The ERROS Connectionist Database – Data in Perpetuity

The ERROS Connectionist Database was designed, from the very beginning, to ensure that data, once entered, can never be lost without trace and that all data can be totally relied upon to be there - in perpetuity.

Changes to records in many traditional applications involve updating the record in place, thus overwriting the existing data which is then lost. There are occasions when this is appropriate and updating in place can be allowed in ERROS applications. All metadata and user data are stored in the Connectionist Database and all changes are journaled (logged) in ERROS, and both the “before” and the “after” versions of the data are recorded as part of the ERROS Audit Trail so no data is ever lost,. A security officer can view, and, if necessary, roll back the changes. The security officer can switch off journaling, but only when ERROS is not working. If this happens, the change of state is recorded, and ERROS cannot be operated until journaling is restarted.

A major difference between the ERROS Connectionist Database and most traditional applications is that, whereas, in traditional file systems, all data about a contact is generally stored in a single record, in ERROS each separate attribute value is stored in a single record.

In most databases, deleting a record means that it is removed from the database and disappears without trace. In ERROS, deletion of a record means that it is not removed but simply marked as deleted. It is no longer seen by regular users but can be retrieved by a user with adequate authority who may be able to re-activate it. Although records can be totally removed, operators are not normally given authority to remove any records. This means that relational integrity is always maintained, even with records that have been deleted. If any records are removed, they will still be in the audit trail.

Rather than updating records in place, it is more normal in ERROS to mark a record as deleted and replace it with a new record. Thus, if, say, a person's telephone number is changed, the old telephone number record is marked for deletion and a new telephone number record is created. Thus the history is maintained.

Each transaction is automatically given a unique transaction number and a timestamp. The operator making the change is identified, as are the program and the ERROS application used to make the change. There will not be any gaps in the numbers. The audit trail record with the transaction number cannot be rolled back nor can the journal records.

It would be theoretically possible to use a standard utility, such as SQL, to modify the ERROS database, and, if journaling had been switched off, the change would not be recorded. However, the design of the ERROS database prohibits this. The ERROS database has a 17 part, 28 byte binary key field. SQL would not be able to find any particular record. It might be able to cause random damage, but this can be stopped by removing authority to use SQL. If a user had access to a compiler, they would not be able to do anything other than cause random damage as the layout of ERROS data is defined in the ERROS database which they could only access if they were an authorised ERROS user. If the ERROS database is encrypted, then no such random damage can be caused as the database can only be altered by ERROS itself.

Whether a record has been updated in place, or marked as deleted, or actually removed from the database, the ERROS audit trail and journal records ensure that all changes are recorded and that no data is lost, either accidentally or malevolently. The ERROS Connectionist Database has a verifiable and auditable record of every transaction.

As all metadata, which defines the user data, is also stored in the Connectionist Database, in other words in the same database as the user data that it defines, the same features apply to all metadata. Thus the audit trail records all application changes. Thus, although records can be changed, the extraordinary functionality of the ERROS Connectionist Database means that all metadata and user data will always be available in perpetuity.

Because all ERROS metadata and user data changes are journaled, an ERROS high availability option allows multiple local or remote duplicate servers anywhere in the world to maintain up-to-date copies of all metadata and user data. Providing there are no communication problems, these should be no more than a fraction of a second behind the prime system, and one of them could be switched to become the prime system if that fails. Users who simply require read only access could use one of the duplicate servers, so reducing the load on the main system.

Rob Dixon