

# Annotation in Scientific Data: a Scoping Report (Draft)

Peter Buneman

Rajendra Bose  
University of Edinburgh

Denise Ecklund

May 1, 2005

## Abstract

Possibly the most distinctive feature of scientific data is both the manner and the extent of the use of annotation. In this report we attempt to describe how and why annotation is used, to understand some of the generic issues in annotation, and to indicate directions for relevant research, especially in the areas of databases and digital curation.

## 1 What is annotation?

Most people will agree with the dictionary definition of *annotation* as the process of adding comments or making notes on or upon something. Such notes have traditionally served a variety of purposes, including explaining, interpreting or describing some underlying text. Annotation is often for personal use but, more importantly in our context, it can be a means of disseminating useful information. For example, annotated bibliographies and textual criticism are well understood uses of annotation. Annotation of images and plans is also commonplace; much of cartography is about spatial annotation.

The use of on-line, digital data has caused a revolution in the way scientific research is conducted. In every area of science, much investigation now depends not on new experiments, but on databases in which experimental evidence has been stored. However, this evidence is seldom raw experimental data; it is typically some form of interpretation of the data, and annotation is an increasingly important part of that interpretation. Nowhere is this more obvious than in molecular biology, where the value of some databases lies almost entirely in the annotation they add to data extracted from other databases. This added value often represents substantial investment of effort; databases such as UniProt (Universal Protein Resource) (See Section A.1) are supported by a large number of curators or annotators. There is also an increasing amount of machine-generated annotation: pattern recognition and machine learning techniques are being used in biology and astronomy to flag suspect data.

In contrast to annotation within databases, other forms of annotation are externally affixed “over” a body or collection of data. An example of this concerns written comments attached to a set of existing web pages (See section 3.2). This is similar to the concept of superimposed information presented by Maier and Delcambre, that is, “data ‘placed over’ existing [base] information sources to help organize, access, connect and reuse information elements in those sources.” Here we focus on their description of using superimposed information for annotation that “might serve to explicate, evaluate, correct or refute the base information.” [MD99]

Annotation may also be viewed as a temporal phenomenon: a body of related descriptors or other text that may change with time. Consider base or “fundamental” descriptors as a fixed, static quantity of information

that provides essential information about some object, perhaps initially prepared by the object's creator. After its creation, the object may be annotated multiple times by various people (or programs), with annotations adding to a body of "additional" descriptors that grows "alongside" or "on top of" earlier annotation(s) over time.

Annotation is an area broad enough to have multiple interpretations as above. It is also a basic activity in the publication of scientific and scholarly data. It is therefore essential that the database community and the whole community of digital publishers obtain a common understanding of this process. This report is an attempt to do this and to provide some pointers to areas in which research is needed.

## 1.1 Uses and meaning of annotation

Digital items that are the focus of annotation include:

- text documents, including web pages and linguistic corpora
- structured data, including relational and XML databases, and text encodings of biological structures (genes, proteins, etc.) and other phenomena
- 2D images, including medical images
- other multimedia, such as video and audio, including movies and fieldwork video
- 3D visualizations and other types of spatial and temporal data

Although in some cases annotation may serve as "metadata" by describing the structure or purpose of a database, this popular term is so overloaded that we shall try to avoid using it. Consider the case of a simple database of NASA Landsat Earth scene images, which consist of a digital encoding of the satellite sensor readings composing each image together with a description of when the image was obtained, the satellite location, the details of the instrumentation that obtained the image, the details of the encoding, etc. Here the distinction between the image (the data) and the rest (the metadata) is reasonably clear. But as we have remarked, scientific databases typically contain information derived from the original experimental data. For example, the Supercosmos Sky Survey database (Section A.2) is a digest of data collected from a variety of measurements; the sequence data in UniProt is not raw sequence data, but likely to be a translation of DNA sequence data that has been assembled by a variety of gene recognition techniques. Are we to regard these as primary data or annotation of other – possibly transient – data? In short, we find attempts to classify data into data, metadata, annotation data, etc. rather confusing. What is less ambiguous is the *process*, which we call annotation, of adding data to something that already exists.

Within this broad definition, different communities associate different shades of meaning to the word annotation to explain, interpret, or describe (as in bioinformatics and the semantic web), as well as to: classify or categorize (2D images, audio, video); evaluate or distill (as in computational linguistics); or clarify or refute (book and movie reviews on the web, wikis). In bioinformatics, annotating (the digital representation of, or database record for) a genomic sequence means assigning or interpreting its function [RDS04]. This meaning is gaining ground: about half of the roughly 150 combined IEEEExplore, ACM Digital Library and Web of Science citations for 2003-2004 with "annotation" in the title use this connotation of the term.

For those involved with the Semantic Web, annotating a web resource means specifying machine-processible meaning for it [ZM03]. However, web sites like Amazon or the Internet Movie Database today allow the

more traditional sense of annotation by supplementing their book or movie reviews with (possibly moderated) customer comments. Wikis greatly expand on this concept by allowing unchecked additions, modifications, or annotations of a set of web pages, typically for a particular interest group.

For those working with digital images, annotating an image often means identifying a section of the image to comment upon or simply providing a caption. Annotating a film clip could mean attaching text or audio descriptions, facts or interpretations to different temporal sequences of video. Other examples mentioned in the literature include annotating photos or images with geolocation, annotating programming code, and annotating 3D data visualizations. There is interest in performing these sometimes mundane or lengthy tasks automatically or semi-automatically.

The role of annotation in scientific work is integral: for annotation in these communities, the focus is on description or interpretation from trusted sources that may inform further interpretation or research, possibly performed by others in other organizations or outside the community. In his prescient 1945 essay "As We May Think" which describes the idea of constructing hyperlink-like "trails" of related pieces of information, Vannevar Bush states: "A record, if it is to be useful to science, must be continuously extended, it must be stored, and above all it must be consulted." [Bus45] Understanding contemporary forms of annotation and databases, and the interplay between the two, is the key to facilitating the continuing extension of the digital scientific record.

## 1.2 Outline of the rest of the paper

In the following sections we first look at a series of examples of annotation in order to attempt to distill out some key concepts. We next look at the handful of systems and theories that have been constructed to deal with annotation either in some generic fashion or in a specific domain. Finally we indicate areas in which some research, standardization, and codes of good practice are needed.

## 2 Annotation examples and concepts

### 2.1 Annotation and the evolution of database structure

Figure 1 shows a single entry in UniProt. As we have remarked, it is debatable what parts of this entry one would regard as data, metadata or annotation. However, from a database perspective, the entry illustrates several interesting points, including the evolution of structure. The structure of the entry is an old, purpose-built file format with a two-letter code giving the meaning of each line of text. Notice that the comment lines (CC) have become structured with entries of the form `- ! - FUNCTION: . . .` which provide a degree of machine-readability of the comment text. These entries were presumably not anticipated by the designers of the original format, and the alternative of specifying some further two-letter codes for these entries, was presumably ruled out as it would confuse existing software designed to parse the format. There are now 26 such subfields, one of which has additional machine-readable internal structure.

The important observation here is that annotation plays an important part in the evolution of both the form and content of data. What was once unknown or regarded as *ad hoc* annotation has become part of the database structure. It is almost certainly the case that the curators of UniProt now make extensive use of database technology and that what is exported in Figure 1 is a "rendering" or database view of the internal data. While database management systems provide some help with structural evolution, it is always problematic. This is something we shall comment more on in Section 4.1.

```

ID 11SBCUCMA STANDARD; PRT; 480 AA.
AC P13744;
DT 01-JAN-1990 (REL. 13, CREATED)
DT 01-JAN-1990 (REL. 13, LAST SEQUENCE UPDATE)
DT 01-NOV-1990 (REL. 16, LAST ANNOTATION UPDATE)
DE 11S GLOBULIN BETA SUBUNIT PRECURSOR.
OS CUCURBITA MAXIMA (PUMPKIN) (WINTER SQUASH).
OC EUKARYOTA; PLANTA; EMBRYOPHYTA; ANGIOSPERMAE; DICOTYLEDONEAE;
OC VIOLALES; CUCURBITACEAE.
RN [1]
RP SEQUENCE FROM N.A.
RC STRAIN=CV. KUROKAWA AMAKURI NANKIN;
RX MEDLINE; 88166744.
RA HAYASHI M., MORI H., NISHIMURA M., AKAZAWA T., HARA-NISHIMURA I.;
RL EUR. J. BIOCHEM. 172:627-632(1988).
RN [2]
RP SEQUENCE OF 22-30 AND 297-302.
RA OHMIYA M., HARA I., MASTUBARA H.;
RL PLANT CELL PHYSIOL. 21:157-167(1980).
CC -!- FUNCTION: THIS IS A SEED STORAGE PROTEIN.
CC -!- SUBUNIT: HEXAMER; EACH SUBUNIT IS COMPOSED OF AN ACIDIC AND A
CC BASIC CHAIN DERIVED FROM A SINGLE PRECURSOR AND LINKED BY A
CC DISULFIDE BOND.
CC -!- SIMILARITY: TO OTHER 11S SEED STORAGE PROTEINS (GLOBULINS).
DR EMBL; M36407; G167492; -.
DR PIR; S00366; FWPULB.
DR PROSITE; PS00305; 11S_SEED_STORAGE; 1.
KW SEED STORAGE PROTEIN; SIGNAL.
FT SIGNAL 1 21
FT CHAIN 22 480 11S GLOBULIN BETA SUBUNIT.
FT CHAIN 22 296 GAMMA CHAIN (ACIDIC).
FT CHAIN 297 480 DELTA CHAIN (BASIC).
FT MOD_RES 22 22 PYRROLIDONE CARBOXYLIC ACID.
FT DISULFID 124 303 INTERCHAIN (GAMMA-DELTA) (POTENTIAL).
FT CONFLICT 27 27 S -> E (IN REF. 2).
FT CONFLICT 30 30 E -> S (IN REF. 2).
SQ SEQUENCE 480 AA; 54625 MW; D515DD6E CRC32;
MARSSLFTFL CLAVFINGCL SQIEQQSPWE FQGSEVWQQH RYQSPRACRL ENLRAQDPVR
RAEAEAI FTE VWDQDNDEFQ CAGVNMIRHT IRPKGLLLPG FSNAPKLIFV AQGFGIRGIA
IPGCAETYQT DLRRSQSAGS AFKDQHQKIR PFRGDLVVV PAGVSHWMYN RGQSDLVLIV
FADTRNVANQ IDPYLRKFYL AGRPEQVERG VEEWERSRKGSSGEGSGNI FSGFADEFLE
EAFQIDGLV RKLKGEDDER DRIVQVDEDF EVLLPEKDEE ERSRGRYIES ESESENGLEE
TICTLRLKQN IGRSVRADVF NPRGGRISTA NYHTLPILRQ VRLSAERGLV YSNAMVAPHY
TVNSHSMVYA TRGNARQVV DNFQSVFDG EVREGQVLMIPQNFVVIKRA SDRGFEWIAF
KTNDNAITNL LAGRVSQMRM LPLGLVLSNMY RISREEAQRKYGQQEMRVL SPGRSQGRRE

```

Figure 1: An entry from UniProt

## 2.2 Location and attachment of annotations

The annotations in the CC fields in Figure 1 appear to refer to the entire UniProt entry. Reading down, one finds feature table (FT) lines that contain “fine-grain” annotation about different segments of the sequence data. There is a subtle difference between the two forms of annotation. The CC annotations are understood to refer to the whole entry because they occur *inside* that entry. The FT annotations are *outside* the structure being annotated and therefore require extra information, in this case a pair of numbers specifying a segment, to describe their attachment to the data. Notice that this assumes a *stable* co-ordinate system. If the sequence data were to be updated with deletions or insertions, attachment of annotations would be problematic.

Consider another, fanciful, example of a fine-grain attachment in which one wants to say something like “The third author of the first citation is John Doe”. One could imagine inserting “John Doe” in the actual text of the RA line, but this is likely to interfere with any software that parses this line. Alternatively one could place it externally in some other field of the entry. Once again, this assumes that the “co-ordinate system” is stable. For example, it assumes that the numbering of the citations does not change when the database is updated.

Another issue is the attachment of an annotation to several entries/objects in any of the databases we are considering. One could place the same annotation (with references to all relevant entries) in each of the relevant entries, but this is a standard example of “non-normalized” data. The solution is to build a separate annotation table, or “stand off markup” [TM97] with links to the appropriate entries. Again, this requires an extension to the existing structure of the database.

## 2.3 Annotating annotations

Anyone who has used a web site (for example, [www.amazon.com](http://www.amazon.com) or [www.eopinions.com](http://www.eopinions.com)) that includes opinions or reviews will have seen requests such as “Would you like to comment on this review?” or “5 out of 6 readers found this opinion useful.” Since annotations frequently carry opinions, the ability to attach another opinion to an existing opinion is essential. But there is no real difference between a system that allows annotations of annotations and a newsgroup that supports *threads*. Since very few scientific databases are designed to support *ad hoc* annotation, there is little opportunity for these databases to be used as vehicles for scientific discussions; however there is some evidence, which we discuss in Section 3.1, that indicates that such databases are desirable.

The topic of annotating existing annotations introduces questions about tracking and retrieving the *provenance* of multiple annotations. One already existing research area concerns the ability to retrieve the provenance (lineage) of base data, apart from annotations, which is of interest to many consumers of scientific data products. In fact, some forms of annotation may assist in resolving the provenance of an object.

The provenance of annotations, however, is a separate issue. If one considers annotation as a body of “additional” metadata that can grow on top of earlier annotation over time as described in Section 1, then we can imagine wanting to recover the provenance of any one of multiple annotations for a given object. For example, one may want to see the lines of connection between annotations and their sources (who or what created them), or the lines of connection between various annotations. To our knowledge this topic has not yet been addressed in the literature.

## 2.4 Querying annotations

Work in the ediKT project at Edinburgh (<http://www.edikt.org>) concerning a case study involving the Edinburgh Mouse Atlas Project (EMAP) suggests that users of the mouse atlas want to be able to query annotations for two distinct purposes: (1) to locate annotations where the annotation values themselves are of interest (“show me all annotations which have a value of ‘gene expression pattern X’”); and (2) to locate annotations where the associated base data values are of interest (“show me all the annotations associated with the following mouse atlas images”). Many existing annotation systems provide only a limited ability to query over annotation values. For example, consider systems for web page annotation: queries on this type of annotation might be limited to find capabilities supported in the client browser.

Supporting annotation queries for case (1) is more likely to be straightforward than case (2). For the second case one needs to know where the annotation is attached to the base data and perhaps why it is attached. How this is captured in the database and expressed in the query is an open question.

## 3 Annotation system examples

The following examples feature systems used by various communities that are investigating and implementing annotation for their own purposes, including bioinformatics and biology, and computational linguistics. More general systems for annotating web resources and data within relational databases are also being developed.

### 3.1 Annotating genomic sequences

The Distributed Annotation System (DAS) [SSD02] is an Open Source project that allows a client to view annotations on genomic sequence data gathered from on the order of tens of existing *annotation* or *reference servers*. Queries are made to these servers via HTTP GET or POST; annotation servers respond with lists of annotations across a specified segment of a genome.

IBM developerWorks uses a similar client/server architecture to provide a general solution for annotation of digital data (<http://www-106.ibm.com/developerworks/webservices/library/ws-annotation.html>). In this scenario, an annotation is an XML document that is linked to a target data object (for example, a database, word processing document or spreadsheet) with a unique preexisting identifier (or one generated by a hash value). They define an Annotation Web services API consisting of methods for communication between an annotation client and server for creating, updating, and retrieving annotations and annotation structure definitions. The API is an implementation-neutral set of SOAP-based Web services calls. [Wei03] refers to a system to store and retrieve annotation for the drug discovery process based on the IBM InsightLink product which contains an implementation of the Annotation Web services API.

Many other systems exist for annotating genomic data. The SEED project [RDS04] is similar to DAS, but more ambitious in infrastructure. The project focuses on allowing an individual researcher to perform rapid gene sequence annotation, to integrate his private data with public databases during the annotation process, and to view annotation for related biological function across many organisms rather than for just one organism.

MyGrid [ZGG<sup>+</sup>03] includes projects that use a graphical workflow editor to assist bioinformatics researchers in using a series of annotation-related web services during the process of annotating a genome sequence. This work also experiments with semi-automatic semantic labeling of annotation workflows.

### 3.2 Annotating web pages

The Third Voice, now defunct, was an early attempt at creating a system to allow third parties to annotate arbitrary static HTML pages. The annotations were stored in a separate database, and when an HTML page was viewed through a proprietary, intermediate, web site, the annotations would appear as “sticky” notes on an otherwise standard rendering of the HTML page. Annotations could be marked in order to restrict their visibility.

The Third Voice illustrated the difficulty (also experienced by writers of “screenscraping” software) of finding stable points of attachment in HTML pages that are subject to modification. If the software could no longer determine the point of attachment of an annotation, the annotation would “fall off” the page and would be displayed at the bottom.

Similarly, the W3C Annotea project views annotations as “comments, notes, explanations, or other types of external remarks that can be attached to any Web document or a selected part of the document without actually needing to touch the document.” (<http://www.w3.org/2001/Annotea/>) (See also [KKPR01]) Annotea uses RDF/XML on annotation servers linking annotation URIs to URIs for the sections of web pages that are the targets of annotation. The Amaya browser functions as an Annotea client.

For the general problem of annotating (describing) web resources created by others, [HVS03] focus on what they term the deep web, for example the RDBMSs behind the generation of dynamic web pages. They suggest the process of deep annotation, in which a database administrator exposes the RDBMS structure via descriptive metadata on the web server, which allows a potential annotator to annotate the dynamically created web pages according to their own set of terms. A person or program could potentially query the RDBMS using the annotator’s description terms.

### 3.3 Annotating medical images

Some systems have been designed to create web-accessible collections of annotated medical images. Gertz [cite] develop a graph model of annotations: annotation nodes serve to connect specific image region-of-interest nodes with concept nodes from a controlled vocabulary. Graph edges define the relationship between nodes: for example, *annotates* or *ofConcept*. They also develop a framework for querying annotation graphs based on path expressions and predicates, which they test in a prototype system.

The Edinburgh Mouse Atlas Project (EMAP) involves two types of annotations for images. EMAP provides annotations that make connections between both a standard anatomical nomenclature and the results of tissue-level gene expression experiments with regions of 3D mouse embryo tissue images and 2D tissue slices. This project provides a suite of tools, including an interactive website (<http://genex.hgu.mrc.ac.uk/intro.html>). The tools allow one to browse text nomenclature and make queries about gene expressions that return sets of images or a list of genes expressed for a given embryo image. Another way to query for gene expressions is to interactively select an area of a 2D image.

EMAP involves centralized editorial control and curation; an editorial review board decides whether to accept gene expression experiment results, and regions of images are manually colored by an expert.

## 4 Areas for further research

One of the most useful effects a report such as this could have would be to help the designers of a new database, schema or data format to prepare their data for annotation. Of course, some databases, especially

those in bioinformatics, are designed to receive annotation. But we have seen many examples of the need to accommodate *ad hoc* annotation and the need for *ad hoc* annotation to migrate to a more systematic form of annotation, that is, to become part of the regular database structure.

We identify two major issues here. The first is *extensibility* of data in order to allow for the migration of *ad hoc* annotation into the structure of the database. The second is providing some notion of *location* so that annotations can be attached to the relevant parts of the database. We deal with these two issues separately, and close by discussing research issues for annotation and relational databases.

## 4.1 Annotation and extensibility

This topic has always been of major concern in databases. How does one change the structure of a database without having to modify existing applications that use the database? Relational databases provide a partial answer to this. One can create a new table or add a column to an existing table without having to modify most basic SQL queries that run against this data<sup>1</sup>. Since the query language of SQL is the only method of extracting data from a relational database, there is some guarantee that other applications will not be affected by these extensions. In programming languages, the use of extensible record types is well-understood (see, for example, [Rém93],) and there are standard techniques for dealing with extensibility in object-oriented languages and databases.

Unfortunately the story is not nearly so simple for XML. The equivalent of adding a column to a relational table would be the addition of a new sub-element to each of a sequence of elements. A simple application of XPATH that uses only “vertical” axes is unlikely to be affected; however whether a DOM or SAX application would work with such a modification depends very much on where the sub-element is inserted and how the application is written.

Type systems for XML such as DTDs and XML-Schema offer little of help here. In designing a DTD for extensibility one would like to say something like “an A element must contain B, C, and D elements, and may contain other elements.” Unfortunately there is no way of doing this, and attempts to do this lead are often counter-productive [Cho02]. XML-Schema has an extensible ALL specification.. But one cannot, using ALL, specify the order of the sub-elements. Given this, and given that most XML applications do not require a DTD or Schema in order to function, it is probably a good idea to use a very small fragment of XML-Schema, something corresponding to a nested relational model in designing extensible exchange formats and community models.

## 4.2 Attachment and co-ordinates of annotations

We noted, in Section 2.2, that annotations were sometimes placed inside the annotated object and sometimes outside and that many annotations were, for reasons of database security, necessarily stored externally. External annotations require some kind of co-ordinate system in order to specify how they are to be attached to the data. It is worth a brief digression not observe that the point of attachment does not tell us everything. Consider the annotation of one value of the table shown in Figure 2. and consider some possibilities for *(annotation)*:

1. This is a prime number

---

<sup>1</sup>The behavior of operations like *natural join* could be changed by adding a column. Fortunately the use of this operation is rather rare.

Name	Office	Shoesize	Tel	...
Jane	19	7	2341	...
Fred	17a	43	2314	...
Bill	17b	9	4123	...
...	...	...	...	...

<annotation>

Figure 2: A simple annotation

2. This is probably a European shoe size
3. This is way too big (for a shoe size)
4. This is way too big (for Fred)
5. The normal range is 5-14

All of these are perfectly valid attachments, but the referent requires some explanation. In (1) the annotation has nothing to do with the location; it is an annotation on the value that could be attached to any occurrence of the number 43. By contrast, in (2) the annotation has to do with the column (or domain) and could reasonably be attached to any other occurrence of 43 in the **Shoesize** column. Similarly for (3), though this is less informative. The only annotation that is specifically about the relationship between the value, 34, and the location, the **Shoesize** field of the **Fred** tuple, is (4). Finally, (5) is an annotation that should be attached to the schema, rather than the data; however the schema is not normally seen in views of the data.

To return to the specification of attachment of external annotations, consider first how one would specify the attachment in Figure 2. One would provide the name of the table, identifier for the tuple, and the name (**Shoesize**) of the field within the tuple. The tuple identifier could be a key, or it could be the internal tuple identifier provided by the database management system. It is regarded as bad practice to modify a key and it is impossible to change an internal tuple identifier (they last for the lifetime of a tuple and are never re-used). Thus the (table name, tuple identifier, field name) triple should serve as a stable “co-ordinate system” for attachment in a well-defined relational database.

The same idea can be extended to hierarchically structured data such as XML; the details are straightforward [BDF<sup>+</sup>02] and are not given here. The point is first that the designers of new data sets should not only describe the schema, they should also describe a co-ordinate system for the attachment of annotations. Second, if the data set is updated, the updates should respect the co-ordinate system. One should not, for example, recycle identifiers or field names.

As we have noted earlier, specifying co-ordinates for temporal and spatial data is a major issue and beyond the scope of this survey.

### 4.3 Annotation and relational databases

Relational databases have had an extraordinarily successful history of commercial success and fertile research. It is not surprising, therefore, that database researchers would first attempt to understand annotation in the context of relational databases. One of the immediate challenges here is to understand how annotations should propagate through queries. If one thinks of annotation as some form of secondary mark-up on a table, how is that mark-up transferred to the result of a query. If, for example, an annotation calls into

question the veracity of some value stored in the database, one would like this information to be available to anyone who sees the database through a query of *view*.

Equally important is the issue of backwards propagation of annotations. We consider, as a loose analogy, the BioDAS system, based on the DAS system discussed in section 3.1. The users see and annotate the data using some GUI, which we can loosely identify with a database view. The annotation is transferred backwards from the GUI to an annotation on some underlying data source and is then propagated forwards to other users of the same data. Following the correspondence, the question is how does an annotation propagate through a query both backwards and forwards?

It is easy to write down the obvious set of rules for the propagation of annotation through the operations of the relational algebra. However, because of nature of relational algebra, inverting these rules is non-deterministic. An annotation seen in the output could have come from more than one place in the input. To take one example: suppose one places an annotation on some value in the output of a query  $Q$ . Of all the possible annotations on the source data (the tables on which  $Q$  operates) is there one which causes the desired annotation – and only that annotation – to appear in the output of  $Q$ . The complexity of this and several related annotation problems have been studied in [BKT02] which also shows the connection with the view delation problem.

In [BCTV04] a practical approach is taken to annotation in which an extension of SQL is developed which allows for explicit control over the propagation of annotations. Consider the following simple join query

```
SELECT  R.A, R.B, S.C
FROM    R, S
WHERE   R.B = S.B
```

Suppose the source is annotated. Presumably an annotation on a  $B$  value of  $R$  should propagate to the  $B$  field of the output, because  $R.B$  is given as the output. But should an annotation on a  $B$  field of  $S$  also be propagated to the  $B$  field of the output? The structure of the SQL indicates that it should not, but the query obtained by replacing the first line by

```
SELECT R.A, S.B, S.C
```

is equivalent, so maybe the answer should be yes. The idea in [BCTV04] is to allow the user to control the flow of annotation by adding some further propagation instructions to the SQL query. The paper shows how to compute the transfer of annotations for the extended version of SQL and demonstrates that for a range of realistic queries the computation can be carried out with reasonable overhead.

The work we have described so far has been limited to annotating individual values in a table. Recently Geerts *et al.* [GKM05] have taken a more sophisticated approach to annotating relational data. What they point out is that it is common to want to annotate *associations* between values in a tuple. For example, in the query above one might want to annotate the  $A$  and  $B$  fields in the output with information that they came from input table  $R$  and the  $B$  and  $C$  fields with information that they came from table  $S$ . To this end the introduce the concept of a *block* – a set of fields in a tuple to which one attaches an annotation and a *colour* which is essentially the content or some property of the annotation. They also investigate both the theoretical aspects and the overhead needed to implement the system.

## 5 Acknowledgements

Robert Mann, Amos Storkey, Bonnie Webber

## A The cited databases and systems

### A.1 UniProt

UniProt (Universal Protein Resource) is a curated protein sequence repository created by combining three other existing databases of protein information. It consists of some 100,000 entries each of the order of a few kilobytes. The entries contain bibliographic and taxonomic data, comments and annotations on the sequence itself.

This is a widely-cited major resource for biologists. About 150 people are currently involved in the maintenance of UniProt.

### A.2 The Supercosmos Sky Survey

This is a database [GSS<sup>+</sup>02] in which a number of astronomical observations are reduced to a very large but simple table of some  $10^8$  rows and 300 columns. Each row represents an astronomical “object”: its position in the sky (which is used as a key), parameters to describe its shape and luminosity at various wavelengths, etc.

### A.3 The CIA World Factbook

This well-known database [CIA] contains geographic and demographic data for the countries of the world. Its maintenance appears to be as labour-intensive as any of the other database we have cited in this paper, and is presumably constructed from a mixture of published statistics and intelligence reports.

If the curators of the World Factbook are guilty of not citing their sources, the rest of the world is even more guilty. Any description of the population of a country of the form “31,842 (July 2001 est.)” is likely to have been taken from the World Factbook, and it is almost always given without attribution.

### A.4 OMIM

This is an interesting example of a database which, until 1998, was a publication that was revised every two years or so. It is a database of genetic traits and disorders consisting of ...

### A.5 The International Movie Database

## References

- [BCTV04] Deepavali Bhagwat, Laura Chiticariu, Wang Chiew Tan, and Gaurav Vijayvargiya. An annotation management system for relational databases. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases*, pages 912–923, Toronto, Canada, 2004. Morgan Kaufmann.
- [BDF<sup>+</sup>02] Peter Buneman, Susan Davidson, Wenfei Fan, Carmem Hara, and Wang-Chiew Tan. Keys for XML. *Computer Networks*, 39(5):473–487, August 2002.

- [BKT02] Peter Buneman, Sanjeev Khanna, and Wang-Chiew Tan. On the Propagation of Deletions and Annotations through Views. In *Proceedings of 21st ACM Symposium on Principles of Database Systems*, Madison, Wisconsin, 2002.
- [Bus45] V. Bush. As we may think. *The Atlantic Monthly*, June 1945.
- [Cho02] Byron Choi. What are real DTDs like? In *WebDB*, pages 43–48, 2002.
- [CIA] The world factbook. <http://www.cia.gov/cia/publications/factbook/>.
- [GKM05] Floris Geerts, Anastasios Kementsietsidis, and Diego Milano. "MONDRIAN: Annotating and querying databases through colors and blocks". Technical report, LFCS, School of Informatics, University of Edinburgh, 2005.
- [GSS<sup>+</sup>02] Jim Gray, Don Slutz, Alexander S. Szalay, Ani R. Thakar, Jan vandenBerg, Peter Z. Kunszt, and Christopher Stoughton. Data mining the SDSS Skyserver database. Technical Report MSR-TR-2002-01, Microsoft, 2002.
- [HVS03] S. Handschuh, R. Volz, and S. Staab. Annotation for the deep web. *Intelligent Systems*, 18(5):42–48, 2003. See also IEEE expert.
- [KKPR01] J. Kahan, M.-R. Koivunen, E. Prud'hommeaux, and R.R. Swick. Annotea: an open RDF infrastructure for shared Web annotations. In *Proceedings of the tenth international conference on World Wide Web*, pages 623–632. ACM Press, 2001.
- [MD99] D. Maier and L. Delcambre. Superimposed information for the internet. In *Proc. WebDB*, 1999.
- [RDS04] R. Overbeek, T. Disz, and R. Stevens. The SEED: a peer-to-peer environment for genome annotation. *Comm. ACM*, 47(11):46–51, 2004.
- [Rém93] Didier Rémy. Type inference for records in a natural extension of ML. In Carl A. Gunter and John C. Mitchell, editors, *Theoretical Aspects Of Object-Oriented Programming. Types, Semantics and Language Design*. MIT Press, 1993.
- [SSD02] L.D. Stein, S.Eddy, and R. Dowell. Distributed Sequence Annotation System (DAS), 2002. <http://biодas.org/>.
- [TM97] Henry S. Thompson and David McKelvie. Hyperlink semantics for standoff markup of read-only documents. In *SGML '97 Conference Proceedings*, pages 227–229, Barcelona, Spain, 1997.
- [Wei03] H.J.R. Weintraub. The need for scientific data annotation. *Abstracts of Papers of the American Chemical Society*, 226:303–304, 2003.
- [ZGG<sup>+</sup>03] J. Zhao, C. Goble, M. Greenwood, C. Wroe, and R. Stevens. Annotating, linking and browsing provenance logs for e-science. In *Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data*, Sanibel Island, Florida, 2003. Online proceedings (at ISWC 2003).
- [ZM03] Z. Zhihong and Z. Mingtian. A distributed description logic approach to semantic annotation. In *Parallel and Distributed Computing, Applications and Technologies, 2003. (PDCAT)*, 2003.