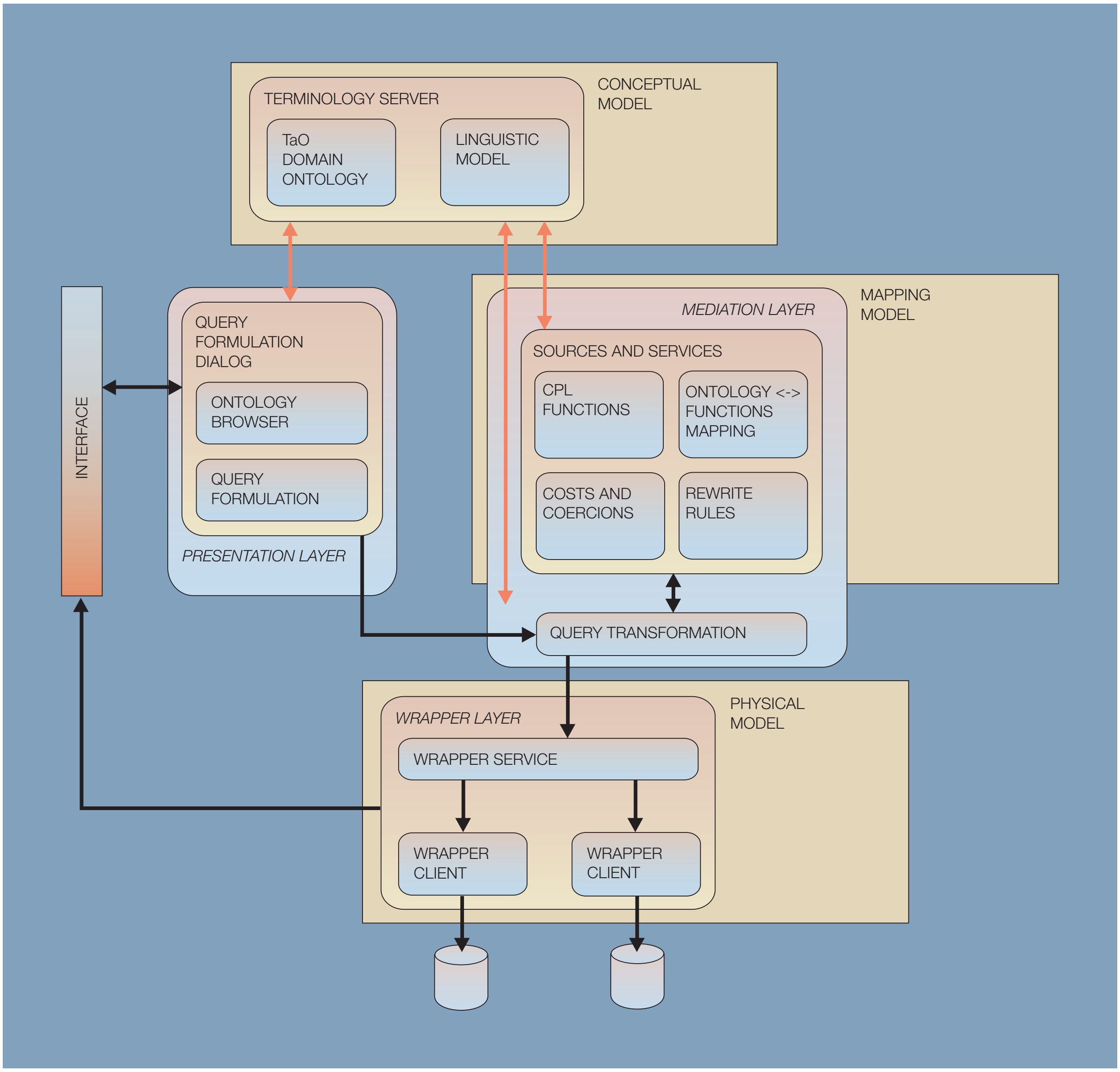
The TAMBIS interface, dynamically driven by the TeS services, supports the following:

- Browsing the ontology, so that users can find out what they can retrieve and how they can ask questions. The browser acts as an educational guide to information availability.
- Controlled incremental building and manipulation of query expressions through interaction with a visual representation of the ontology
- Instantiation of certain concepts with values (e.g., species with human)
- Identification of concepts that should be returned in the result
- Bookmarking queries

Users may construct complex queries by combining appropriately sanctioned concepts and roles. So, rather than testing whether an expression is correct by classifying it (as in other Description Logics), the interface forces users to create only expressions that are classifiable. The user interface does not force users to freely type their expressions; instead they are able to select from options presented to them that are guaranteed to produce a legal expression. The user interface guides the user as to which roles may be applied to any given concept at any given point during the construction of a query concept. Consequently, we need to be able to navigate between concepts and to expose which roles can legally be applied to a concept.

Figure 5 shows one of the ontology browsers on the concept Protein. The concept currently in focus occupies the center of the frame, with its subsuming and subsumed concepts and its sanctioned roles dis-

Figure 4 The TAMBIS component architecture

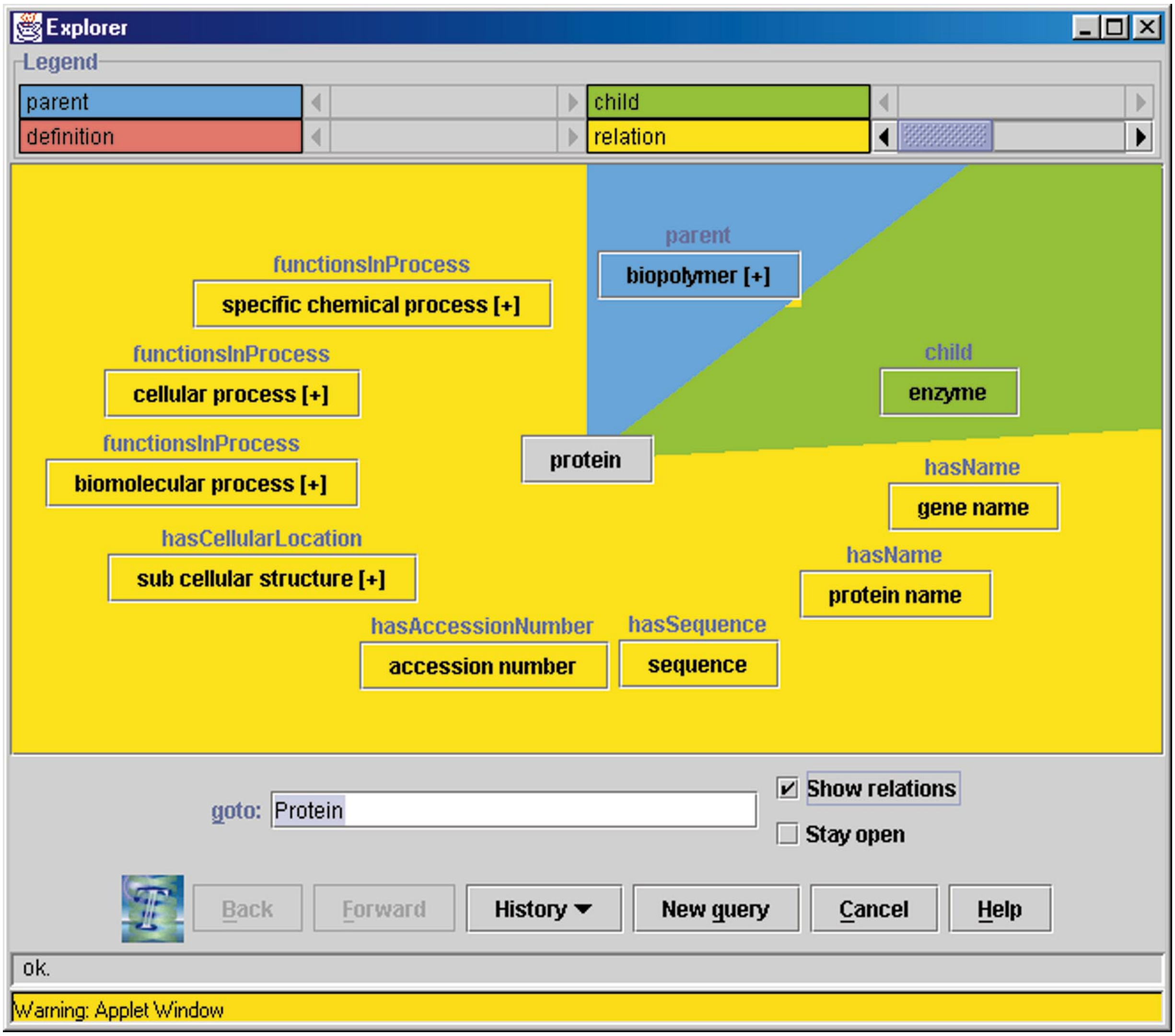


played around it. Another TAMBIS browser, Figure 6, presents the local subsumption hierarchy around the concept; this is used for substituting a more general, more specific, or sibling concept when manipulating a query expression.

The sanctioning mechanism allows us to explore potential legitimate relationships between concepts, restricting the user to asking questions it is sensible to ask. The role browser displays the legally sanctioned role-role filler pairs that may be applied to a base concept. Figure 7B gives those that can be applied to Protein which HasComponent Motif. Selecting the role filler selects the role (unfilled roles are not catered for).

Figure 8A shows the query expression for the query in Figure 1, and Figure 8B shows the CPL that the

Figure 5 An ontology browser showing the concept hierarchy and sanctioned roles for Protein



goto: Protein

☑ Show relations

□ Stay open

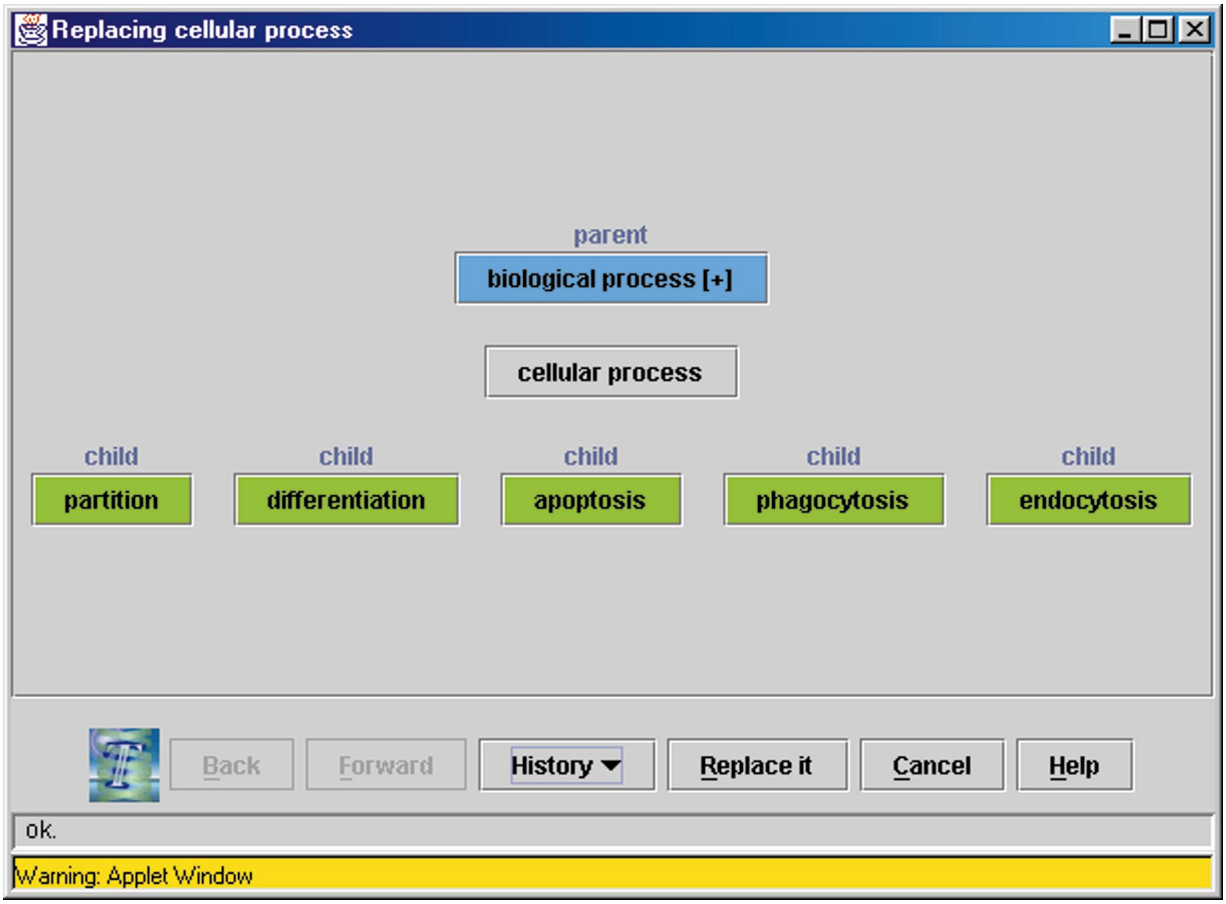
ok.

Warning: Applet Window

rewriter has produced. A query is built up from a base concept by adding or removing criteria using the role browser and substituting concepts for sibling concepts or those that are more specific or general by using the explorer browsers. The equivalent English expression for the concept is generated using the linguistic capabilities of the TeS and displayed at the top of the window. The subquery components of the query can be aggregated into complex expressions that can be manipulated semi-independently, in that the browsers operate on them independently, but the options available are dependent on the overall context of the query expression.

Although there is little space here to describe the manipulation interface in detail, we can give an idea of it. The primary results of Query 1 in Figure 1 are motifs, and the concept representing motifs forms the base of the query concept. The TAMBIS main screen holds a "find" dialog box with Explore and Build Query buttons, or bookmarks for previous queries. Suppose we pick a bookmark for the concept Motif isComponentOf Protein. Selecting "Build Query" launches a query builder window, with the query represented as two linked buttons (as in the two linked buttons near the top of Figure 8A). This concept represents all motifs that are components of proteins.

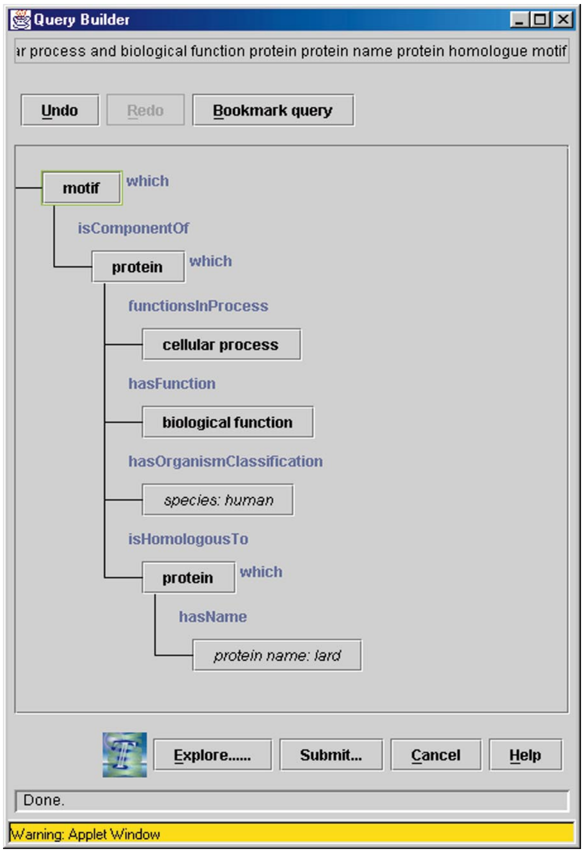
Figure 6 A local subsumption hierarchy browser centered on cellular process



We need to restrict the query so that it applies to only certain sorts of proteins. Clicking on the "protein" button invokes a menu displaying options for actions upon this concept (the concept menu). If we choose to Explore protein, we obtain an ontology browser as in Figure 5. If we choose to "restrict via a relationship," the ontology is asked what relationships this concept can hold. The answer returned by the TeS is displayed for the user as a list, as in Figure 7B. Criteria selected and accepted are added to the concept in the query builder window. In this case, the criteria functionsInProcess cellularProcess, hasFunction biological function, isHomologousTo protein, and hasOrganismClassification species are chosen. Now the query builder shows a new concept shown in Figure 7A. Each constituent concept is shown as a button, and the concepts are linked by lines representing the relationships between them. The "species" concept button is in an italic typeface—this indicates that the user may set a value for that concept. Choosing "set value" from the concept menu invokes a dialog so that the string "human" can be assigned to this concept. In this sense, the user is as much an information source as any of the bioinformatics databases. This value is also displayed in the query builder.

The Protein filler of isHomologousTo denotes homology to all proteins and hence should be restricted by, perhaps, a hasName relationship in the same way as before. The concepts cellularProcess and biologicalFunction are perfectly permissible but would result in a very general query. To restrict the query to proteins whose cellularProcess is apoptosis and whose biologicalFunction is antigen, these concepts should be specialized. By selecting the concept menu

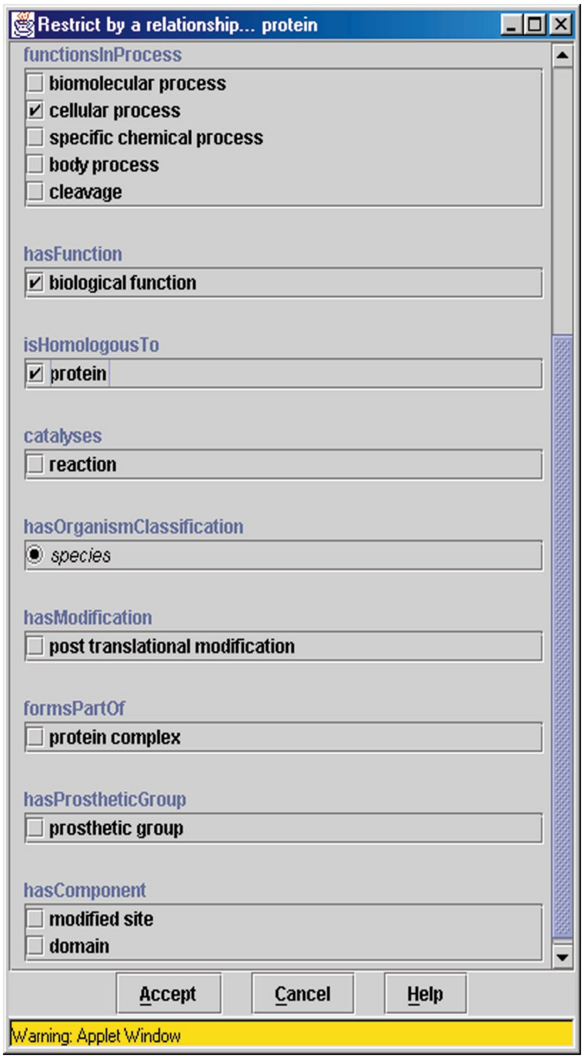
Figure 7A A rather general query expression after selecting roles and fillers



Query Builder

ır process and biological function protein protein name protein homologue motif

Figure 7B The roles and fillers sanctioned for Protein that has a Motif component



Restrict by a relationship... protein

functionsInProcess
- □ biomolecular process
- ☑ cellular process
- □ specific chemical process
- □ body process
- □ cleavage

hasFunction
- ☑ biological function

isHomologousTo
- ☑ protein

catalyses
- □ reaction

hasOrganismClassification
- ● species

hasModification
- □ post translational modification

formsPartOf
- □ protein complex

hasProstheticGroup
- □ prosthetic group

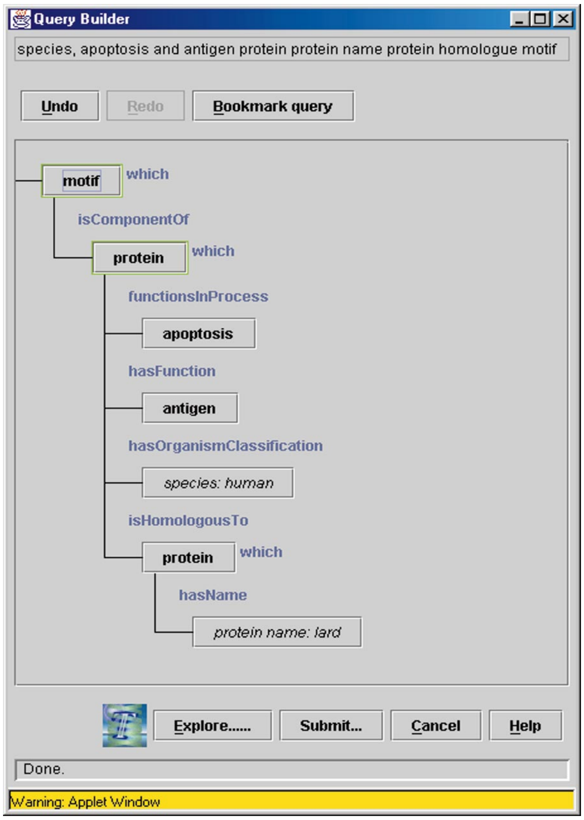
hasComponent
- □ modified site
- □ domain

option "replace with a kind of this concept" on cellularProcess, the user interface asks the TeS what the parents and children of the concept are and displays them as a hierarchy as shown in Figure 6. The user navigates to the more specialized concept apoptosis and selects "Replace," substituting apoptosis for cellularFunction in the query. A similar approach is taken for biologicalFunction. The final result is Figure 8A. Pressing the "Submit" button initiates the process of transforming the conceptual query into the concrete query plan shown in Figure 8B.

The compositional concept Query 1 has been classified and installed into the classification lattice as a result of its construction. Figure 9 shows the explorer browser on Query 1, illustrating some of the *defined* compositional concepts (bottom left). This demonstrates that the interface dynamically uses the TeS services to access the current state of the classification lattice. The visualization of defined concepts is usually disabled because it can be disturbing for users.

A number of ontology browsers have been designed to support ontology development[25,26] or retrieval.[27] However, query construction has been left to HTML-form-based query tools that require the user to have some knowledge of the query language and the modeling language (e.g., what a frame or slot is). In TAMBIS users are shielded from these concerns

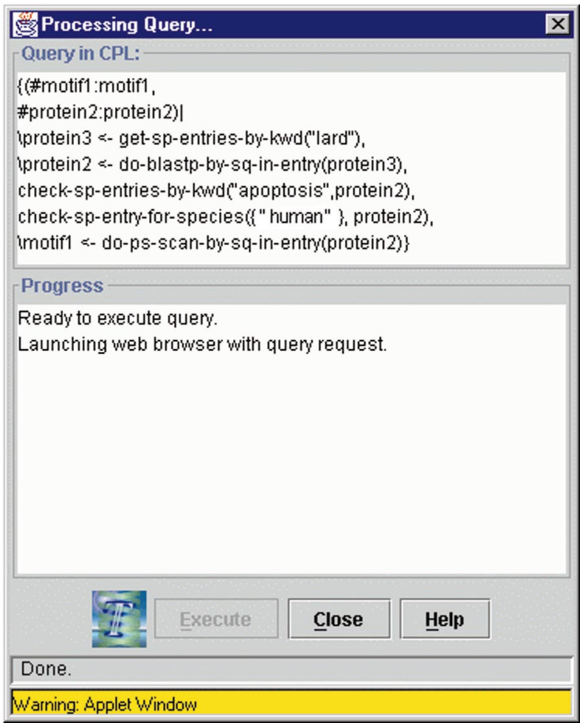
Figure 8A The query builder expression for Query 1 in Figure 1



species, apoptosis and antigen protein protein name protein homologue motif

Done.

Figure 8B The CPL produced



```
{(#motif1:motif1,
#protein2:protein2)|
\protein3 <- get-sp-entries-by-kwd("lard"),
\protein2 <- do-blastp-by-sq-in-entry(protein3),
check-sp-entries-by-kwd("apoptosis",protein2),
check-sp-entry-for-species({"human" }, protein2),
\motif1 <- do-ps-scan-by-sq-in-entry(protein2)}
```

Ready to execute query.
Launching web browser with query request.

Done.

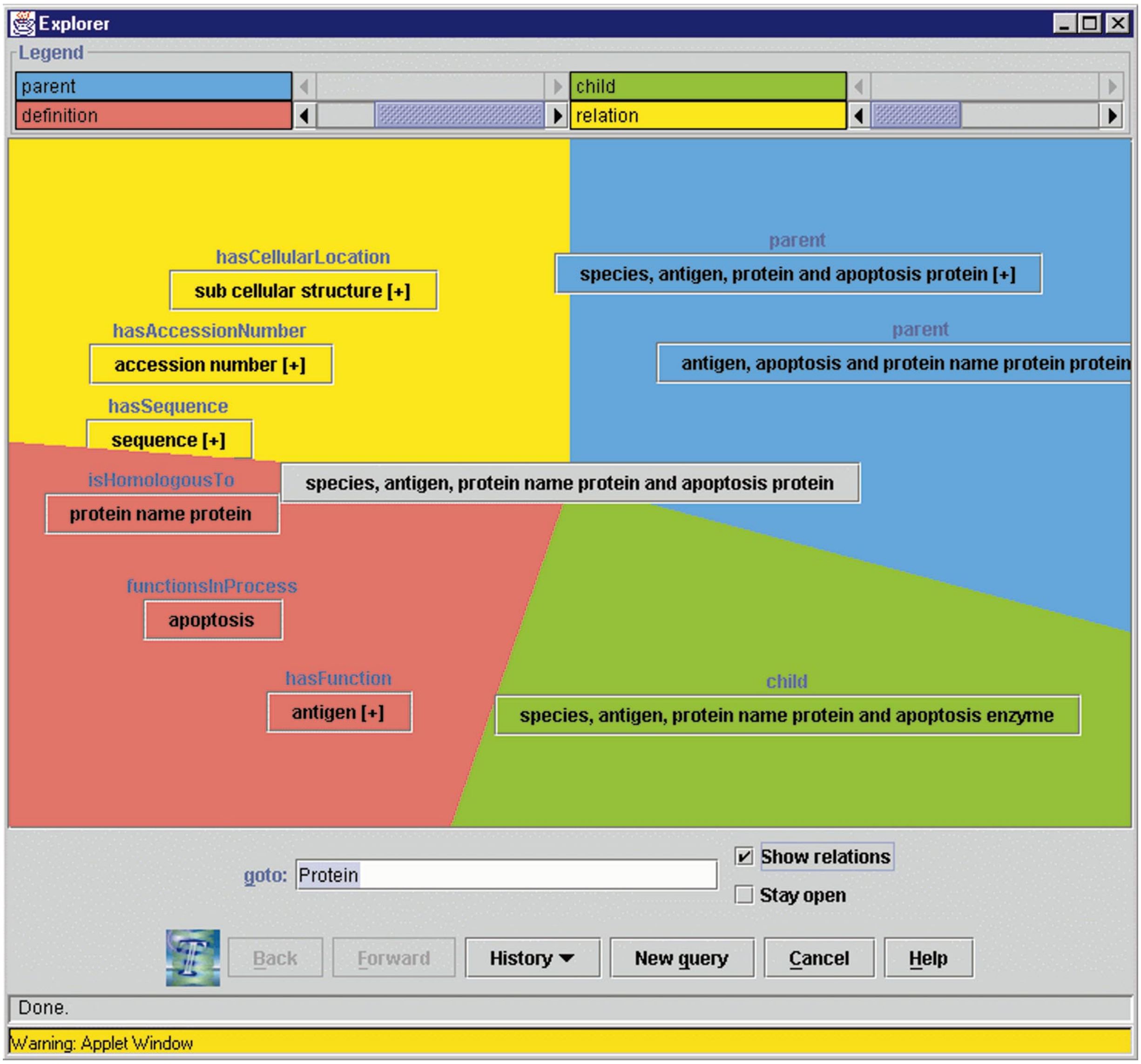
leaving them to concentrate in the domain rather than in the knowledge representation of the domain. In addition, the graphical layout makes the query much more readable and, consequently, comprehensible. More details on the interface can be found in Bechhofer et al.[28]

Related work

The closest work to TAMBIS in the biological domain is the Object-Protocol Model (OPM),[29] which uses an object model to implement a unifying schema. OPM does not provide source transparency, and queries are expressed in a variant of Object Query Language (OQL), or in a link-following manner through a graphical interface. SRS,[4] Entrez,[30] and BioNavigator[31] link several databanks and processes together through World Wide Web (WWW) front ends. Their source-linking functionality is similar to TAMBIS, but there is no source or schema transparency, no ability to issue complex declarative and multidatabase queries, and limited flexibility, since only a limited range of access paths through predefined links or retrieval functions are available. Systems such as BioKleisli[32] and DiscoveryLink[33] are lower-level middleware solutions that concentrate on multisource query languages, wrapping sources, and intersource query optimization. They offer format and location transparency but do not hide the sources from the user and do not offer schema or data reconciliation. TAMBIS sits on top of such a middleware layer as we have already done with an earlier version of CPL/BioKleisli.[22]

Many proposals have been made that use domain ontologies, often expressed in a Description Logic, in intelligent "read-only view" mediation services.[23,34–36] However, none of these has the ontology-driven query dialog of TAMBIS. But most are also based on query decomposition and expansion and send subqueries to source databases.[37] Unlike TAMBIS, most [e.g., Levy et al.[35] and Mena et al.[23]] target sources that provide declarative query facilities. Many[34,35,38–40] adopt a "top-down" approach us-

Figure 9 Explorer browser on Query 1 from Figure 1



goto: Protein

☑ Show relations

□ Stay open

Done.

Warning: Applet Window

ing a global schema encompassing relevant information. However, TAMBIS does not express the data held in the sources as views over this schema.[41] SCOPE[42] and DQW (Data Quality Warehouse)[21] emphasize semantic reconciliation of heterogeneous sources, using ontologies to identify interschema semantic relationships and representing them as assertions. In TAMBIS there are no interschema relationships; all mappings are from local concepts to global ones. The emphasis is not on *discovering* semantic similarities or conflicts but on *managing* the already identified global-local concept mappings.

Carnot[36] used the Cyc knowledge base as the global schema, which is a general ontology rather than the detailed domain ontology used in TAMBIS. The scope of its follow-on, InfoSleuth,[43] is more concerned with the provision of a generic agent architecture for information integration than the development of a specific user-centered retrieval mediator such as TAMBIS.

A particular application of InfoSleuth, EDEN,[44] and CoopWARE[40] uses a global domain ontology to support agent-based semantic interoperability.

Unlike TAMBIS, Information Manifold[35] adopts a "local as view" approach,[41] where the sources are described as views over a global ontology expressed in the CLASSIC Description Logic system. This approach potentially makes the source descriptions more modular, proving useful when sources change or join the federation. However, efficient execution of queries expressed in the ontology across multiple sources is much more difficult. OBSERVER[23] associates each source with an ontology (again using CLASSIC) that describes its contents. It thus exploits existing ontologies or describes each source using a new ontology, and relates them. However, query processing is targeted more at the selection of a single source for answering a query than at global planning for queries run over multiple sources. TSIMMIS (The Stanford-IBM Manager of Multiple Information Sources)[38] uses a lightweight integrating object model and places great emphasis on wrappers, concentrating on automated wrapper generation. There is no attempt to create a declarative integrating domain model, and no schema reconciliation. The closest project to TAMBIS is SIMS.[34] It is based around a source-independent domain model expressed in the LOOM Description Logic system; queries expressed against this model are rewritten to source-dependent queries also in LOOM. The query planner and optimizer of SIMS have influenced those in TAMBIS. TAMBIS differs from SIMS in that our sources rarely provide query interfaces, and in having a visual query construction interface provided by TAMBIS. The first TAMBIS prototype reported here does not follow SIMS in having source models in the Description Logic, though the next generation TAMBIS adopts a variant of this approach.

Discussion

TAMBIS aims to provide *complete transparent access*: transparency of location, source, and data structures. A substantial global domain ontology supported by a range of terminological services is the cornerstone of the provision of this transparency. The biologist interacts with a single schema through one interface, using a single language. The schema and query language are effectively the same. The ontology of some 1800 assertions was developed by a biologist and bioinformatician.[11]

A *visual query interface* that supports the exploration of the ontology and the controlled incremental creation and manipulation of complex query expressions is made possible through the use of the terminological services. We know of no comparable query interface supported by a knowledge-based information integration system. Although the interface looks somewhat like a form-based interface to a database system, it is able to detect a range of biologically nonsensical queries that more conventional query interfaces to databases would be happy to compile, but which are sure to yield empty results.

We can express source-independent declarative *complex queries* that range across *multiple diverse sources*. A sources and services model associates concepts from the ontology with source-dependent wrapper functions written in CPL. The ontology forms a canonical model to relate the shared concepts to their source-specific counterparts and forms a framework to *manage heterogeneity* between the sources. A query planner uses the sources and services model and the terminological services to generate a source-dependent execution plan from the GRAIL query, which is executed through the BioKleisli middleware.

TAMBIS does not assume that individual sources export query facilities—in bioinformatics very few sources do. Sources are wrapped by CPL at essentially the same level as CORBA** (Common Object Request Broker Architecture**) sources are wrapped, so TAMBIS can be seen as generating output at a level that is typical of that required for use over widely accepted middleware layers.

The first TAMBIS prototype is operational and accessible through a password-protected Java applet at http://img.cs.man.ac.uk/tambis.html. TAMBIS was developed in close collaboration with biologists in academic institutions and pharmaceutical companies, and the applet is currently undergoing trials in a range of universities and specialist national bioinformatics institutes, and at AstraZeneca Pharmaceuticals. Five popular resources have been fully integrated into TAMBIS covering an extensive range of real biological questions that have been difficult to pose before, such as that given in Figure 1. TAMBIS can now pose complex questions that the data sources cannot answer or on which they hold no data, which in itself is a useful piece of information. For example, the following are a list of some of the types of queries that can be formulated in TAMBIS. They range from standard bioinformatics queries to those that are difficult to ask without development of a bespoke program:

- Find human homologues of yeast receptor proteins.
- Find rat proteins that have a domain with a seven-propeller domain architecture.
- Find phosphorylation motifs on human apoptosis receptor proteins.
- Find protein homologues of a protein with a particular accession number.
- Find the binding sites of human enzymes with zinc cofactors.

The evaluation of TAMBIS broadly falls into two categories: technical effectiveness and user usability. From the technical point of view, the TAMBIS prototype has demonstrated that the approach is feasible, identifying issues of source integration and transparency, which we discuss later. From the user interface point of view, we have not undertaken a systematic usability study but rather an informal analysis. On the plus side, the concept browsers and controlled query formulation is recognized as a flexible way of forming complex queries by term combination. This also has the side effect of acting as a tutorial on what requests to the services are available. User trials have also identified a range of issues including: the appropriate presentation of a large ontology on a small amount of screen; the accurate interpretation of the query by the user; and the identification of paths to follow in the construction of a query concept; for example, is a concept specialized by adding a restricting role or choosing a more specialized subclass? The TAMBIS query interface is different from a normal text-based one and takes some familiarization. The emphasis is on querying rather than click-based navigation as in SRS or Entrez. Describing what is wanted rather than describing how to get it is a new paradigm for many bioinformaticians used to thinking in terms of processes rather than questions. By way of a compromise, we suggest that an expert bioinformatician can use TAMBIS to generate a series of "canned" predetermined parameterizable queries and make this the interface for the bench biologist.

Relieving the user of the onerous task of linking the sources passes the creation and maintenance of the illusion of one terminology and one source to the mediation system. The TAMBIS approach is a top-down one. The global domain model was constructed initially, with mappings from the model to the underlying data resource schemas determined subsequently. This "global as view"[41] approach is effective for the purposes of expanding and rewriting queries against the global ontology, which is the main purpose of TAMBIS.

However, this approach carries the cost of (A) building, validating, and maintaining the ontology and (B) building, validating, and maintaining the mediation mappings.[8] There is a strong dependency between the ontology and the resources. This is particularly so because the mappings are often complex. Two features of the web of resources make these difficult tasks, which is why the mediation process is hidden from the user in the first place. The number of resources is large, and the resources change without notification. Resources fall out of use and others emerge. The relationships between resources change as well as their relevance or popularity.

The development of a single schema or ontology is a serious and expensive undertaking best tackled as a joint exercise with others by merging and adopting pre-existing ontologies, with the intention that the result will be reusable by other applications. This task is difficult.[45] The use of a single terminology by a mediator requires that the user know what is in the terminology, understand what the terms and concepts mean, and buy into it. Gaining consensus is particularly difficult because one user's or community's vocabulary might differ from that of another. The ontology will need to be tended and updated to cater to new sources or changes in sources. It also needs to be comprehensive enough to cater to an appropriately adequate range of resource types. Interpretations of concepts often depend on context, and one ontology cannot be viewed as a repository of all possible interpretations.[7,45] Attempts to tackle this issue range from the adoption of *de facto* common vocabularies by a community prepared to adapt to some form of common consensus, for example, the Gene Ontology,[14] to mechanisms for defining ontological commitment, multiple definitions for concepts in the same ontology, and ontological views. Description Logics are particularly convenient for giving multiple definitions to concepts, unlike frame or other object-based schemes. The TAMBIS ontology is designed specifically for one task, retrieval over bioinformatics resources, and thus may not be an appropriate interpretation for, say, reasoning about protein function.[45]

The two major issues that we are tackling in the next TAMBIS prototype are source management and source transparency. Source management—the introduction of new sources and keeping up with changes—is the time-consuming activity in an environment where the data sources are autonomous. Wrappers must be developed and mapped to ontological concepts through the SSM, remembering that the mapping is often complex. Fundamental research into a more flexible "local as view" source integration mechanism, where sources are described independently in terms of a shared terminology, is still at an early stage.[41] The support of automated source integration without hampering the efficient global rewriting machinery is our next challenge. Integration systems such as TAMBIS are generally hampered by the paucity of the accessibility functions of the sources. Many offer only point and click interaction interfaces designed for people, not programs, and HTML or flat file results. Web wrappers that screen-scrape are brittle because sources change their interfaces frequently. Consequently, in practice, mediation systems tend to integrate a rather modest number of sources.

TAMBIS, in contrast to similar systems such as OPM and SRS, is very transparent; users are completely shielded from any notion of which source will be used to answer their query, or the order in which the sources will be used. In some cases this position is too extreme; many biologists will be keen to intervene in the rewriting process to direct requests to their favorite sources, to alter the order of execution, or to siphon off intermediate results. At the very least they require an explanation of the query plan and the results to be decorated with their originating source. Removing the responsibility of integration from the user can have the potentially detrimental side effect of taking control from the user, or users perceiving that control has been wrested from them. Confidence in the results is related to control. Given the volatility of the Web, all users seek reassurance on the quality and suitability of the resources used, which requires that the provenance of the results, and an explanation of how they were obtained, is as essential as the results themselves.

A second TAMBIS prototype has already been developed that replaces GRAIL in the Conceptual Model with a far more expressive and mainstream Description Logic, FaCT,[18] and replaces CPL in the Physical Model with Java wrappers. This forms a platform on which we can address the issues of source integration and transparency.

Acknowledgments

This work is supported by the BBSRC/EPSRC Bioinformatics Programme grant BIF/05344, by AstraZeneca Pharmaceuticals, and by EPSRC grant GR/M76607 under the DIM initiative. The Java version of the interface was initially ported by Alex Jacoby. We are grateful for his contribution.

**Trademark or registered trademark of Sun Microsystems, Inc., or Object Management Group.

Cited references and note

1. A. D. Baxevanis, "The Molecular Biology Database Collection: An Online Compilation of Relevant Database Resources," *Nucleic Acids Research* **28**, No. 1, 1–7 (2000).
2. P. Karp, "A Strategy for Database Interoperation," *Journal of Computational Biology* **2**, No. 4, 573–586 (1995).
3. O. Ritter, P. Kocab, M. Senger, D. Wold, and S. Suhai, "Prototype Implementation of the Integrated Genomic Database," *Computers and Biomedical Research* **27**, 97–115 (1994).
4. T. Etzold and P. Argos, "SRS—An Indexing and Retrieval Tool for Flat File Data Libraries," *Computer Applications in the Biosciences* **9**, No. 1, 49–57 (1993).
5. V. M. Markowitz and O. Ritter, "Characterizing Heterogeneous Molecular Biology Database Systems," *Journal of Computational Biology* **2**, No. 4 (1995).
6. A. Ouksel and A. Sheth, "Editorial: Special Section on Semantic Interoperability in Global Information Systems," *ACM SIGMOD Record* **28**, No. 1, 5–12 (1999).
7. A. Ouksel and I. Ahmed, "Ontologies Are Not the Panacea in Data Integration," *Journal of Distributed and Parallel Databases* **7**, 1–29 (1999).
8. C. A. Goble, "Supporting Web-based Biology with Ontologies," *Proceedings of the Third IEEE International Conference on Information Technology Applications in Biomedicine* (ITAB00), Arlington, VA (November 2000), pp. 384–390.
9. R. Stevens, C. A. Goble, P. G. Baker, and A. Brass, "A Systematic Classification of Bioinformatics Queries," *Bioinformatics* **17,** No. 1, 1–9 (2001).
10. S. B. Davidson, C. Overton, and P. Buneman, "Challenges in Integrating Biological Data Sources," *Journal of Computational Biology* **2**, No. 4 (1995).
11. P. G. Baker, C. A. Goble, S. Bechhofer, N. W. Paton, R. Stevens, and A. Brass, "An Ontology for Bioinformatics Applications," *Bioinformatics* **15**, No. 6, 510–520 (1999).
12. W. D. Solomon, GALEN-IN-USE Project Deliverable 9.2, The GRAIL KnoME Knowledge Modelling Environment (1998), http://www.galen-organisation.com.
13. A. L. Rector, P. E. Zanstra, and the GALEN-IN-USE Consortium, "The GALEN-IN-USE Project," *European Health Telematics Observatory Journal* (1996).
14. Ashburner et al., "Gene Ontology: Tool for the Unification of Biology," *Nature Genetics* **25**, 25–29 (2000).
15. A. L. Rector, S. Bechhofer, C. A. Goble, I. Horrocks, W. A. Nowlan, and W. D. Solomon, "The GALEN Modelling Language for Medical Terminology," *Artificial Intelligence in Medicine* **9**, 139–171 (1997).
16. A. Borgida, "Description Logics in Data Management," *IEEE Transactions on Knowledge and Data Engineering* **7**, No. 5, 671–682 (1995).
17. F. M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf, "Rea-

soning in Description Logics," *Foundations of Knowledge Representation*, CSLI Publications (1996), pp. 191–236.

18. I. Horrocks and U. Sattler, "A Description Logic with Transitive and Inverse Roles and Role Hierarchies," *Journal of Logic and Computation* **9**, No. 3, 385–410 (1999).
19. Arbitrary meta-data associated with concepts and roles that take no part in the Description Logic reasoning services.
20. S. Bechhofer and C. A. Goble, "Delivering Terminological Services," *AI*IA Notizie* **12**, No. 1, 27–32 (1999).
21. D. Calvanese, D. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati, "Information Integration: Conceptual Modeling and Reasoning Support," *Proceedings of the 6th International Conference on Cooperative Information Systems* (CoopIS-98) (1998), pp. 280–291.
22. P. Buneman, S. B. Davidson, K. Hart, C. Overton, and L. Wong, "A Data Transformation System for Biological Data Sources," *Proceedings of VLDB*, Zurich (September 1995).
23. E. Mena, V. Kashyap, A. Seth, and A. Illarramendi, "OBSERVER: An Approach for Query Processing in Global Information Systems Based on Interoperating Across Pre-Existing Ontologies," *Proceedings of COOPIS* (1996), pp. 14–25.
24. N. W. Paton, R. Stevens, P. G. Baker, C. A. Goble, S. Bechhofer, and A. Brass, "Query Processing in the TAMBIS Bioinformatics Source Integration System," *11th International Conference on Scientific and Statistical Database Systems* (1999), pp. 138–147.
25. P. D. Karp, V. K. Chaudhri, and S. M. Paley, *A Collaborative Environment for Authoring Large Knowledge Bases* (1997).
26. B. Swartout, R. Patil, K. Knight, and T. Russ, "Towards Distributed Use of Large-Scale Ontologies," *Proceedings of 10th Knowledge Acquisition Workshop* (KAW'96) (1996).
27. D. Fensel, S. Decker, M. Erdmann, and R. Studer, "Ontobroker: How to Make the WWW Intelligent," *Proceedings of 11th Banff Knowledge Acquisition for Knowledge Based Systems Workshop* (KAW'98) (1998).
28. S. Bechhofer, C. A. Goble, R. Stevens, G. S. Ng, and A. Jacoby, "Browsing and Query Formulation in Large Ontologies," *Workshop on User Interface for Data Intensive Systems* (UIDIS), IEEE Press (September 1999), pp. 158–161.
29. I. A. Chen and V. M. Markowitz, "An Overview of the Object-Protocol Model (OPM) and the OPM Data Management Tools," *Information Systems* **20**, No. 5, 393–418 (1995).
30. D. A. Benson, M. Boguski, D. J. Lipman, and J. Ostell, "Genbank," *Nucleic Acids Research* **22**, 3441–3444 (1994).
31. BioNavigator, see http://www.bionavigator.com.
32. L. Wong, "Kleisli: Its Exchange Format, Supporting Tools and an Application in Protein Interaction Extraction," *Proceedings of the IEEE International Symposium on Bio-Informatics and Biomedical Engineering*, Arlington, VA (2000), pp. 21–28.
33. L. Hass, P. Kodali, J. Rice, P. Schwarz, and W. Swope, "Integrating Life Science Data—With a Little Garlic," *Proceedings of the IEEE International Symposium on Bio-Informatics and Biomedical Engineering*, Arlington, VA (2000), pp. 5–12.
34. Y. Arens, C. A. Knoblock, and W.-M. Shen, "Query Reformulation for Dynamic Information Integration," *Journal of Intelligent Information Systems* **6**, No. 2/3, 99–130 (1996).
35. A. Y. Levy, A. Rajaraman, and J. J. Ordille, "Querying Heterogeneous Information Sources Using Source Descriptions," *Proceedings of the 22nd International Conference on Very Large Databases*, VLDB-96, Bombay, India (September 1996).
36. M. P. Singh, P. Cannata, M. Huhns, N. Jacobs, T. Ksiezyk, K. Ong, A. Sheth, C. Tomlinson, and D. Woelk, "The Carnot Heterogeneous Database Project: Implemented Applications," *Journal of Distributed and Parallel Databases* **5**, 207–225 (1997).
37. R. Hull, "Managing Semantic Heterogeneity in Databases: A Theoretical Perspective," *ACM PODS* (1997), pp. 51–61.
38. Y. Papakonstantinou, A. Gupta, H. Garcina-Molina, and J. Ullman, "A Query Translation Scheme for Rapid Implementation of Wrappers," *Proceedings of DOOD 95*, Springer-Verlag, Inc. (1995).
39. S. Bergamaschi, S. Castano, and M. Vincini, "Semantic Integration of Semi-Structured and Structured Data Sources," *ACM SIGMOD Record* **28**, No. 1, 54–59 (1999).
40. A. Gal, "Semantic Interoperability in Information Services: Experiences with CoopWARE," *ACM SIGMOD Record* **28**, No. 1, 68–75 (1999).
41. J. D. Ullman, "Information Integration Using Logical Views," *Proceedings of the International Conference on Database Theory—ICDT97* (1997), pp. 19–40.
42. A. M. Ouksel and C. F. Naiman, "Coordinating Context Building in Heterogeneous Information Systems," *Journal of Intelligent Information Systems* **3**, No. 1, 151–183 (1994).
43. R. Bayardo et al., "InfoSleuth: Semantic Integration of Information in Open and Dynamic Environments," *Proceedings of ACM SIGMOD*, Tucson, AZ (May 1997), pp. 195–206.
44. J. Fowler, B. Perry, M. Nodine, and B. Bargmeyer, "Agent-Based Semantic Interoperability in InfoSleuth CoopWARE," *ACM SIGMOD Record* **28**, No. 1, 60–67 (1999).
45. R. Stevens, C. A. Goble, and S. Bechhofer, "Ontology-based Knowledge Representation for Bioinformatics," *Briefings in Bioinformatics* **1,** No. 4, 398–414 (2000).

Accepted for publication December 22, 2000.

Carole A. Goble *Department of Computer Science, University of Manchester, Oxford Road, Manchester M13 9PL, United Kingdom (electronic mail: carole@cs.man.ac.uk).* Professor Goble co-leads the Information Management Group, which she founded in 1995 with Norman Paton. Her academic career has been spent at the University of Manchester. She gained her B.Sc. degree in computing and information systems in 1982, joined the faculty staff in 1985, and became full professor in 2000. Her principal interests are in meta-data, knowledge representation, and ontologies, and their use in hypermedia, information integration, intelligent user interfaces, and intelligent retrieval. She has worked in a variety of application areas, notably medical informatics, bioinformatics, and conceptual hypermedia and multimedia databases. Current work includes automating scientific database annotation, ontology-driven intelligent interfaces for retrieval and data entry in scientific databases, the visualization of ontologies, and technologies for the semantic web.

Robert Stevens *Department of Computer Science, University of Manchester, Oxford Road, Manchester M13 9PL, United Kingdom (electronic mail: stevensr@cs.man.ac.uk).* Dr. Stevens is a Research Associate in the Information Management Group. He gained his B.Sc. degree in biochemistry at the University of Bristol in 1986; an M.Sc. degree in bioinformatics in 1991, and a D.Phil. degree in computer science in 1996, both at the University of York. His doctoral research was on user interfaces. His specializations are in ontology construction and reconciliation of semantic heterogeneity in bioinformatics resources.

Gary Ng *Department of Computer Science, University of Manchester, Oxford Road, Manchester M13 9PL, United Kingdom (electron-*

ic mail: ngg@cs.man.ac.uk).* Dr. Ng is a Research Associate in the Information Management Group. He gained his B.Sc. degree in 1993 and his M.Sc. degree in computer science in 1995 at the University of Manchester. He recently completed his doctorate degree on visualization techniques for ontology development. His main interest is in user interfaces for knowledge bases.

Sean Bechhofer *Department of Computer Science, University of Manchester, Oxford Road, Manchester M13 9PL, United Kingdom (electronic mail: seanb@cs.man.ac.uk).* Mr. Bechhofer is a Research Fellow in the Information Management Group. He gained his B.Sc. degree in mathematics at the University of Bristol in 1988, and has worked in industry as well as academia. He has been a member of the Information Management Group since it was founded in 1995. His main research interests are in the applications of description logics, particularly as a delivery mechanism for terminologies, semantic meta-data, and ontologies.

Norman W. Paton *Department of Computer Science, University of Manchester, Oxford Road, Manchester M13 9PL, United Kingdom (electronic mail: norm@cs.man.ac.uk).* Dr. Paton is a Professor of Computer Science. He obtained a B.Sc. degree in computing science from Aberdeen University in 1986 and a Ph.D. degree from the same institution in 1989. He was a lecturer in computer science at Heriot-Watt University from 1989 to 1995, before moving to Manchester, where he co-leads the Information Management Group. His research interests have principally been in databases, in particular, active databases, spatial databases, deductive object-oriented databases, and user interfaces to databases. He is currently working on parallel object databases, spatio-temporal databases, and distributed information systems. He is also leading work in the development of an object-oriented data warehouse for yeast genomic data and provides database expertise for a number of databases in microbial eukarytes.

Patricia G. Baker *Sagitus Solutions Ltd., Incubator Building, Grafton Street, Manchester M13 9XX, United Kingdom (electronic mail: p.baker@sagitussolutions.com).* Dr. Baker gained her B.Sc. degree in biochemistry from Liverpool John Moores University in 1990, an M.Sc. degree in computer science from UMIST in 1991, and a Ph.D. degree in computational biochemistry from the School of Biological Science at the University of Manchester in 1995. She has worked in academia and in the biotechnology industry and is currently the technical director of Sagitus Solutions, a company specializing in knowledge-based bioinformatics systems for drug discovery.

Martin Peim *Department of Computer Science, University of Manchester, Oxford Road, Manchester M13 9PL, United Kingdom (electronic mail: peim@cs.man.ac.uk).* Dr. Peim is a Research Associate in the Information Management Group. He obtained his B.Sc. degree in 1981 and a Ph.D. degree in 1989, both in mathematics, at the University of Manchester. He spent two years at the University of Kentucky (1986–1988), before taking up computer science. Before joining the TAMBIS project, he worked on distributed decision support systems, automated reasoning, and hardware design verification.

Andy Brass *School of Biological Sciences, University of Manchester, Oxford Road, Manchester M13 9PL, United Kingdom (electronic mail: abrass@man.ac.uk).* Dr. Brass is a Senior Lecturer in the School of Biological Sciences. He started his career as a theoretical physicist working on molecular modeling of superionic solids, gaining his Ph.D. degree from Edinburgh University in 1987. He then moved to McMaster University in Canada on a NATO fellowship to study aspects of high-temperature superconductivity and strongly coupled electron systems. In 1990 he moved to the University of Manchester to become a founding member of the bioinformatics group, where he has a wide range of projects in protein function prediction, gene expression analysis, intelligent integration, automated curation, and bioinformatics education.