Ubiquitous Data

Gavin Bierman[1] Peter Buneman[2] and Philippa Gardner[3]

Draft of July 25, 2003

Abstract. Distributed database technology has long recognised that it is often expedient to move process to data rather than data to process, and this is also part of the rationale for the Grid. With current technology, a distributed process or query involves only a small number of sites. However the trend is towards increasingly distributed data; and when our queries require data from thousands or millions of sites, ubiquitous computing and ubiquitous data will become one and the same.

Background. The topic of databases emerged from the interplay of two requirements. The first was to have a simple *abstraction* for structured data; the second was to make the manipulation of large quantities of this data both robust and *efficient*. The relational model was invented for the first of these. It proposed a simple tabular data model and an algebra of operations on tables. What is remarkable about this algebra is that it also met second requirement, in that its equational theory served as a basis for query optimisation and queries could be efficiently implemented through query rewriting, indexing and join techniques. This accounted for the success of relational databases; the details are described in any good database textbook.

Despite this success, it is clear that the new paradigm of ubiquitous computing places new demands on database systems. Indeed many of the assumptions hitherto taken for granted have to be readdressed. Two significant problems arising from the assumption of a ubiquitous computing base that we focus on here are the *distribution* of data, and the very structure of the data itself. We consider these problems in turn, highlighting some of the issues, as well as suggesting how some of our research is addressing the problem.

One effect of assuming a ubiquitous computing base is that the traditional database aim of capturing *all* the information relevant to some “enterprise” in a single database, is simply unrealistic. The resources needed to answer a query will be found in a variety of databases distributed across a network. Moreover, it is remarkably expensive to move large volumes of data. Thus it is often better to leave data *in situ* and create appropriate views of the source data by moving the appropriate programs close to the data.

Of course, the need for data integration and distributed query processing has long been recognised in database research: there has been considerable work on querying distributed data sources; indeed giving rise to one of the early, practical, examples of mobile code. In distributed query optimisation one may decide to decompose a query using, say, the formalism of the relational algebra and distribute fragments of the query over a number of sites. More recently, the need to move programs to data is an essential part of the philosophy behind the Grid. In both databases and the Grid, the number of data locations in a typical distributed query is very small.

New models for distributed data and computation. The move to a ubiquitous computing base gives focus to a new issue: *scale*. Over the last ten years there has been a steady progression towards higher degrees of distribution and increasing mobility of data. An interesting case study is the field of bioinformatics in which there are some 500 public databases and many times that number in commercial use. Although the computing substrate in bioinformatics is far from ubiquitous, the trends towards ubiquity are evident. For example, (a) Only an handful of these databases contain source experimental data. The others are constructed by a process of filtering, transforming, cleaning and annotating data in other databases. (b) The structure of the databases evolves to capture new forms of scientific knowledge. (c) Many of the databases make use of purpose-built data formats. Our query languages and optimisation techniques must be adapted to cope with these. (d) An increasingly important activity is monitoring other data sources for the appearance of new information. Some important scientific discoveries have been made by monitoring the “stream” of genetic sequence data.

In addition to challenging conventional database technology, bioinformatics raises new issues that have largely been ignored by database research. One is *provenance*. Given that fragments of data are being copied between databases

[1] University of Cambridge Computer Laboratory, Cambridge, CB3 0FD. `gmb@cl.cam.ac.uk`

[2] LFCS, Division of Informatics, University of Edinburgh, Edinburgh EH9 3JZ. `opb@inf.ed.ac.uk`

[3] Computing Department, South Kensington Campus, Imperial College, London, SW7 2AZ. `pg@doc.ic.ac.uk`

on an unprecedented scale, how do we record where these fragments have come from? Provenance essential for data quality almost any form of data exploration and it is closely related to an increasingly important form of scientific communication, *annotation*, in which knowledge is disseminated by by a process of overlaying annotations on existing data and making those annotations visible to others. This brings up new challenges and new theoretical underpinnings for database research. At the same time we have to do this in systems in which the access rights to data and ownership issues are fully understood. How do we guarantee that in a highly distributed environment, in which the data is highly distributed and in which the structure may only be partially understood, that access privileges to some piece of data are appropriately preserved as it is coped and transformed?

Confluence of models The situation in biology is a hint of things to come. The authors have, in various ways, been involved in the development of new models for both data and computation that result from the need to support highly distributed and mobile data. These range from object-oriented models and models for semistructured data and XML to process calculi and other models for distributed computation on the Web. Developing the analysis techniques, languages and tools for these various data models is by no means straightforward. We have had to revisit all the work that has been developed for databases – the query languages, type systems, the storage and optimisation techniques – and rework them for these new models.

Recently a ‘trees with pointers’ model has been emerged as a possible abstraction of the semistructured model of data. Given this data model one can define a logic and pattern-matching language for describing, analysing and manipulating semistructured data. The spatial logic can be used to reason about the horizontal structure of trees (such as schemas for semistructured data), the more usual vertical path structure (such as path expressions), and the pointer structure (such as the IDs and IDREFs of XML). A very similar model also promises to provide a substrate in which one can fully describe provenance and annotation of data: a simple language based on pattern-matching form the basis of data transformations in which the source of any data element is explicit.

Of course, computer scientists model almost everything with trees, so is the connection between semistructured data and process calculi at all substantive? We believe so. Languages that have been derived from the database view of semistructured data (notably query languages and programming for XML) turn out to be closely related to ambient logics for querying graphs. Also there has been activity in database development to embed processes in data, producing some close relationships with process calculi. At the same time there is are continuing developments with type systems that embrace both the programming and data abstractions that have emerged.

The future of ubiquitous data. We conclude by speculating on how databases will be used in ubiquitous computing. Consider a distributed health-care system. There are huge problems associated with keeping patient records. One increasingly popular proposal is that each individual should “own” their medical record and give out all or part of the record to trusted parties on demand. A medical record of the future will contain vast amounts of sensor data and probably one’s genetic sequence data. Now suppose that a researcher wants to correlate, say, the occurrence of a cardiovascular condition with some genetic structure. The task is not to integrate a few databases, but literally millions of them. This task cannot be performed by any form of centralised processing; it requires code to be moved to the source data and run, locally, as a trusted process. It will require the efficient monitoring of data streams and localised querying and integration. And unless we are confident that the systems we are using are robust and can be guaranteed to have the appropriate properties, it is unlikely that we shall let them come anywhere near our medical records!

Summary. Models of distributed processing and distributed database access have, until quite recently, been entirely separate topics. In next few years we may expect them to converge and we may expect to see the following advances:

- Querying, transforming and integrating highly distributed data.
- Unifying models of mobility for both processes and data that capture provenance.
- Ubiquitous monitoring of data.
- Support from programming language design to facilitate such application development .
- The confluence of ideas from semi-structured data community and the process algebra community.

The challenge. In 15 years find a sound theoretical underpinning for languages and data associated with the web.