AN IT INFRASTRUCTURE FOR LONG TERM RESEARCH & DEVELOPMENT AT THE CRC FOR CONSTRUCTION INNOVATION

Robin Drogemuller
Division of Manufacturing & Infrastructure Technology, CSIRO
Robin.Drogemuller@csiro.au

Keith Hampson
Cooperative Research Centre for Construction Innovation
k.hampson@construction-innovation.info

Kwok-Keung Yum
Division of Manufacturing & Infrastructure Technology, CSIRO
Kwok-Keung.Yum@csiro.au

SUMMARY

The Australian Cooperative Research Centre for Construction Innovation (CRC CI) started operations in July 2001. One of its aims is to address the relatively low level of R&D activity within the Architecture, Engineering, Construction and Facilities Management industry in Australia. This paper briefly describes the general goals of a Cooperative Research Centre within the Australian national context. Information & Communication Technologies (ICT) are playing a significant role in the deliverables of the CRC CI. This has necessitated the definition of an ICT architecture at both the software application level and the project server level to provide a framework to maximise the effectiveness of the CRC CI's R & D expenditure. The current architecture is described and some key issues that will need to be resolved are identified.

INTRODUCTION

The most significant commitment to Architecture, Engineering, Construction and Facilities Management (AEC-FM) research in Australia has been made by the Australian Government by establishing a new Cooperative Research Centre for Construction Innovation (CRC CI) with funding and in-kind commitments to the value of A$64 million over seven years. The CRC CI is a collaborative venture between 19 industry, government and research partners across Australia, with key links internationally. The CRC CI is an example of the Australian Government's program to build stronger links between industry, universities and research agencies to achieve world class research and innovation.

Information and Communication Technologies (ICT) are expected to be major deliverables from CRC-CI projects and to play a major part in the dissemination of results to industry. In order to maximise the return on investment the CRC CI must ensure that the software development projects share as much common expertise and code as possible. It is also necessary to link with current commercial applications to maximise the benefit to the market from the CRC CI's products and hence the market penetration.

A fundamental assumption is that interoperability and the use of the IFCs (IAI, 2002) will provide the link with current commercial software. It is also recognised that a migration path from current practice to the "desired future state" is necessary for acceptance of new software products by the market.

The following sections review the current work within the CRC CI and the strategic issues which have arisen from this work.

THE COOPERATIVE RESEARCH CENTRE FOR CONSTRUCTION INNOVATION

The CRC CI has an ambitious vision "Our vision is to lead the Australian property and construction industry in collaboration and innovation" (CRC for Construction Innovation, 2002). It aims to foster change throughout the AEC-FM industry. The revised program/platform structure (Figure 1) recognises the fundamental role of ICT within the CRC's activities.

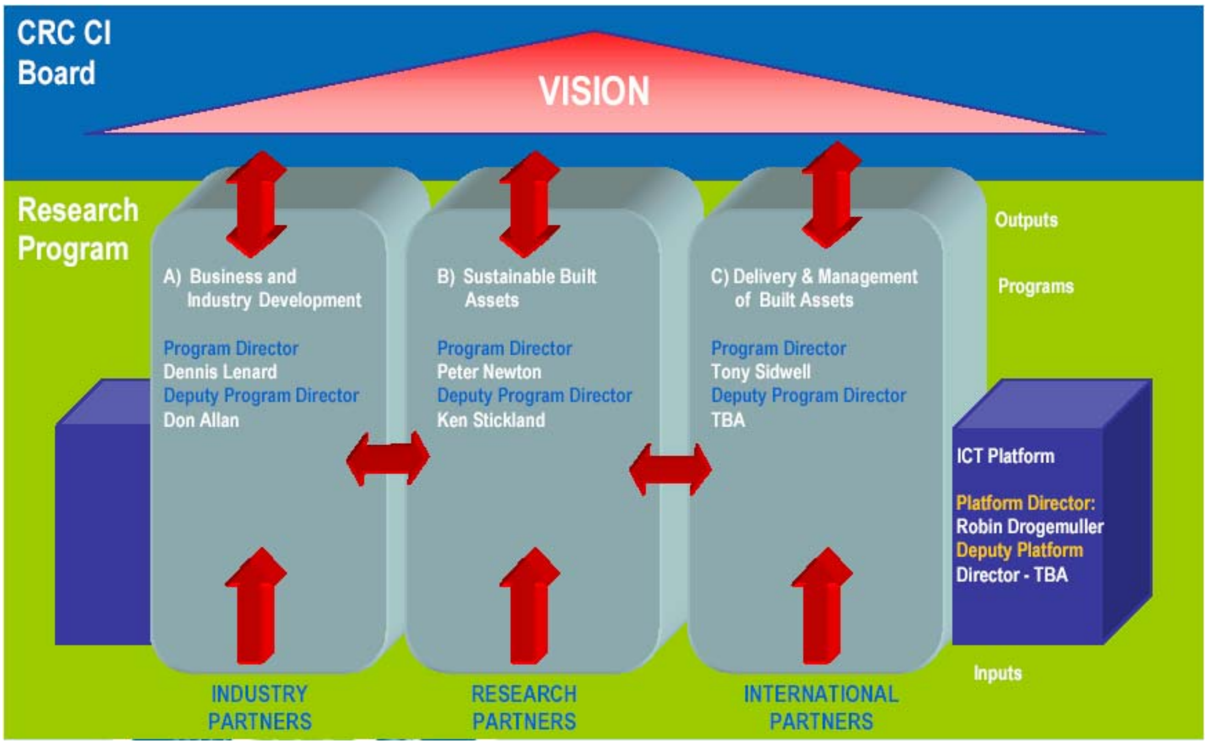


Figure 1 CRC CI Programs & Platform

The current range of software projects (CRC for Construction Innovation, 2002) is wide, although the width of the domain each covers is relatively narrow. The CRC CI sees no need to replicate existing commercially available software unless there are significant advantages to be gained through increased functionality that are not being pursued by the existing software houses. The aim is to improve the efficiency of the AEC-FM industry through increasing the extent, usefulness and power of the range of available software.

The major software-based projects currently underway within the CRC CI cover the following areas:

- Semi-automation of quantity takeoff and estimating for structural building elements
- Automation of compliance checking for access by disabled people
- Design Activities within Virtual Environments
- Environmental sustainability analysis of commercial buildings

Projects that have been approved and will be starting in the near future include:

- Immersive design environments and collaboration over high bandwidth networks
- Multiple views of information in virtual worlds
- Contractors planning and scheduling decision support
- Parametric design for multi-storey commercial/residential developments

The general principle underlying the selection of appropriate projects is that the CRC CI should not directly compete with existing commercial products or major research initiatives that are moving towards commercialization. The CRC CI will provide better value to its stakeholders and the AEC-FM industry by adding value in new areas rather than duplicating other work.

SOFTWARE DEVELOPMENT CONSTRAINTS & REQUIREMENTS

The CRC CI is developing some leading edge software applications to meet the needs of the industry partners. These applications are making use of recent (in commercial timeframes) research and development results such as multi-user databases, product modelling, software agents and virtual environments. The software infrastructure adopted within the CRC CI needs to meet the following constraints on software-based projects:

1. Need to be flexible enough to support research projects
   Research projects, by their nature, have higher risks than normal software development projects. Initial assumptions are often changed, new methods of achieving the aims are identified and new aims emerge. It is important to provide flexibility in programming and in delivery of the resulting software. This has lead to the separation of data, domain knowledge and user interfaces in the current projects.

2. Need to be able to commercialise software in the long term
   One of the aims of the CRC Program is to lead to the commercialization of research. Consequently, software based projects need to ensure that initial decisions do not interfere with commercialization needs. This means that all issues need to be addressed on the route to commercialization. The research team need to have ideas on how the entire problem area will be tackled rather than just cover the proof-of-concept. This also means that commercial quality components need to be used early in the project if databases or other third party software will be used. It is also important to provide interfaces to commercially available software where this is appropriate. Currently, the major mechanism for achieving this is to import IFC (IAI, 2002) files. In the near future interfaces will also be developed to the XML interfaces defined by the IAI (IAI, 2002) and BLIS (BLIS, 2002).

3. Several separate teams working in parallel on different aspects
   Each project within the CRC CI must have two industry partners and two research partners. Since the participants are often located in different cities, modularity of the software is important in minimising coordination issues. The separation of data, domain knowledge and user interface also assists here.

4. Related projects may not be sequential
   The development of commercial software products requires significant effort. Since most CRC CI projects are also high risk, there is no guarantee that all of the necessary components will be completed when planned. This may be due to technical difficulties or to constraints on cash flow or in-kind contribution.

5. Similar or identical components may be used across multiple projects
   Many of the software projects share sub-problems. Using a modular approach makes the sharing of code and optimization of effort simpler.

6. Need to use current software development theory
   Most existing software for the AEC-FM industry has a long history, in software terms. This means that they need to maintain some relation with their legacy code. The new software being developed by the CRC CI can use newer techniques such as a three tiered architecture and the various design patterns (e.g.Gamma *et al*, 1995).

There are two architectures being used within CRC CI software projects. These are based on the assumption that the current “single user” paradigm will gradually change to a “shared data” or “project database” paradigm. The current single user paradigm allows the creators and editors of information to have exclusive write control of a “block” of data. The project database paradigm is a shared model approach where users can edit their respective “views” of information in parallel. The single user architecture will also need to support the project database paradigm if the CRC CI is to get maximum return on its investment.

SINGLE USER/CLIENT-SIDE ARCHITECTURE

Most current software for the AEC-FM industry has an architecture similar to that shown in figure 2. The primary user interface (UI) indicates the “type” of software – CAD, estimating, scheduling, etc. The domain specific knowledge required to support this primary UI underlies the primary UI and stores the relevant data in a proprietary internal database. Some software supports supplementary UIs that allow additional views of the data within the internal database. These supplementary UIs are normally read-only. Examples of these supplementary UIs are the list views available in ArchiCAD (Graphisoft, 2002) and the basic schematic drawing view available in Timberline's CAD Integrator (Timberline, 2002).

Most AEC-FM software supports some method(s) for importing and exporting data. The most widely used is the DXF format but this has significant limitations and may be phased out by Autodesk in the near future. The Industry Foundation Classes (IFCs) (IAI, 2002) are the richest format for exchanging data but there are technical and commercial issues in the uptake of the IFCs by software vendors. The major constraint on the exchange of file based data within the AEC-FM industry is the lack of a standard. The IFCs are the most likely candidate for an industry-wide standard.

Some commercial software (ie AutoCAD (Autodesk, 2002a) and Revit (Autodesk, 2002b) supports the direct export of data into external databases. This is useful when dealing with non-geometrical data, but getting detailed geometrical data exported to a database is difficult. If project databases are ever going to achieve widespread use then the export of all relevant data is essential.

Some commercial software also supports external APIs that allow programmers to write extensions to the software or to add additional file formats for import or export. These external APIs can also be used to provide direct export of data to external databases.

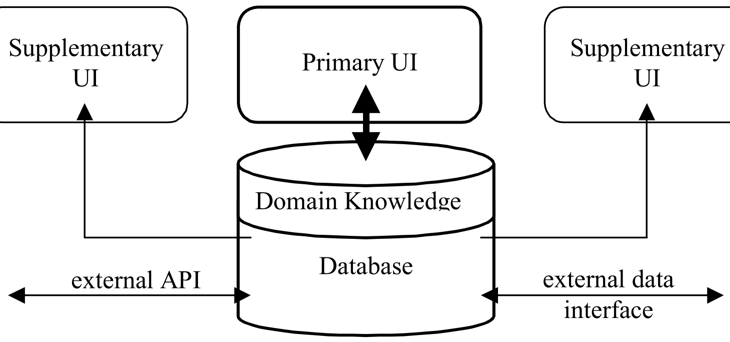


Figure 2 Typical Architecture of Current Software

The CRC CI is using a slightly different approach for the single user architecture (figure 3). The core difference is the use of a proprietary EXPRESS compatible object-oriented database, the EDM Data Manager (EPD, 2002). This provides all of the persistent data storage for the application. All access to the database is through the standard API. Above the database are separate programs which contain the relevant domain knowledge and provide the UI. Where a program can work automatically, that is an intelligent agent, then this can work continuously without need for a user interface. The database automatically supplies both the external API and an interface to the IFCs.

The EDM Data Manager provides a number of functions that simplify the R & D efforts. Firstly, it allows the definition of the database schema in EXPRESS with the EDMVisual EXPRESS software. Since EXPRESS is widely used within the product data modeling (PDM) world, this simplifies the definition of compatible data structures. Secondly, the implementation of EXPRESS-X, a mapping language, allows the definition of mappings between the various versions of the IFCs (and possibly other EXPRESS schema) and the schema used internally. This frees the CRC CI projects from the cycle of IFC releases. Another feature within EDM Developer which is available and looks promising is "business objects". These potentially provide domain specific views of the underlying EXPRESS model. However, this feature has not yet be exploited within CRC CI projects.

A possible question is "Why develop a separate schema, why not use an IFC schema within the database?" There are several reasons for defining a separate schema:

- Which version of the IFCs should be the internal representation, IFC 2.0, 2x, 2x amendment 1? Mappings can be defined from all of these to the CRC_schema.
- Optimisations can be built into the CRC_schema for the objects of interest to our projects. It is not reasonable to expect a schema defined for information exchange to be appropriate for internal data structures.
- New entities can be added to the CRC_schema and then mapped across to the various IFC schemas as IfcProxy objects without compromising the internal performance of our software.

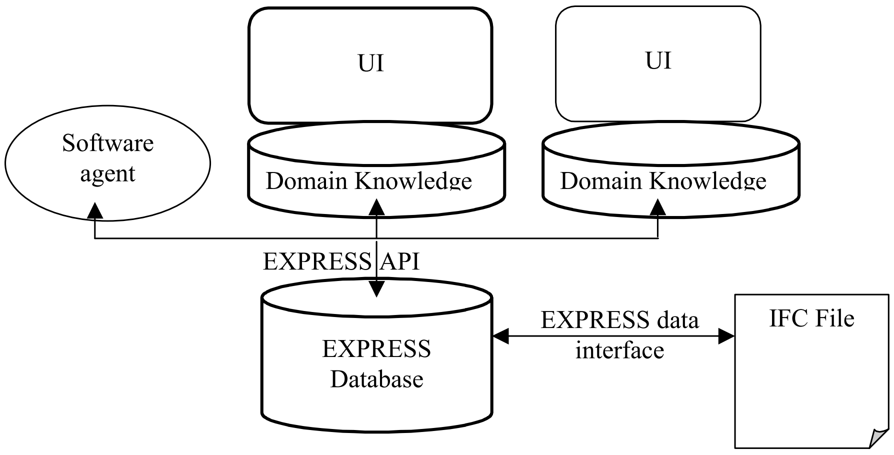


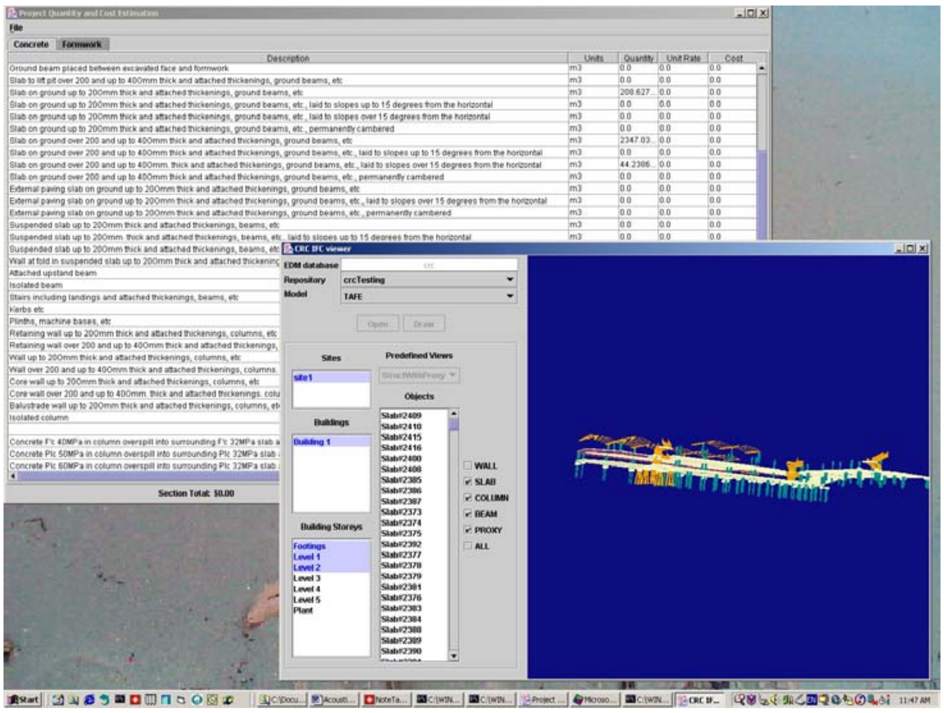
Figure 3 CRC-CI Stand-alone Architecture

One of the projects within the CRC CI is developing quantity take-off and estimating software (figure 4). The unique aspects of this project are the close integration of the estimating UI with a three dimensional viewer and the ability of the user to define their own items in the Bill and have these automatically merged with the decision tree used to classify the building elements against the items. The close integration of the Bill and the viewer also allows the user to ask the following important questions and to receive an understandable response:

- Which element(s) in the Viewer correspond to this item in the Bill?
- Which item in the Bill corresponds to this element in the Viewer?
- Which items in the Viewer are included in the Bill?
- Which items in the Viewer are not included in the Bill?

The software architecture provides the following advantages:

- The development of all of the software components is relatively independent of the others through cleanly defined software interfaces;
- The functionality of the components can be adapted or extended without impacting the other components



Description	Units	Quantity	Unit Rate	Cost
Ground beam placed between excavated face and formwork	m3	0.0	0.0	0.0
Slab to lift pit over 200 and up to 400mm thick and attached thickenings, ground beams, etc	m3	0.0	0.0	0.0
Slab on ground up to 200mm thick and attached thickenings, ground beams, etc	m3	208.627...	0.0	0.0
Slab on ground up to 200mm thick and attached thickenings, ground beams, etc., laid to slopes up to 15 degrees from the horizontal	m3	0.0	0.0	0.0
Slab on ground up to 200mm thick and attached thickenings, ground beams, etc., laid to slopes over 15 degrees from the horizontal	m3	0.0	0.0	0.0
Slab on ground up to 200mm thick and attached thickenings, ground beams, etc., permanently cambered	m3	0.0	0.0	0.0
Slab on ground over 200 and up to 400mm thick and attached thickenings, ground beams, etc	m3	2347.03...	0.0	0.0
Slab on ground over 200 and up to 400mm thick and attached thickenings, ground beams, etc., laid to slopes up to 15 degrees from the horizontal	m3	0.0	0.0	0.0
Slab on ground over 200 and up to 400mm. thick and attached thickenings, ground beams, etc., laid to slopes over 15 degrees from the horizontal	m3	44.2386...	0.0	0.0
Slab on ground over 200 and up to 400mm thick and attached thickenings, ground beams, etc., permanently cambered	m3	0.0	0.0	0.0
External paving slab on ground up to 200mm thick and attached thickenings, ground beams, etc	m3	0.0	0.0	0.0
External paving slab on ground up to 200mm thick and attached thickenings, ground beams, etc., laid to slopes over 15 degrees from the horizontal	m3	0.0	0.0	0.0
External paving slab on ground up to 200mm thick and attached thickenings, ground beams, etc., permanently cambered	m3	0.0	0.0	0.0
Suspended slab up to 200mm thick and attached thickenings, beams, etc	m3	0.0	0.0	0.0
Suspended slab up to 200mm. thick and attached thickenings, beams, etc., laid to slopes up to 15 degrees from the horizontal	m3	0.0	0.0	0.0
Suspended slab up to 200mm thick and attached thickenings, beams, etc				
Wall at fold in suspended slab up to 200mm thick and attached thickening				
Attached upstand beam				
Isolated beam				
Stairs including landings and attached thickenings, beams, etc				
Kerbs etc				
Plinths, machine bases, etc				
Retaining wall up to 200mm thick and attached thickenings, columns, etc				
Retaining wall over 200 and up to 400mm thick and attached thickenings,				
Wall up to 200mm thick and attached thickenings, columns, etc				
Wall over 200 and up to 400mm thick and attached thickenings, columns.				
Core wall up to 200mm thick and attached thickenings, columns, etc				
Core wall over 200 and up to 400mm. thick and attached thickenings. colu				
Balustrade wall up to 200mm thick and attached thickenings, columns, et				
Isolated column				
Concrete F'c 40MPa in column overspill into surrounding F'c 32MPa slab a				
Concrete Plc 50MPa in column overspill into surrounding Plc 32MPa slab				
Concrete Plc 60MPa in column overspill into surrounding Plc 32MPa slab				

Section Total: $0.00

EDM database: crc
Repository: crcTesting
Model: TAFE

Objects: Slab#2409, Slab#2410, Slab#2415, Slab#2416, Slab#2400, Slab#2408, Slab#2385, Slab#2386, Slab#2387, Slab#2373, Slab#2374, Slab#2375, Slab#2392, Slab#2377, Slab#2378, Slab#2379, Slab#2381, Slab#2376, Slab#2383, Slab#2384, Slab#2388, Slab#2389, Slab#2390

WALL [✓ SLAB] [✓ COLUMN] [✓ BEAM] [✓ PROXY] ALL

Figure 4 Estimating & Viewing Modules

SERVER-SIDE ARCHITECTURE

One research issue that the CRC CI is addressing is the role of centralised project databases. Several researchers [Eastman, 1999; Amor and Faraj, 2001] have considered the possibility of centralised databases holding all of the "shared" information within a project. The general consensus has been that it will be a very challenging task. However, the potential benefits of such central data stores mean that continuing these efforts is worthwhile. One advantage that the CRC CI has over similar efforts is the parallel contribution from research projects examining the various roles of the parties in construction procurement contracts. This may inform the ICT work on the appropriate levels of sharing of information for various forms of construction procurement contracts.

Many of the people working on the research projects do not have a formal ICT background so it is important that they do not have to develop a high level of skill in computer programming or product modelling in order to complete their assigned tasks. This especially applies to post-graduate research students who are working on the projects as they have limits on the time to complete their studies. This means that it must be possible to present simplified models and interfaces to these simplified models. In addition, the programming skills to access information can not be too rigorous.

Provision of a single platform that can support the R&D software life cycle, from research project through prototype to commercial product is a significant constraint. In order to maximise the outcomes from the CRC CI disconnections during the software development cycle must be minimised.

At the International Alliance for Interoperability meetings in Tokyo in October 2002, three of the groups working on IFC server projects agreed to collaborate on defining a common interface to their server projects. The groups were EPM, EuroStep and the IFC Model Server project to allow interoperability at the server interface indicates a coming of age for interoperability for the AEC-FM industry. The implication is that the AEC-FM industry in general and IFC-based development projects in particular must start considering how they will interface with project servers. A project server is defined as a single server or interface that provides access to all of the relevant project information whether this is generated in a CAD system, spreadsheet-based software, etc.

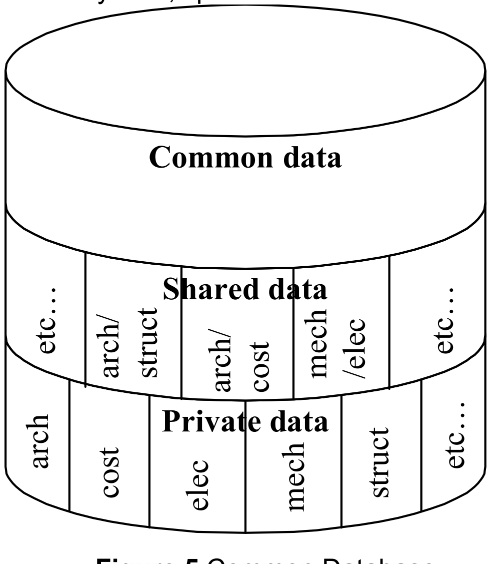


Figure 5 Common Database

For the purposes of this discussion we consider three types of information that can be stored in a project database:

- Common data that must be accessible to multiple members of the project team.
- Shared data which is required by members of the team for coordination purposes between just these members
- Private information that represents the core intellectual property of each project team member

There are three basic structures that could be used for a project database:

- common database (figure 5)
- partially federated database (figure 6)

- federated database (figure 7)

The main issue that needs to be considered when assessing the advantages and disadvantages of each is the privacy of the information and the level of trust placed in the database administrator. A subsidiary issue is the capability of companies within the AEC-FM industry to manage shared data and databases.

The technical issues of maintaining privacy, access control, etc are resolvable using current database technology.

For the purposes of this discussion we will assume that common information should be accessible to everyone in the project team, shared information should be accessible to the parties who need it for coordination purposes but it does not matter if other team members see it and private information needs to be shown at a particular level of granularity (defined in the project agreement) and for a defined scope, but no further. Common information would include the information given in a standard set of contract documents. Shared information would include the information exchanged between project team members which is used to coordinate the various contributions. Private information may include the library objects used by the architect, unit rates used by the quantity surveyor and estimator and the productivity factors used by the project planner.

If we assume the unrestricted exchange of electronic information within the project and possible outside it (approval authorities, etc), then the architect does not have much option but to include those parts of the library objects relevant to the project. However, the architect would not want to provide their entire library. The quantity surveyor will also need to expose some cost information, but once again this will only be for the particular project and to the required level of detail.

The shared database model is acceptable for the common and shared information but trusting a database administrator with all of the intellectual property used throughout a project is potentially dangerous. The two federated models both provide security for intellectual property (IP) at the level we believe is required within our industry. The database administrator can control who accesses the information and also how frequently. Controlling the frequency for access is necessary to prevent other parties from reverse engineering IP by iterating through a range of alternatives.

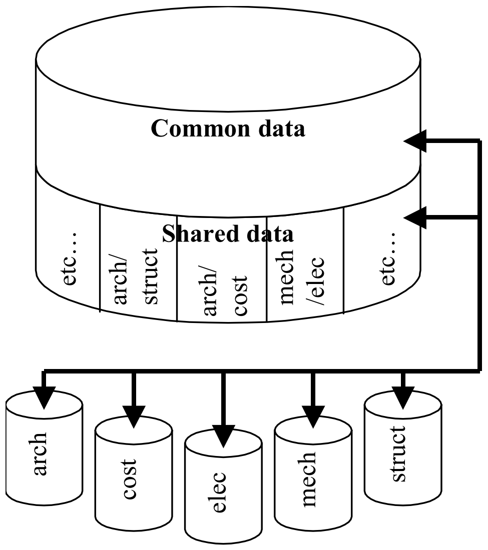


Figure 6 Partially Federated Database

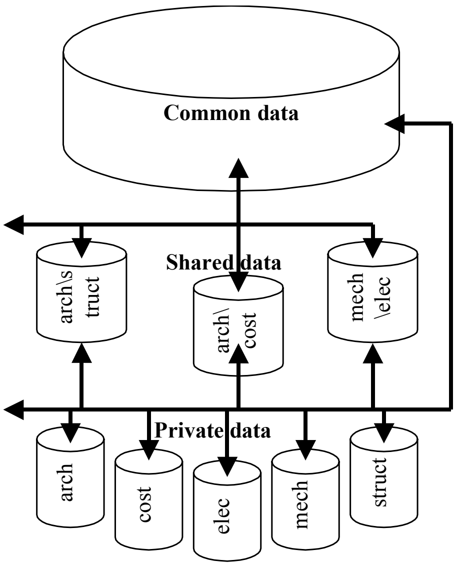


Figure 7 Federated Database

The projects that use the project server architecture only provide this capability as an optional service. The user configuring the project at initiation decides whether it will be hosted on a stand-alone or a server database. This is designed to allow commercial users to shift to a project server architecture at a time of their choosing rather than forcing them to change.

The work using the project server is only at the initial stage of development. There are many issues that need to be resolved for multi-user EXPRESS based databases. These include:

- Defining appropriate methods for access control;
- Handling different versions of the same project and instances within a project;
- Defining the “views” of the different members of the project team – which objects, which aspects of those objects and methods of interaction with the objects
- etc

Underlying all of these are the capabilities that will be required from a computer science perspective, many of which are covered by database research. However, some of these issues will need adaptation to suit the specific requirements of the AEC-FM industry.

CONCLUSION

The CRC-CI needs to ensure that its research projects can also be commercialised in order to meet the requirements of the CRC program. This imposes greater constraints on project scoping and selection than for a research project or a stand alone software development project. The CRC-CI needs to use a consistent software architecture to ensure that the maximum benefit is achieved from the effort. Since the CRC-CI has a life of 7 years it also must ensure that the deliverables from projects will be relevant in the future.

The CRC CI is using a “single user” architecture, which is consistent with current practice and technology. This is provided by the standard EDM developer database system. However, the consistency between the EDM Server and the standard EDM database means that the single user architecture can also support the emerging technology of “project servers”. This provides the “future proofing” that the CRC CI requires and will maximise the return on investment by both the CRC CI and the users of the CRC CI’s software products.

REFERENCES

Amor, R. and Faraj, I., 2001, Misconceptions about Integrated Project Databases, ITcon journal, Vol. 6, pp.57-68, http://www.itcon.org/2001/5/

Autodesk, 2002a, http://www.autodesk.com/

Autodesk, 2002b, http://www.revit.com/

BLIS, 2002, http://www.blis-project.org/

CRC for Construction Innovation, 2002, http://www.construction-innovation.info/

Eastman, C., 1999, Building Product Models: Computer Environments Supporting Design and Construction, CRC Press

EPM, 2002, http://www.epmtech.jotne.com/

Gamma E., Helm R. Johnson R. & Vlissides J., 1995, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley

Graphisoft, 2002, http://www.graphisoft.com/

IAI, 2002, http://www.iai-international.org/iai_international/

Timberline, 2002, http://www.timberline.com/