Tenth ERCIM Database Research Group Workshop: *Heterogeneous Information Management*
4-5 November 1996, Prague, Czech Republic

Federating Legacy Databases about People - in practice

Brian J Read

b.j.read@rl.ac.uk

CLRC Rutherford Appleton Laboratory
Chilton, Didcot, Oxon OX11 0QX, UK

Abstract

A single corporate database is rarely feasible due to a legacy of heterogeneous applications and an environment of continually changing requirements. Sharing basic information about people within an organisation is a compelling need but presents practical problems. A People Table can act as a focus for separate applications to share and exchange data of mutual interest. Unambiguous identification of individuals cannot be perfect. Such a federated information system requires clearly specified protocols for data maintenance and to minimise the effect of semantic differences.

1. THE DATABASE MYTH

An organisation might desire to have a single, unified information system to store and manage all its data. (It could be a distributed system, but it would still have a single data model with homogeneous semantics.) However such an ideal is rarely achieved. The two main reasons are:

a) The operating environment evolves. Business requirements change. Organisations restructure or perhaps merge. Even if much effort and time can be invested in creating a single system, timely response within one complex system to multiple external forces is difficult. The corporate database is elusive. It needs a strong central political will and a stable (moribund?) business.

b) There is rarely a "green field" situation to start with. Normally, applications that process data already exist. In practice there is a problem coping with disparate legacy systems. This is not just the inconvenience of different hardware and data management software. More fundamentally, differing semantics (such as different grading schemes or inconsistent business rules) are the real difficulty in the integration of heterogeneous data.

Thus the single, corporate database is often a false ideal. The pragmatic alternative is federation of applications, these having limited, negotiated integration determined by defined benefit and cost.

2. MOTIVATION: THE PEOPLE PROBLEM

This paper concentrates on the task of integrating systems processing data about people. It draws on the practical experience of managing such data within CLRC. This is a scientific research laboratory of 1800 staff located at two main sites: Rutherford Appleton Laboratory (RAL) in Oxfordshire and Daresbury Laboratory (DL) three hundred kilometres to the north in Cheshire.

Like many medium-to-large organisations, CLRC has numerous sections or departments that keep records of data relevant to their own particular activity and the people involved. Thus, for example, Personnel Group stores career histories of staff, Payroll Section keeps employee details, the library records borrowers, the administrators of various computer systems register sets of users, Security issues site passes to staff, contractors and visitors, Transport Section authorise driving permits, secretaries hold distribution lists of their group or department, the large scientific facilities (eg lasers, synchrotrons) schedule experimental time slots for investigators, ... The list is almost endless! And all these try to hold much the same data about people such as their name, telephone number, location and department.

An obvious conclusion may be drawn. It is that there is much duplication of effort in maintenance of personal data. Propagation of updates is erratic or non-existent. Mailing lists, for example, become out-of-date. Could there at least be a mechanism for sharing data of common value?

3. THE REALITY

Ideally, the solution is that all the administrative activities mentioned should use a single database containing details of people in the organisation. In practice, there are difficulties. There will be existing legacy systems for which the cost and effort of re-implementation in an integrated system are neither available nor justified. There may not be the political will - data owners can be very possessive of "their" data and reluctant to co-operate.

Another difficulty is in defining the scope of any integration. Each application probably holds data to a lesser or greater extent on a different set of people. For example, Payroll Section is not concerned with visiting scientists, but they have an office and telephone. Personnel Group does not hold records of contractors' staff, but they are issued with site passes. Many university scientists visit only for a few days to conduct experiments. The library lends to other than just CLRC staff. The super-computing service has users located all over the country.

Thus we are concerned with many, partially overlapping sets of people. A boundary needs to be drawn. Thought has be given to defining just who should be in an organisation's *people database*. There may be no advantage in including external people of interest to only a single activity. (In that case a, possibly very large, "addendum" people table would be maintained by the application concerned.)

Another snag is different applications regard the same person in different rôle so that supposedly common attributes are rôle dependent.

4. THE WAY FORWARD

What is needed is a practical framework in which the various applications can make use of others' data. The information most in demand is a person's work location, telephone number, e-mail address and position in the organisational structure. However there will be other data collected by one activity that can be of value to another. Given that a single, comprehensive database is not practical, and that the information of mutual interest is maintained by different units in the organisation, procedures need to be established whereby either data may be exchanged (copied) from one unit to another (sufficiently often), or provide for one unit being able to query directly another's database. In either case, it is crucial for one application correctly to link the other's people data to its own. That is, a common, unique way of identifying people is needed.

The solution is a central, people file comprising a superset of all those from the individual applications. The latter subscribe to the federated system by maintaining a link between their own people records and this People Table. Such foreign key links then allow queries across multiple applications.

Note that an application opts into this scheme. For some, that have little in common with the others, the benefit to its manager, or to the organisation as a whole, may not justify the effort of joining.

5. THE PEOPLE TABLE

For each person that an application wants to hold data about, it needs either to identify that person in the People Table, establishing the link, or it must invoke the addition of a new record the People Table. To help make the identification with an existing person, additional data that might be helpful from other applications can be joined in and inspected.

The principle of the People Table is that it should contain just sufficient personal data about each person to identify him or her uniquely, but not any data that are rôle or application dependent. In practice, this aim cannot be perfectly realised. Thus the attributes might just be name, sex and possibly date of birth, plus some unique (primary) key. Each of these is not without difficulty.

The People Table assigns a unique key ("personal identifier" or *PID*) to each record. This has no semantics associated with it and is independent of any key used by any other application. Existing, real-world identifiers fail: unpaid visitors have no payroll number; foreign scientists have no national insurance (social security) number; an expired or lost security pass might be replaced by a new one with a different serial number; and so on.

For a person's name, one would probably record title, forename (given name), initials and surname (family name). Surname is the most reliable attribute, but even here ambiguities and mistakes are possible, particularly in punctuation. For example,

> *Lawrence-Wilson* or *Lawrence Wilson*
> *MacBeth* or *Macbeth* or *McBeth* or *Mac Beth*
> *St John* or *St. John*

while foreign names, particularly from very different cultures, can cause confusion.

A significant difficulty with surnames is that they do occasionally change, typically on marriage. The effect is that a person can seem to "disappear" from a database. Saving a link to the old name can be a worthwhile complication.

Forenames are less reliable, though useful in distinguishing persons with the same surname. Someone known as "Ted" would have initial "E" (for "Edward"). That might not even be the first initial. Besides, the forename or all the initials might not be recorded originally.

Even sex can be problematic. A new postal borrower of a library book might have a title and name from which the sex is indeterminate and so needs to be inserted as unknown.

Date of birth is a fairly precise discriminator. Unfortunately it often needs to be treated as confidential. Its use then by other applications may be restricted. Privacy controls complicate database administration.

A person's title (eg Dr, Ms, Prof, Eur Ing) is fairly unreliable in helping to identify someone. It can change. However since it is a piece of information desired by so many applications, there is a strong case for including it in the People Table and trying to maintain it.

6. THE FEDERAL PEOPLE DATABASE

The federal people database consists of the People Table together with tables from one or more applications that contribute data about people that are to be shared. Each of these applications may be one or more tables linked to the People Table by the *PID*. That is, the application will have the *PID* as a foreign key. If it is only concerned with federal people, then it will not need its own separate primary key.

The federal people database may be distributed, possibly reflecting the location of the component applications. However it is characterised by a single schema and database management system.

An application that does not contribute data about people but just makes use of shared data can be in the federal people database system but need not be. In the latter case, it either accesses it with a distributed query from its local system or receives relevant data by replication.

The separate applications in the federal people database are managed and updated independently, the only constraint being maintenance of the link with the People Table. These sub-databases may be the "real" production database or a replica of it. The protocols for how these applications interact with the federal people database are dealt with in the following sections.

7. DATA MANAGEMENT PRINCIPLES

In a federated data system, there are three ground rules to establish:

1. Each data item is updated in only one place. This is the familiar, fundamental database principle which needs to extend across a federation.

2. Each data item has a defined owner who is responsible for the correct and timely updating of it. The owner can delegate the duty of maintaining it to someone else or to a group of staff, but retains responsibility. Usual locking protects against simultaneous updates.

3. The separate, autonomous applications agree to share or exchange data. Participation is voluntary; though within an organisation, higher management might insist.

8. TIMELINESS

Often overlooked in consideration of federal systems is the issue of timeliness of database updating. This has two facets:

1. If data replication is involved, then this must be sufficiently often. The frequency is dictated by the need to ensure that the delay introduced is fail safe.

2. The updating application must provide an adequate service level for the dependent applications. For example, if Personnel staff only insert new employees monthly (in time for salary payment, say), then that's no use for the Computer Service which needs the new person's details in order to register a new user within a day of starting work. Thus a satisfactory agreement on the efficiency and reliability of updating needs to be reached (or imposed). Failing this, the alternative is for the application expecting to receive the data to create a local, temporary record for the person and then to watch out for the later arrival of the federal one. Such a complication is covered in (c) in the next section.

There are two other conditions for sharing data. One is that an application using data provided by another must have confidence in its accuracy, at least to a degree sufficient for its own purposes. The other condition is that there should be no misunderstanding or conflict in the semantics of shared data. One application's definition of, say, a person's employment "status" might not correspond exactly to the needs of another. (They could, for example, disagree over whether maternity leave counts as working.) However, semantic heterogeneity is a well recognised (if not solved) multi-database problem.

9. RECEIVING FEDERAL DATA

An application wishes to make use of data about people from the federal database. The relevant data are defined as a view (in the sense of a query) on the People Table joined on the *PID* with data from zero or more other applications sharing their data in the federal database. The view selects the required attributes and a predicate determines the relevant people. The application may also exist in the same database system or it may receive the federal data as an extract file.

Consider the different situations for the use of the data thus extracted:

a) The data received are entirely sufficient for the purpose of the application.

   As a simple example, it may be just names and locations needed to produce address labels in a selected department.

b) The federal data need to be merged with the application's separate data.

For example, a mailing application records in its own table the number of copies of a regular publication to send to each person. It updates the name and address of each person by merging in data from the federal database.

The procedure for such a data merge then has three cases to handle:

replace: Existing records in the application's table (attributes: *PID*, *name*, *location*, *quantity*) are overwritten with possibly new values for the *name* and *location* of each person record.

insert: A new person not already in the local application's table must be inserted. Local attributes (eg *quantity* in the example) are set to a default value or null. A report might be triggered notifying the local administrator of newcomers.

delete: If a person is no longer received from the federal database (eg has moved to another department), then a rule is applied to decide if the record is deleted in the local table. (It probably would be in our example.)

Note that the federal *PID* is stored in the local application as the primary key. In the comparison of the latest extract of federal data with the local table, the *PID* determines the choice of replace, insert or delete action.

c) The local application maintains its own set of people, supplementing those in the federal people database.

For example, a Computer Service might register "internal" users whose details are obtained from an organisation's federal people database, while the details of its "external" users (who have no other relevance to the organisation) are maintained by computer support staff.

In this situation, the procedure for merging in the federal data is similar to that described in (b) above. The local application will need its own primary key for its combined table of internal and external users. The federal *PID* will be retained as a foreign key to use in the merge procedure, external users being distinguished by a null value.

A complication arises if the two sets of people, federal and local, can overlap or the boundary could shift. An individual might move from one set to the other. Thus in the example, there would then be a need to check that when about to insert a new internal person that he or she is not already registered as an external computer user, and vice versa. In such checking, we have the "people identification problem".

10. CONTRIBUTING FEDERAL DATA

As opposed to using or receiving federal data, an application may be the source, sharing some or all of its data. Either the application is contained within the federal database system and updates its shared tables there directly, or it sends a copy of its data to the federal database. In both cases, one has the data merge situation in the sense of considering replace, insert or delete of people records.

Again the key to the merge procedure is comparison of unique key values. Assume that the application contributing to the federal data has its own primary key for its people. Then the federal people table stores that attribute as a foreign key to determine the processing actions on incoming data.

For inserting new federal people records, the "people identification problem" again arises. However, people dropped by the application cannot just be deleted from the federal database as other applications may be linking to them. (Even if one can establish that no application in the federation currently wants an individual, the record, with its *PID*, may sensibly be retained in case that person "reappears".)

11. EXCHANGING DATA WITH THE FEDERAL PEOPLE DATABASE

An application might both receive data from, and contribute (different) data to, the federal database. An example might be a personnel database which supplies the organisation's federal people database with job grade details in exchange for office locations of staff that are maintained by departmental administrators.

For two-way data exchange, the replication procedures are a combination of those outlined in the two previous sections. In the example of the personnel database, it would only get back from the federal database the records for those individuals for whom it has previously contributed data. Such records are recognised as possessing a non-null value for the foreign key that is the personnel's primary key.

12. MULTIPLE FEDERAL DATABASES

Two organisations, each with a federal people database, might merge. They may replicate to each other their staff records. (Thus allowing each, for example, to have a unified telephone directory.) However there may be no point in replicating other people data, that may be quite extensive, which are of no interest to the other organisation.

This situation is similar to the previous section. Each federal people database would hold the other's primary key as a foreign key. Mutual replication sufficiently often works as long as update responsibility is cleanly partitioned. The partitioning may be

vertical - different sites or applications update different attributes; and/or

horizontal - each site, say, updates its own people.

Otherwise, without global transactions, update conflicts can arise. These are resolved by rules or manual intervention.

The major difficulty is the "people identification problem". In this case, someone might transfer from one organisation to the other, or might have an office, say, in both.

13. THE PEOPLE IDENTIFICATION PROBLEM

The crucial step in federating data about people is recognising when two people records that arise from separate applications really concern the same individual. Whenever a new record is inserted, either into the federal people database or into that of an application, then one has to decide if that individual already exists in the database. As discussed earlier, there is no universal, unique attribute of a person that is always available. Thus we need appropriate strategies to handle this uncertainty.

In on-line data entry, the operator might type in a person's name, and the system could respond with a pick list of candidates already existing in the database. They could be chosen as having the same surname and, say, first initial. Names with similar spelling or pronunciation might also be included. The operator then decides if the new entry should be identified with an existing people record or is genuinely new. To help in the judgement, the pick list could display alongside the name some additional attributes available from the application or from the federal database.

If the data entry is a batch operation, or the federal names table is not available to the application on-line, then the system's decision to insert or update (or lack of decision) needs later, manual review. For this, "exception" reports can generate listings of possible missed matches.

For example, a daily or weekly report might be run to list all the unmatched individuals with the same surname and at least one common initial. The local application could then decide if a locally held individual can be identified with someone in the federal names table. Conversely, it could decide to flag that individual as explicitly ruled out of consideration. In the latter case, the flag would prevent the candidate being thrown up by succeeding exception reports.

Less frequently, say monthly or quarterly, a more extensive check might be conducted. All records flagged as excluded might be reconsidered. (An individual's circumstances can change.) The matching condition might be relaxed, thus yielding a much longer list of (less likely) candidate matches to assess.

Inevitably, the occasional duplicate will slip through and an individual will occur in a names table more than once. Sometimes there is too little information available about someone at the time to make a sound judgement. Sometimes an individual reappears with a new name. Another circumstance where duplication is conceivable is when multiple applications all want to insert a new person at roughly the same time and they check with a slightly out-of-date federal names table. An example is that of a new employee who soon after arrival, in separately managed applications, is assigned a payroll number, issued with a security pass and registered on the computer.

Whatever the cause of duplication, it should be relatively rare, detected reasonably promptly and corrected thoroughly. The success of a federal data system depends on attention to such data integrity.

14. CONCLUSION

Managing data in the real world is messy. Three main causes are:

a) Individuals (or individual groups or organisations) want a degree of autonomy over "their" data, so federal databases are more common than distributed ones;

b) The evolution of data systems due to changing environment and changing requirements mitigates against the viability of an all-encompassing global database;

c) Identifying people can never be completely reliable or automatic. Ultimately one needs human judgement.

Thus while formal techniques and methods for data system design exist and are very valuable, they are not sufficient. Tackling problems such as the above is the real job of an information systems engineer.

15. ACKNOWLEDGEMENTS

My colleagues who are involved day-to-day in the implementation and operation of various people-oriented database applications in CLRC have contributed to the development of the work reported here. Particular recognition is due to John Bateson, Judy Lay, Kevin O'Neill and Julia Thomas.