Annotating scientific data: why it is important and why it is difficult.

Rajendra Bose
University of Edinburgh

Peter Buneman
University of Edinburgh

Denise Ecklund
Objective Technology Group

Abstract

Annotation of existing data is becoming a standard tool in many branches of e-science. Increasingly, databases are being built to receive annotation, and other tools are being developed to annotate existing databases. Annotation is becoming an important part of communication among scientists. In this paper we review various kinds of annotation systems and describe the importance of designing databases in such a way that they can receive annotation. This includes designing *extensible* databases and the need for some form of *co-ordinate system* for the attachment of annotations.

1 Annotation: adding to existing structure

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 for dissemination of knowledge. 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 apparent 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. One example is UniProt (Universal Protein Knowledgebase) [ABW+04], which is supported by upwards of 100 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 similar to the way sticky notes are now attached to PDF documents and web pages. [MD99], discusses "superimposed information" – 'data placed over existing [base] information sources to help organise, access, connect and reuse information elements in those sources."

The importance of annotation was, as with many so many other issues, recognised as important by Vannevar Bush [Bus45] who says "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." Annotation is ubiquitous on the Web – in Wikis, review/opinion sites, newsgroups, etc. It is now 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 some understanding of this process and the associated pitfalls and technological requirements.

1.1 A framework for annotation

In order to compare various types of annotation systems we suggest an informal framework that consists of the following basic components:

An annotation is some set of data elements that is added to an existing base or target that possesses structure. In order to create an annotation, some form of attachment point is used implicitly or explicitly. We shall use the term *co-ordinate system* for the mechanism for describing the attachment point. Some care is needed, during database design, in making sure that the co-ordinate system is durable. Moreover one frequently finds several co-ordinate systems in simultaneous use. Understanding the mappings between the co-ordinate systems is seldom straightforward. Let us consider some examples (which are discussed in more detail later in this paper):

- Cartographic data. The use of multiple co-ordinate systems (such as longitude and latitude vs. a local grid) is commonplace in cartography and mappings between such systems are well understood. The point of attachment to a image representation of a map is specified by such a co-ordinate system. However, recent map data now relies on some form of object-oriented representation of cartographic features, and attachment is, presumably, to some object identifier. Note that there is a subtlety about what is being annotated; moreover the correspondence between the two co-ordinate systems is no longer a simple 1-1 mapping.
- Molecular biology. This has moved in the reverse direction. The original co-ordinate systems were the gene identifiers used in the various databases. Only recently have the linear (chromosome, offset) co-ordinates determined by genetic sequencing been discovered and, once again, the mapping is not 1-1.

The various aspects of annotation are illustrated in Figure 1. The genome column summarises two of the co-ordinate systems in use in that domain. The HBP column shows that while the co-ordinate system may be simple, the attachment process is not. AstroDAS is interesting in this context because its purpose is precisely to reconcile the co-ordinate systems in a variety of database. It is a database in which the annotations of the objects are the co-ordinates (typically relational keys) of objects in other catalogues. The intention of this annotation is to support the more general forms of cross-database annotation, which we describe below.

In the remainder of the report, we present a series of examples of scientific annotation in Section 2, and refer to our basic framework to help compare them. In Section 3 we discuss some key concepts of database annotation and suggest them as topics for further research.

2 Examples of scientific annotation systems

2.1 UniProt database annotation

Perhaps the most well-known examples of database annotation are to be found in bioinformatics and in the design of information systems like UniProt, which consists of an assemblage of databases including Swiss-Prot, an annotation database produced by specialist curators, and TrEMBL, which provides automated annotations for proteins (`http://www.ebi.uniprot.org/about/background.shtml`). These databases were created to to disseminate protein sequence data and associated analyses. They were designed specifically to receive annotation.

The curators of Swiss-Prot [BA00] are quite specific about which fields in the database they regard as annotation and which are "core data". In figure 2, the boxed areas are those classified as annotation, the publication and taxonomic entries, perhaps because the Swiss-Prot organisation was not responsible for its creation, are regarded as core data. This database illustrates a number of interesting aspects of annotation which we discuss further in Section 3.1. Note that several of the fields have "pointers" to entries in other databases, which provide mappings between co-ordinate systems.

	HBP image annotation	Astrodas	Genome(human)	
Annotation target:				
what	brain image	celestial object in astronomy catalogue	gene	
structure	pixel array	catalogue (RDBMS schema)	database schema	sequence
co-ordinate system	(x, y) co-ordinates of pixel	catalogue + object id (relational key)	object id ("accession number")	chromosome + offset
Annotation:				
what	domain-specific term (ontology element)	mappings to other catalogues	free and structured text	
location of attachment	2D contour	catalogue+id	key + attribute	chromosome + start/stop positions
purpose	link Web-accessible images to and find instances	share assertions across different catalogues	general comments and classification	

Figure 1. Comparison of annotation systems

2.2 Genome annotation

In contrast to Swiss-Prot, BioDAS [SED02] is an external annotation system for a variety of databases. The Distributed sequence Annotation System (DAS, later BioDAS) protocol [SED02] was designed to serve this purpose. The architecture is that of an "open" client-server annotation system communicating via an extension of HTTP; significantly, the addition of new annotation servers requires only minimal coordination between data providers.

BioDAS includes a client capable of requesting both (1) the coordinate system or "reference map" of base pairs for a specific genome from a reference server, and (2) a set of uniquely identified sequence annotations, anchored to the reference map by start and stop values, from an annotation server [DJD+01]. The client requests are URLs that are constructed according to simple conventions in an HTTP request; the servers respond to these requests with a Generic (genomic) Feature Format (GFF)-derived XML document [Ens06]. The Ensembl Genome Browser web application (`www.ensembl.org`) employs DAS functionality.

IBM developerWorks uses a similar client/server architecture to provide a general solution for annotation of digital data (`http://www-106.ibm.com/developer\-works/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. [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.

Other systems exist for annotating genomic data. The SEED project [RDS04] is similar to BioDAS, 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+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 labelling of annotation workflows.

```
ID   11SB_CUCMA     STANDARD;      PRT;   480 AA.
AC   P13744;
DT   01-JAN-1990, integrated into UniProtKB/Swiss-Prot.
DT   01-JAN-1990, sequence version 1.
DT   21-MAR-2006, entry version 51.
DE   11S globulin beta subunit precursor [Contains: 11S globulin gamma
DE   chain (11S globulin acidic chain); 11S globulin delta chain (11S
DE   globulin basic chain)].
OS   Cucurbita maxima (Pumpkin) (Winter squash).
OC   Eukaryota; Viridiplantae; Streptophyta; Embryophyta; Tracheophyta;
OC   Spermatophyta; Magnoliophyta; eudicotyledons; core eudicotyledons;
OC   rosids; eurosids I; Cucurbitales; Cucurbitaceae; Cucurbita.
OX   NCBI_TaxID=3661;
RN   [1]
RP   NUCLEOTIDE SEQUENCE [MRNA].
RC   STRAIN=cv. Kurokawa Amakuri Nankin;
RX   MEDLINE=88166744; PubMed=2450746;
RA   Hayashi M., Mori H., Nishimura M., Akazawa T., Hara-Nishimura I.;
RT   "Nucleotide sequence of cloned cDNA coding for pumpkin 11-S globulin
RT   beta subunit.";
RL   Eur. J. Biochem. 172:627-632(1988).
RN   [2]
RP   PROTEIN SEQUENCE OF 22-30 AND 297-302.
RA   Ohmiya M., Hara I., Mastubara H.;
RT   "Pumpkin (Cucurbita sp.) seed globulin IV. Terminal sequences of the
RT   acidic and basic peptide chains and identification of a pyroglutamyl
RT   peptide chain.";
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: Belongs to the 11S seed storage protein (globulins)
CC       family.
CC   ---------------------------------------------------------------------
CC   Copyrighted by the UniProt Consortium, see http://www.uniprot.org/terms
CC   Distributed under the Creative Commons Attribution-NoDerivs License
CC   ---------------------------------------------------------------------
DR   EMBL; M36407; AAA33110.1; -; mRNA.
DR   HSSP; P04776; 1FXZ.
DR   InterPro; IPR006045; Cupin_1.
DR   InterPro; IPR007113; Cupin_region.
DR   InterPro; IPR011051; Cupin_RmlC_type.
DR   InterPro; IPR006044; Seedstore11s_pln.
DR   Pfam; PF00190; Cupin_1; 2.
DR   PRINTS; PR00439; 11SGLOBULIN.
DR   PROSITE; PS00305; 11S_SEED_STORAGE; 1.
KW   Direct protein sequencing; Pyrrolidone carboxylic acid;
KW   Seed storage protein; Signal; Storage protein.
FT   SIGNAL        1     21
FT   CHAIN        22    296       11S globulin gamma chain.
FT                                 /FTId=PRO_0000032028.
FT   CHAIN        22    480       11S globulin beta subunit.
FT                                 /FTId=PRO_0000032027.
FT   CHAIN       297    480       11S globulin delta chain.
FT                                 /FTId=PRO_0000032029.
FT   MOD_RES      22     22       Pyrrolidone carboxylic acid.
FT   DISULFID    124    303       Interchain (between gamma and delta
FT                                chains) (Potential).
FT   CONFLICT     27     27       S -> E (in Ref. 2).
FT   CONFLICT     30     30       E -> S (in Ref. 2).
SQ   SEQUENCE   480 AA;  54626 MW;  BCD8A83DD1AED93C CRC64;
     MARSSLFTFL CLAVFINGCL SQIEQQSPWE FQQSEVWQQH RYQSPRACRL ENLRAQDPVR
     RAEAEAIFTE VWDQDNDEFQ CAGVNMIRHT IRPKGLLLPG FSNAPKLIFV AQGFGIRGIA
     IPGCAETYQT DLRRSQSAGS AFKDQHQKIR PFREGDLLVV PAGVSHWMYN RQGSDLVLIV
     FADTRNVANQ IDPYLRKFYL AGRPEQVERG VEEWERSSRK GSSGEKSGNI FSGFADEFLE
     EAFQIDGGLV RKLKGEDDER DRIVQVDEDF EVLLPEKDEE ERSRGRYIES ESESENGLEE
     TICTLRLKQN IGRSVRADVF NPRGGRISTA NYHTLPILRQ VRLSAERGVL YSNAMVAPHY
     TVNSHSVMYA TRGNARVQVV DNFGQSVFDG EVREGQVLMI PQNFVVIKRA SDRGFEWIAF
     KTNDNAITNL LAGRVSQMRM LPLGVLSNMY RISREEAQRL KYGQQEMRVL SPGRSQGRRE
//
```

Figure 2. An entry from UniProt

2.3 Annotating biomedical images

Some systems are designed to create web-accessible collections of annotated biomedical images. Gertz et al. [GSG+02] develop a graph model of annotations for use in the Human Brain Project (HBP): 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; one such relationship is "annotation of". They also develop a framework for querying annotation graphs based on path expressions and predicates, which they test in a prototype system. Column (1) of Figure 1 refers to HBP image annotation.

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

2.4 AstroDAS: Annotating astronomy catalogues

Over the past several decades, databases or catalogues of celestial object observations, recorded by disparate telescopes and other instruments over various time periods, have migrated online. Central to the astronomical community's concept of a global "Virtual Observatory" is the ability to identify records in these different catalogues as referring to the same celestial object. Because the recorded location of a celestial object may vary slightly from catalogue to catalogue due to unavoidable measurement error at the instrument level, the general catalogue matching problem cannot be solved by spatial proximity alone, and some researchers develop their own complex algorithms for matching celestial objects across different catalogues.

To provide astronomers with the ability to share their assertions about matching celestial objects directly with their colleagues, we have created prototypes for AstroDAS, a distributed annotation system partly inspired by BioDAS [BMPR06]. AstroDAS features an annotation database with a web service interface to store and query annotation, and resolves queries on astronomy catalogues using mapping tables that are dynamically constructed from annotations of celestial object matches. The AstroDAS prototypes complement the existing OpenSkyQuery system for distributed catalogue queries.

The ultimate aim of AstroDAS is similar to the goal of the earlier BioDAS: to record and share scientific assertions with a wider com-

munity. Whereas biologists use annotation in BioDAS to interpret the DNA sequences in a genome, however, astronomers seek to share the mapping of entities derived from their research across established scientific databases. Specifically, astronomers want to be able to share their identification of matching celestial objects within the existing federation of disparate catalogues.

3 Concepts and research topics in database annotation

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, which we discuss further in the following sections. We also discuss annotation queries, research topics in relational database annotation, and annotating annotations.

3.1 Annotation and the evolution of database structure

We return to the Swiss-Prot example of annotation within databases: Figure 2 shows a single entry in Swiss-Prot. It is debatable what one should classify 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 Swiss-Prot now make extensive use of database technology and that what is exported in Figure 2 is a "rendering" or database view of the internal data. While database management systems provide some help with structural evolution, it is always problematic. In this respect, databases designed with conventional (relational or object-based) structuring tools offer better prospects for extensibility than XML structured with DTDs or XML-Schema which are, at heart, designed to express the serialisation of data.

3.2 Location and attachment of annotations

The annotations in the `CC` fields in Figure 2 appear to refer to the entire Swiss-Prot 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 also publishes under the alias John Doe". One could imagine inserting this text in the text of the Reference Author (`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

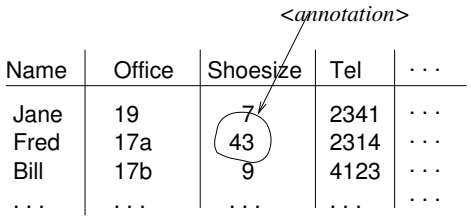

Figure 3. A simple annotation

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-normalised" 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.

We have already noted that annotations are sometimes placed inside the annotated object and sometimes outside and that many annotations are, for reasons of database security, necessarily stored externally. External annotations require a 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 3. and consider some possibilities for ⟨*annotation*⟩:

1. This is a prime number
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 annotations, 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 frequently transformed in views of the data, and the attachment of such annotations may be problematic.

To return to the specification of attachment of external annotations, consider first how one would specify the attachment in Figure 3. 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 reused). 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+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.

3.3 Querying annotations

Work in the ediKT project at Edinburgh (`http://www.edikt.org`) with 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.

3.4 Annotating relational databases: recent work

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 2.2. 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 deletion 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

$$\begin{array}{ll} \mathsf{SELECT} & R.A,\ R.B,\ S.C \\ \mathsf{FROM} & R, S \\ \mathsf{WHERE} & R.B = S.B \end{array}$$

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

$$\mathsf{SELECT}\ R.A,\ S.B,\ S.C$$

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.* [GKM06] have taken a more sophisticated approach to annotating relational data. What they point out is 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. However, as we have indicated in Section 3.2 that attachment may be even more complex, requiring associations between data and schema, for example.

3.5 Provenance and Annotation

The topic of *data provenance* is of growing interest and deserves separate treatment. However there are close connections with annotation. One view of the connection is that provenance – information about the origins of a piece of data – is simply another form of annotation that should be placed on data. It is certainly true that there are many cases where provenance information is added after the creation of data. However, it would be much better if provenance were captured automatically, in such a way that it becomes an intrinsic part of the data.

A more interesting connection is to be found in [BCTV04] and related papers. Much data in scientific databases (e.g. the "core data" in Figure 2) has been extracted from other databases. If the data in the source database has been annotated, surely the annotations should be carried into the receiving database. If the receiving database is a simple view of the source data, then the mechanisms described in [BCTV04], or some generalisation of them, should describe both provenance and how annotations are to be copied. However, manually curated databases are more complex than views, and in this case understanding the movement of annotations is still an open problem.

4 Conclusions

Although we have not done enough work to substantiate this claim, we believe it likely that most of the 858 molecular biology databases listed in [Gal06] involve some form of annotation. Moreover, as we have tried to indicated, annotation is of growing importance in other areas of scientific research. The success of new databases will depend greatly on the degree to which they will support annotation. In this respect, the following points are crucial both in database design and in systems architecture:

- the provision of a co-ordinate system to support the attachment of annotations,
- the linkage or mapping of that co-ordinate system to other, existing, co-ordinate systems, and
- the need for extensibility in databases that are designed to receive annotations.

In each of these areas, there is further research needed. Moreover, annotations often express complex relationships between schema and data. To bring this into a uniform framework is a challenge for both database and ontology research.

5 Acknowledgements

We would like to thank Mark Steedman, Robert Mann, Amos Storkey, Bonnie Webber, as well as the members of the Database Group in the School of Informatics. This survey has been supported in part by the Digital Curation Centre, which is funded by the EPSRC e-Science Core Programme and by the JISC.

References

[ABW+04] R. Apweiler, A. Bairoch, C.H. Wu, W.C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M.J. Martin, D.A. Natale, C. O'Donovan, N. Redaschi, and L.S. Yeh. Uniprot: the universal protein knowledgebase. *Nucleic Acids Res*, 32:D115–D119, 2004.

[BA00] A. Bairoch and R. Apweiler. The SWISS-PROT protein sequence database and its supplement TrEMBL. *Nucleic Acids Research*, 28:45–48, 2000.

[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+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.

[BMPR06] R. Bose, R. Mann, and D. Prina-Ricotti. Astrodas: Sharing assertions across astronomy catalogues through distributed annotation. In *International Provenance and Annotation Workshop (IPAW 2006)*, pages 193–202, Chicago IL, 2006. Springer, LNCS 4145.

[Bus45] V. Bush. As we may think. *The Atlantic Monthly*, June 1945.

[DJD+01] R Dowell, R Jokerst, A Day, S Eddy, and L Stein. The distributed annotation system. *BMC Bioinformatics*, 2(7), 2001.

[Ens06] Ensembl: Information: Data: External data: About the distributed annotation system (das), 2006. http://www.ensembl.org/info/data/external_data/das/index.html.

[Gal06] Michael Y. Galperin. The molecular biology database collection: 2006 update. *Nucleic Acids Research*, 34:D3–D5, 2006,. Database issue.

[GKM06] Floris Geerts, Anastasios Kementsietsidis, and Diego Milano. Mondrian: Annotating and querying databases through colors and blocks. In *ICDE*, page 82, 2006.

[GSG+02] Michael Gertz, Kai-Uwe Sattler, Frederic Gorin, Michael Hogarth, and Jim Stone. Annotating scientific images: A concept-based approach. In Jessie Kennedy, editor, *14th International Conference on Scientific and Statistical Database Management (SSDBM 2002)*, pages 59–68, Edinburgh, Scotland, 2002. IEEE Computer Society.

[MD99] D. Maier and L. Delcambre. Superimposed information for the internet. In *WebDB 1999 (Informal Proceedings)*, pages 1–9, 1999. http://www-rocq.inria.fr/ cluet/WEBDB/maier.pdf.

[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.

[SED02] Lincoln D. Stein, Sean Eddy, and Robin Dowell. Distributed sequence annotation system (das) specification version 1.53. Technical report, 21 March 2002 2002.

[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+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).