

# Escaping the Relational Database Paradigm: Case Management in the High Court of Australia

James Popple

Deputy Registrar, High Court of Australia  
Visiting Fellow, Faculty of Engineering and Information Technology, Australian National University

Tony de la Fosse

Deputy Marshal (Administration), High Court of Australia

The Australian Institute of Judicial Administration Inc.  
*Technology for Justice* Conference  
Melbourne, Australia

23 March 1998

## Introduction

The High Court is the highest court in the Australian judicial system. Its functions are to interpret and apply the law of Australia; to decide cases of special federal significance, including challenges to the constitutional validity of statutes; and to hear appeals from Federal, State and Territory courts.

The Principal Registry of the High Court is in Canberra, in the Court's own building within the Parliamentary Triangle. In addition, there are offices of the High Court Registry in the capital of each State and in the Northern Territory. High Court officers staff the Canberra, Sydney and Melbourne offices of the Registry; staff of state and federal courts provide registry counter services in the other six capitals.

The High Court Registry provides traditional registry services to the Court. These include managing the case flow of the Court, providing information to Justices concerning matters filed, and providing information to the profession and the public about the case flow and practice of the Court. Registry staff also respond to counter, telephone and written enquiries concerning the status and disposition of cases.

An appreciation of the judicial workload of the Court can be gained from the table in Figure 1, extracted from information in the 1996–97 Annual Report.<sup>1</sup> In the 1996/97 year, a total of 1,175 matters were filed in the Court.

In early 1996 the Court decided to replace its case management system. The old system was developed in 1987 and operated on a Prime mid-range 5340 processor. It was written in the Prime Information (Pick) language, running under the Primos operating system. The system was script-based and provided Registry staff with basic information concerning the status of cases before the Court. By early 1996 the system had reached the end of its useful life, and the cost of supporting the proprietary Prime processor was excessive.

---

<i>Category</i>	<i>Matters</i>
Special leave applications (civil)	268
Special leave applications (criminal)	101
Appeals (civil)	51
Appeals (criminal)	18
Writs of summons	700
Applications for removal	9
Orders nisi and applications referred to Full Court	21
Electoral matters	0
Other	7
<i>Total</i>	1,175

Figure 1: Categories of matters filed in the High Court of Australia in the 1996/97 year.

---

The Court required a new case management system which:

- facilitated workflow;
- had a graphical user interface;
- was intuitive to use;
- allowed “objects” to be embedded within the database;
- was robust;
- was secure;
- had an open architecture, and was capable of running on a variety of platforms;
- was flexible enough to allow routine changes to its database structure be made by Registry staff;
- integrated seamlessly with existing desktop applications;
- provided extensive statistical information;
- had advanced reporting and ad-hoc query functionality; and
- (optionally) allowed data at the field level to be published to the Internet.

After preparing formal specifications and reviewing existing case management systems in a number of other courts, it was decided to construct the new system “in-house” rather than attempt to modify an existing system. At that time, it was decided that Lotus Notes might offer an appropriate environment in which to construct the new system. To confirm that a system developed using Notes could meet all of the Court’s specifications, a prototype system was constructed. The prototype was completed in mid-1996, and the Court decided to proceed with the project utilizing Notes.

Price Waterhouse Urwick was selected to construct the system. Application development proceeded through 1997 and the final data migration from the old to the new system was completed on Christmas Day. The new system “went live” for the first time on the 2 January 1998.

## Lotus Notes

The decision to develop the system using Notes involved a departure from a traditional relational database architecture. Unlike traditional relational database solutions, emerging *groupware*<sup>2</sup> products such as Notes allow unstructured information (text, sound, pictures, video, etc.) to be embedded in a database, and indexed and manipulated in new ways. Workflow functionality can also be provided, and selected information from the database can be dynamically published directly to the Internet.

Lotus Notes databases consist of any number of *documents*. These documents, in turn, contain *fields* which can contain, essentially, any kind of electronic information. A scripting language, LotusScript, allows Notes developers to construct sophisticated mechanisms for processing the information contained in these databases. Document templates, or *forms*, determine how the information in a document is displayed, allowing information to be hidden for clarity or for security purposes, if required. *Views* display hierarchical lists of documents to assist the user in traversing a database.

All of the data in the documents that make up a database (including any embedded objects, or *attachments*, which are part of a field of a document) can be indexed for full text searching purposes.

Notes has the largest installed base of any groupware product, and provides considerable functionality. As well as having the flexibility to process any kind of electronic information, Notes has the ability to access relational and transaction based data stores. It also provides a client/server message system and discussion databases. Tools exist which allow large amounts of data to be replicated between Notes and other database systems. These features allow Notes to act as the central point of access to distributed organizational data, regardless of its source.

*Domino* is an addition to Notes that brings the functionality of Notes that allows content to be dynamically published to the Internet or to an Intranet. Using Domino, a Notes client on a LAN, or a web browser on the Internet, can securely access the same Notes database.

## System structure

The new case management system comprises several interconnected Notes databases:

- the *cases database* holds information about cases in the High Court;
- the *representatives database* holds information about the representatives of parties to matters in the Court (firms of solicitors, or self-represented persons);
- the *practitioners database* is the register of legal practitioners who are entitled (by virtue of their ability to practice as practitioners in one or more states or territories) to practice in any federal court, including the High Court;
- the *letters database* holds correspondence generated by the system;

Lotus Notes - P3/1998 Boys and Ors v. Australian Securities Commission and Ors

File Edit View Create Actions Section Window Help

View Past Events View Pending Events

**Full Case Record** P3/1998

Long title Alan Harold BOYS and Ronald George HOWARD and Desmond Frank CRAWLEY and Anthony Hayes DOUGLAS-BROWN and Anthony Howard LEIBOWITZ v. AUSTRALIAN SECURITIES COMMISSION and ERNST & YOUNG (a firm) and PHILLIPS FOX (a firm) and Peter Raymond QUIGLEY and Kevin Lee CHRISTENSEN

Short title Boys and Ors v. Australian Securities Commission and Ors

Alternative title Case number P3/1998

Type Leave Status Current

**Details**

Jurisdiction  Civil  Criminal

Date commenced 05/02/98 Age 6 weeks

Representative status represented Time since last step

Officer responsible James Popple Originating registry Perth

Party status  inter partes  ex parte

File location Canberra

Associated cases [Goto](#)

Appeal case [Goto](#)

Courts below 8/01/98 Federal Court of Australia(French J, Heerey J, Merkel J)

**Subject**

**Parties**

**Applicant**

1.1 Alan Harold Boys [Hunt & Humphry - EAST PERTH, Marshall Timothy McKenna Ph : (08)9221 0200, CH.MM:1453]

1.2 Ronald George Howard [Hunt & Humphry - EAST PERTH, Marshall Timothy McKenna Ph : [Edit](#)]

**Respondent**

1. Australian Securities Commission [Australian Securities Commission - PERTH, Michael John

Figure 2: An example case document from the cases database, showing some of the fields available for storing information about cases of this particular type.

- the *reports database* generates tables suitable for inclusion in the High Court's annual report; and
- the *private database* holds, in a completely distinct database, information about cases which is accessible to only a few High Court officers (e.g. the real name of parties to a matter which involves the custody of children, who are known by their initials).

The interconnectedness of these databases is such that a user of the system need not be aware of the division of information within the system. Information is passed between the different databases, and users move between databases by following links between documents. So, for example, in a document in the cases database, the user is shown the name and contact details of a party's legal representatives. That information is obtained by the cases database from the representatives database. At the click of a button in the document in the cases database, the system opens a document in the representatives database allowing information on that firm of solicitors to be changed. Those changes are reflected in the document in the cases database when that document is next opened.

Lotus Notes - P3/1998 Applicant 1.1BOYS

File Edit View Create Actions Text Window Help

View Alias

**Applicant 1.1** P3/1998

Type  Natural Person  Non-Natural Person  Office Holder

Family name

Given names

Capacity

Represented  Unrepresented  No appearance

**Representation**

Firm Hunt & Humphry - EAST PERTH Ph : (08)9221 0200

Reference

Town agent

Solicitor Marshall Timothy McKenna Ph : (08)9221 0200

Counsel

**Notes**

Created by Sonia Seymour on 05/02/98 18:32:38. Last Edited by Tony DE LA FOSSE on 19/03/98 16:06:29.

Figure 3: An example party document from the cases database.

### *The cases database*

The cases database contains three different kinds of document: cases, parties and events. There is a case document for every matter, or case, which comes before the Court (see Figure 2). Each contains information about the case: what type of case it is (e.g. an appeal, an application for prerogative relief, a writ of summons) what it is called, what it is about, where and when it was commenced, which officer is responsible for it, the location of the (paper) file, etc. The fields displayed in a case document vary depending on the type of the case: only those fields that are relevant to cases of that type are displayed.

There is a party document for each party to each case (see Figure 3). This document stores information about who that party is, and who are its legal representatives. A summary of information about each party to the case is displayed in the case document and, by clicking a button, the user is taken to the party document for that case.

Similarly, from each party document there are links into the representatives database, connecting a party with information about its firm of solicitors (if represented) or with contact information (if self-represented). There are also links from party documents into the practitioners database, connecting a party with its solicitors and barristers.

Lotus Notes - Australian Government Solicitor - SYDNEY

File Edit View Create Actions Section Window Help

Exit Representative

**Australian Government Solicitor**

Type  Firm  LIP

Firm Australian Government Solicitor

**Address**

Postal address GPO Box 2727

City/suburb SYDNEY

State NSW Post Code 2001

Telephone (02) 9581 7777 Facsimile (02) 9581 7778

DX number DX 444 E-mail address

DX exchange SYDNEY URL

**Account**

**Notes**

New address as from 16/11/92.

Created by HCA2 on 18/12/97 03:46:20. Last edited by Sonia Seymour on 25/02/98 18:41:12.

Figure 4: An example representative document from the representatives database.

Information about events which have occurred, and which may occur, in a case is stored in event documents. These events are the essence of the system's structure, and are discussed below.

### *The representatives database*

A party's representative is either a firm of solicitors or her/himself (if self-represented). The representatives database is a collection of documents with contact information about these firms and persons (see Figure 4).

### *The practitioners database*

Like the representatives database, the practitioners database is an uncomplicated collection of documents. Each document contains contact and admission details for a legal practitioner, and includes links to firm details in the representatives database where appropriate (see Figure 5).

(Practising certificates and certificates of good conduct, issued to practitioners, are generated by this database and details of their issue are stored here for future reference.)

Lotus Notes - Mr Marshall Timothy McKenna

File Edit View Create Actions Section Window Help

Certificate

**Barrister and Solicitor** Marshall Timothy McKenna WA

Family name: McKenna  
 Given names: Marshall Timothy  
 Title: Mr Gender:  Male  Female

**Address**

Firm: Hunt & Humphry - EAST PERTH Ph : (08)9221 0200 [Add] [Goto] [Delete] [New]

Postal address: c/- Hunt & Humphry  
 Level 2, Hyatt Centre, 20 Terrace Road  
 City/suburb: EAST PERTH  
 State: WA Postcode: 6004  
 Telephone: (08) 9221 0200 Facsimile: (08) 9221 0211  
 DX number: DX exchange: E-mail address:

**Admission**

Capacity: Barrister and Solicitor  
 State: WA

**History**

Original admission: 19/04/93 Batch number:  
 Original capacity: Barrister and Solicitor

Figure 5: An example practitioner document from the practitioners database.

### *The letters database*

At present, there are 22 standard letters which are generated by the case management system (see Figure 6). These letters are constructed upon the occurrence of various events; a letter is produced for each party to a case, as appropriate, and stored in the letters database. Some of these letters are quite complicated, and have many variants depending upon the values of different fields in the event document which generated them, and upon fields in case, party, representative and practitioner documents.

Once generated, each letter can be individually edited if necessary. Each can be printed immediately, or stored for printing (with other letters) later. The user can opt to print each letter to a printer, with a postal or a DX (document exchange) address as appropriate. An e-mail gateway can be enabled if required. The user can also opt to send each letter directly to a fax gateway, transmitting the letter by facsimile without ever having to print it.

Once printed, or faxed, a letter document can be marked as having been processed and stored in the letters database as a record of correspondence.



Figure 6: An example letter from the letters database.

### *The reports database*

The High Court's annual report includes 30 different tables analyzing the judicial workload over different periods of time. Each of these tables can be generated by the reports database, which interrogates the cases database for the information it needs.

### *The private database*

Like most documents kept on the paper file associated with each case, most of the information stored in the case management system is public information. Information, which is not public, is stored in the "private" (classified) database. So, for example, if a case involves the custody of children, the parties to that case are known only by their initials. The appropriate party document is changed so that only the initials are kept in that document: the party's real name is stored in a new document in the private database, with a link between the party document and the private document (see Figure 7). Only users with a high security level can access the private database, so only those users can find out the real names of parties who are known by their initials.

The screenshot shows the Lotus Notes 'New Party' form. The main form is titled 'Applicant 1.6' with a reference number 'P3/1998'. It includes fields for 'Type' (Natural Person, Non-Natural Person, Office Holder), 'Family name' (Citizen), 'Given names' (John David), and 'Capacity'. There are also radio buttons for 'Representation' (Represented, Unrepresented, No appearance). Below these is a 'Representation' section with 'In person' information: 'Mr John David Citizen Ph : (02)9999 8888'. There are 'Add', 'Goto', 'Delete', and 'New' buttons. A 'Notes' section is also visible. A 'Party Alias' dialog box is open, showing fields for 'Family name' (Citizen), 'Given names' (John David), 'Capacity', and 'Alias' (J.D.C.). The dialog has 'OK' and 'Cancel' buttons.

Figure 7: One of the links between the cases database and the private database. In this (fictitious) example, an alias is about to be created for a party. The party's full name and alias will be stored in the private database, and will be accessible (via the cases database) only by users with a sufficient level of access. After the "OK" button has been clicked, only the alias will be stored in the cases database.

Storing sensitive information in a separate database, means that users cannot use Notes's powerful full text searching capability to circumvent restrictions the system places upon their access to information. A search for "Smith" would not return a party document for a person with that family name if that person were to be known by an alias because the word "Smith" would not occur in that party document (as a visible or a hidden field). A user could only connect the word "Smith" and that person's party document via the private database.

### Making Notes (slightly) relational

In a traditional, relational database, information can be thought of as being stored in tables. Every record (or row) in a table is distinct due to a unique field (or combination of fields) called the primary key. Every field (or column) has a distinct name, but the same field can appear in many tables. The requirement of referential integrity in a relational database ensures that tables within that database are consistent with each other.

A Notes database is not a relational database. Much of its flexibility and power derives from its being freed of the constraints of relational databases. This distinction is best illustrated by example—an example which demonstrates how case management system developed for the High Court makes use of Notes's flexibility, but also forces Notes to behave like a relational database in some ways.

In every party document in the cases database, information is displayed about that party's legal representatives (unless the party is self-represented). In addition to the name and contact number for the firm of solicitors representing that party, there is displayed the name and contact number of the solicitor on the record (that is, the practitioner with carriage of the matter) and that party's counsel.

If a practitioner's phone number changes in the practitioners database (because she/he has moved firms) that change is reflected in the information displayed in the party document only if the practitioner is that party's counsel—not if the practitioner is that party's solicitor. The system behaves like a relational database (ensuring that that practitioner's new phone number appears beside her/his name wherever it is displayed throughout the system) only if the practitioner is a barrister; Notes's freedom from referential integrity means that a different phone number might appear beside a given practitioner's name in several party documents.

This asymmetrical treatment of practitioners reflects the difference between barristers and solicitors. If a barrister changes phone numbers, she/he is still likely to be the appropriate contact for a given party in a given case. If a solicitor changes phone numbers, she/he has most likely changed firms; although solicitors sometimes take their clients with them when they move firms, it is more likely that the appropriate contact number for that party will remain unchanged because the solicitor's replacement (or someone close to that telephone) will have carriage of the case. (Of course, if a solicitor changes numbers and takes a client with her/him, the contact number displayed in the party document can be updated.)

## Events

For the case management system, an *event* is something that happens in a case. These events fall into one or more of three categories:

- a document having been filed (e.g. an application for special leave to appeal);
- correspondence having gone out from the Registry (e.g. a draft index to an appeal book having been sent to all parties for comment); or
- something having happened (e.g. a case having been heard by the Court).

An event can be shared by more than one case. So, for example, when several cases are heard together, a single hearing event—a single event document—is linked to each of those cases.

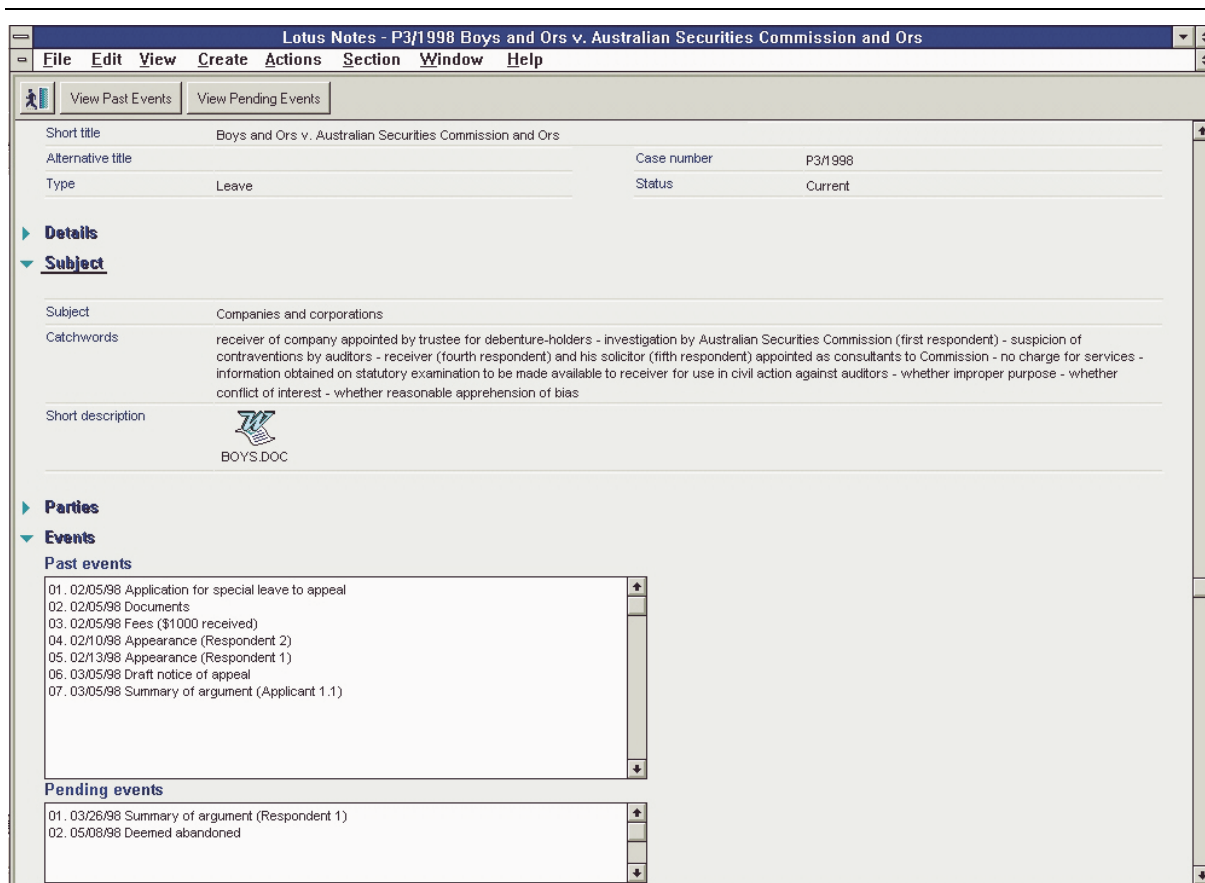


Figure 8: An example case document, from the cases database, showing the lists of past and pending events.

An event is either *past* or *pending*. A past event has happened; a pending event may happen. Each case document has two scrollable windows (one for past events, one for pending) which list all of the events attached to that case. This allows the user to see, at a glance, the status of the case: what has happened in the case thus far, and what is expected to happen next (see Figure 8).

An event becomes past when the user confirms that the event has “happened”. An event becomes pending when it is created by a user (because she/he expects that it will happen, and wants to indicate that expectation on the system) or it is automatically created by the system because another event has been set to past (because the normal sequence of events, for this type of case, suggests that the pending event will probably follow the event which has just happened).

When the user creates an event for a case, she/he has more than 50 basic events from which to choose. The system assists the user in this choice by suggesting the next most likely event from a list giving the normal sequence of events for cases of that type (see Figure 9). If the required event is not in this first list, the user is able to choose from a list of all events that can occur in cases of that type. Finally (for extremely unusual events) the user is able to choose from a canonical list of all possible events, for all possible case types.

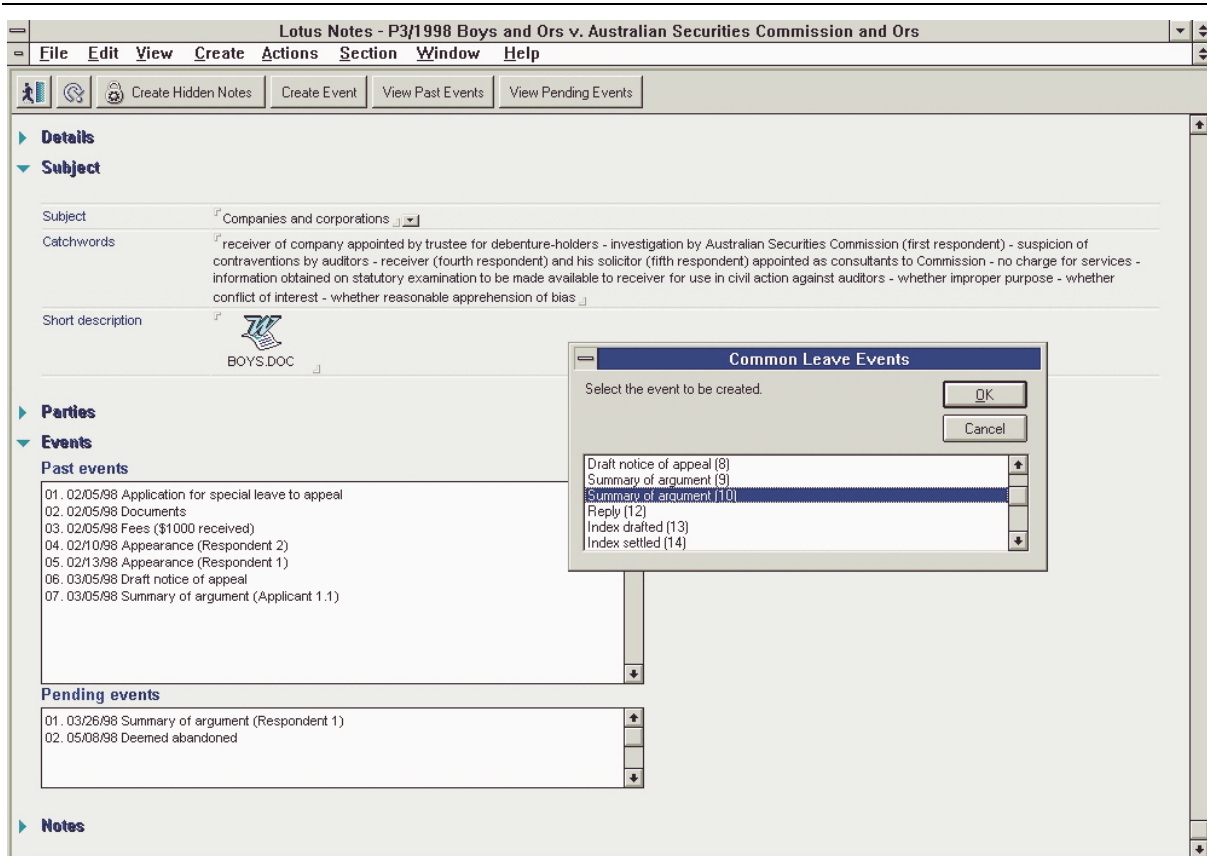


Figure 9: An example case document, with a dialogue box (brought up because the “Create Event” button was clicked) listing the events most likely to occur in a case of this type (an application for special leave to appeal). When the dialogue box is brought up, the event initially selected is that most likely to occur next, given the events that have already occurred in this case.

Sometimes the link between two events is stronger than merely the occurrence of one suggesting the occurrence of another. For example, when an index to an application book is settled (i.e. when an “index settled” event occurs) the application book becomes due at a specified time (i.e. an “application book” event is expected). So, when an “index settled” event is set to past, an “application book” event (with an appropriate date) is added to the list of pending events for that case (see Figure 10). As well as clearly indicating the status of the case—that the index has been settled and the application book is due—the automatic creation of pending events in this way reduces the amount of data entry required of the user by filling-in as many as possible of the fields in the pending event when it is created. For example, the likely date, and filing party of the “application book” event can be filled-in based upon the values of fields in the “index settled” event when that event is set to past.

The fields that make up a given event document vary depending upon the type of the event, allowing various information to be stored about different types of event.

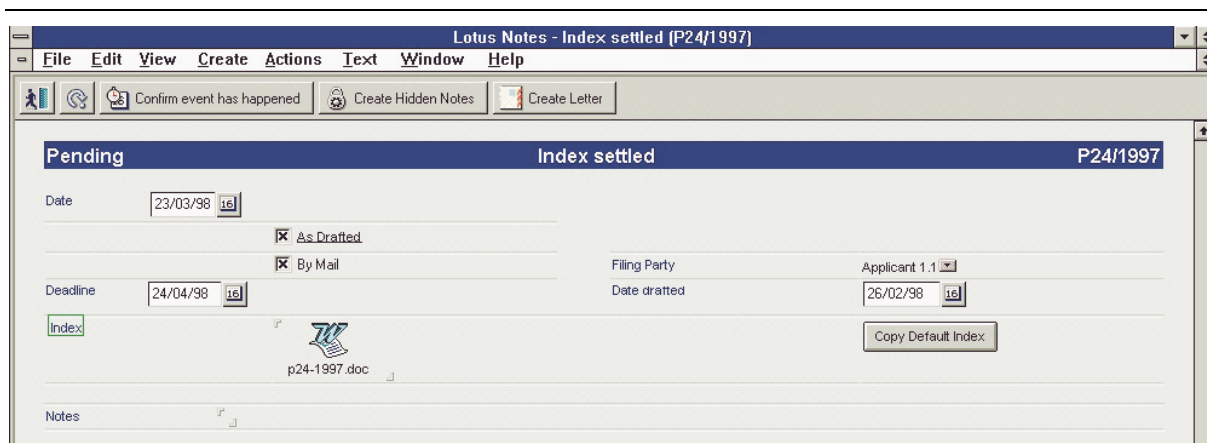


Figure 10: A pending “index settled” event document from the cases database. When the user clicks the “Confirm event has happened” button, the event is set to past and a new (pending) “application book” event is created with various fields set on the basis of the values of fields in the “index settled” event.

The use of events means that on-line copies of transcripts, for example, can be attached to the event document logically associated with that transcript (a “hearing” event) rather than to the case document itself. Notes’s full text searching mechanism means that a search for a string which occurs in the transcript attached to a “hearing” event will return that event, giving the user information about the hearing as well as ready access to a copy of the transcript itself.

Similarly, if a party or its legal representatives were to create and electronically file a document in the Court (a process not presently permitted by the High Court Rules) that document could be automatically attached to an appropriate event document in the cases database. Like any document in the database, the electronically filed document (which would not have to be a Notes document) would also be indexed for Notes’s full text searching.

## Views

Several different views of the documents in each database are available. Users can access different views, depending on their needs. Some of these views restrict the documents shown (e.g. showing only documents for current cases); some sort the documents on different bases (e.g. by case type; by responsible officer); and some do both (e.g. showing only current cases, sorted by case type). Still other views display documents of different types, grouped in a logical fashion (e.g. events, grouped by case; parties grouped by case). A view can list documents (see Figure 11), or display them in a calendar format (see Figure 12).

Of course, none of these views changes the way in which the data is stored in each database. They merely provide different views of the same data, allowing the user to choose the most convenient view for her/his purposes.

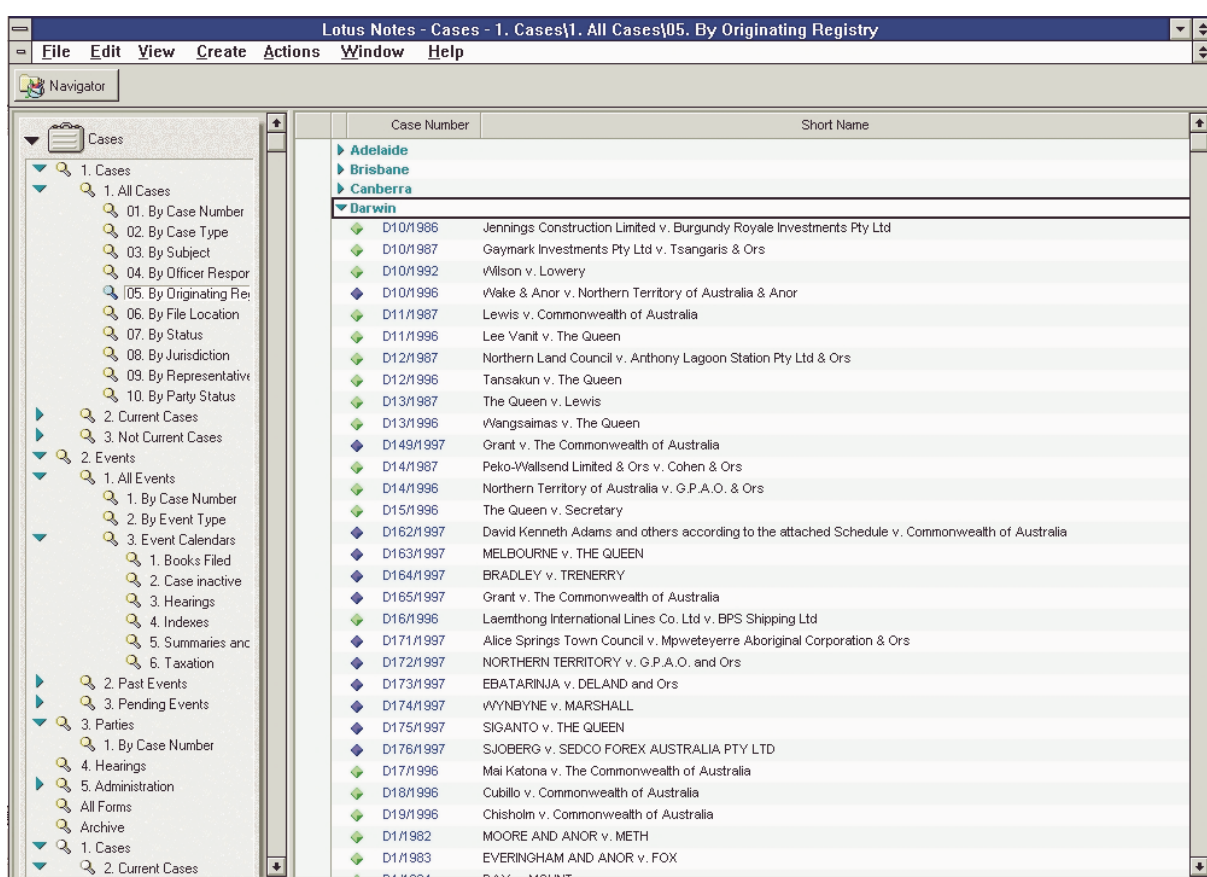


Figure 11: An example view of documents in the cases database: current cases, grouped by their originating Registry.<sup>3</sup>

## Security

Notes has sophisticated built-in security mechanisms. In addition (as discussed above) information held in the case management system which is not public is kept in the private database, reducing the sensitivity of the other databases.

The system requires that each user be assigned to one or more of seven *groups*. Each group of users has specific “access rights” assigned to them. A user’s level of access determines what information she/he can read, and what information she/he can change, right down to the field level.

For example, the composition of the Court which will hear a particular matter will be displayed only to a user with a certain level of access when she/he views the relevant pending “hearing” event. However, when viewing that same event, a user with a lower level of access will see only the number of justices who will hear the matter, and not their names.

The main users of the case management system are Registry staff, though other Court staff and Justices have access. Generally, only Registry staff have levels of access sufficient to change information, whilst other users can only read information.

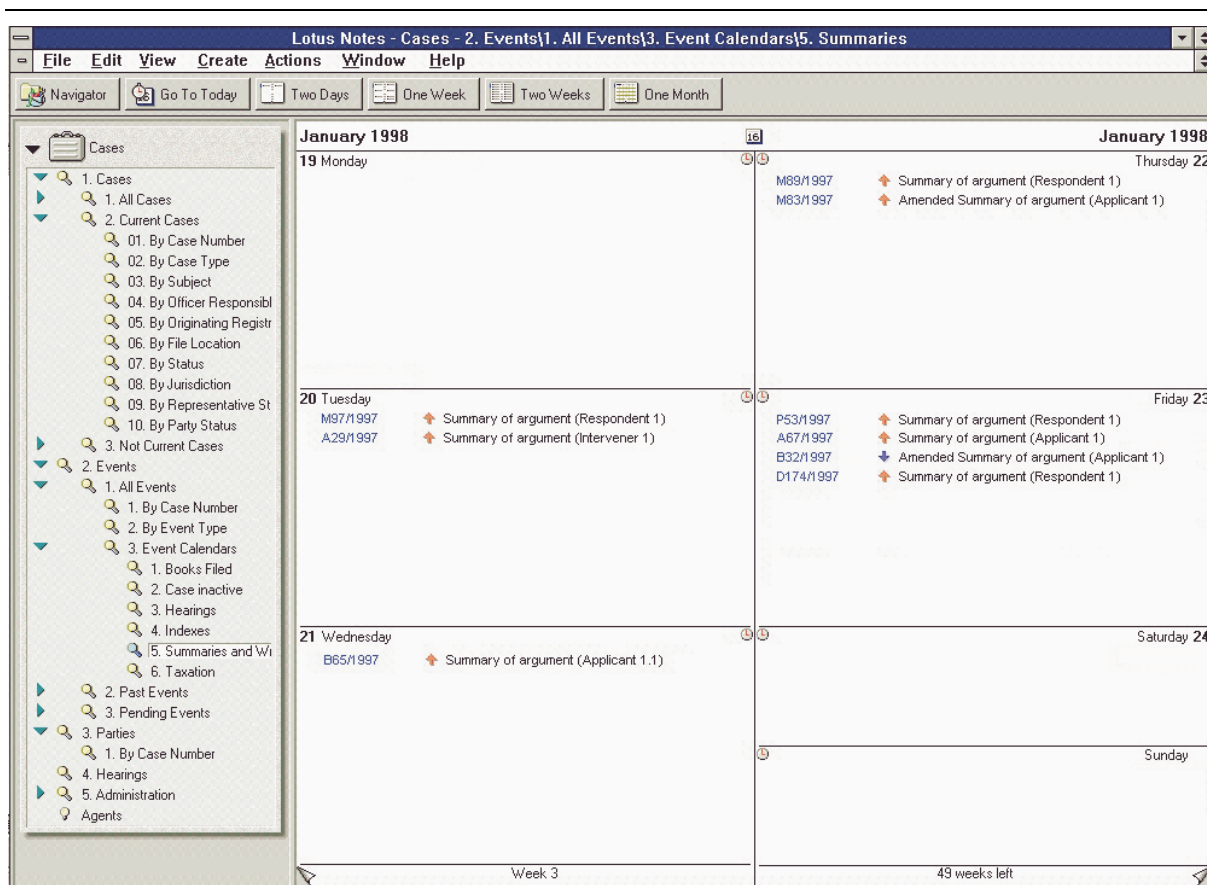


Figure 12: An example calendar view of documents in the cases database: summaries of argument. The red up-arrows indicate past events; the blue down-arrow indicates a pending event (in this example, an overdue one).

## Ad hoc queries

In addition to the generation of statistical tables for the High Court's annual report (discussed above), the system allows the use of Notes's full text searching capabilities to generate ad hoc queries. Whilst packages exist which allow powerful statistical analysis of Notes databases the Registry's needs have to date been met by Notes's simple search interface which allows the user to construct, and save for future use, complex search queries.

## Delivering Notes to the desktop

The Court faced two significant problems in delivering the new system to the desktops of users.

- *Operating System Requirements.* The latest version of Notes (version 4.6) requires the client to be installed on a 32-bit operating system. The Court Registry had a predominance of Windows 3.11 terminals and old 486 PCs. If Notes were to be run on each desktop using traditional client/server architecture, all of the PCs used in the Registry would have to be replaced with new equipment.

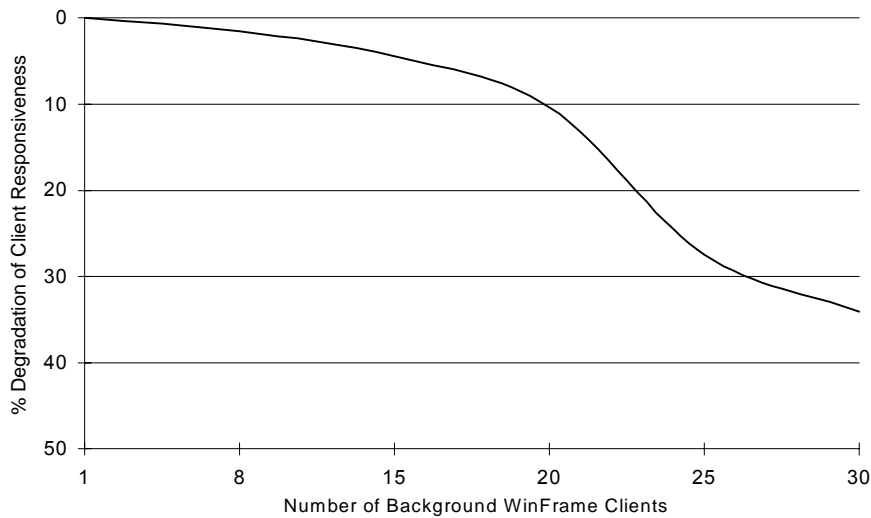


Figure 13: Performance degradation using WinFrame with 1–30 concurrent users.<sup>4</sup>

- *Remote Access Performance.* Judges and Court staff who are geographically isolated from the Court's Local Area Network (LAN) faced delays in accessing applications that run in traditional client/server architecture. (This problem is not unique to Notes, and affects many applications that use a client/server architecture.)

Emerging thin-client technologies such as Citrix WinFrame offer a solution to both of these problems. Thin-client technologies allow remotely based staff to access bandwidth-hungry applications over both standard telephone lines and ISDN connections at performance levels that would normally only be achieved over a 10 Mb per second network connection. Additionally, these technologies allow applications that require a 32-bit operating system to be accessed from desktops using older 16-bit operating systems.

The bandwidth problem stems from the fact that client server applications perform much of their processing of data on the client workstation rather than on the server. This normally requires the installation of a large and complex client. Often the sheer quantity of data passing between the client and server across either an ISDN or dial-up modem connection results in sluggish performance.

Citrix WinFrame allows a user located in another State to take control of a *virtual computer* inside a separate dedicated server located in another location, which will then complete all data processing locally. Importantly WinFrame allows all data processing to be completed locally. The latest version of the software operates by slicing the application's logic, sending only interface information across the connection. Once operational, any 16- or 32-bit application can be accessed remotely at performance levels comparable to those obtained within a 10baseT LAN using structured cabling.

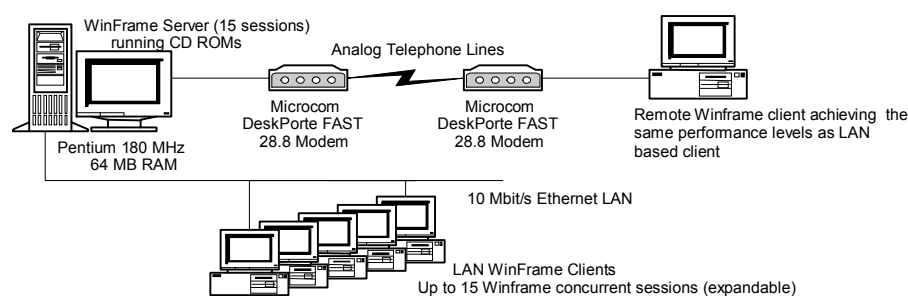


Figure 14: The High Court's WinFrame LAN/remote access configuration.

Whilst a number of different remote control software products were available (such as NTrigue), Citrix WinFrame<sup>5</sup> running on a modest Windows NT server<sup>6</sup> was eventually selected as the preferred solution for the High Court.

The WinFrame licensing arrangements chosen by the Court allow up to 15 concurrent sessions on the Citrix server. There is no significant degradation in response times with this many concurrent users (see Figure 13). Additional user licences can be added at a later stage if necessary; a large Canberra-based organization<sup>7</sup> recently commissioned a 300 concurrent user Citrix WinFrame server.

The High Court's WinFrame LAN/remote access configuration is shown in Figure 14. A user can access the server from an ISDN, dial-in modem or Internet connection using an Intel-based PC (286 or above), Macintosh or Unix workstation. Once connected by modem, 32-bit applications can be run at nearly the same speed as would be achieved if the user were actually connected to the LAN.

One of the benefits of the WinFrame solution is greatly simplified administration of applications. In a traditional client/server architecture, the arrival of a new application or application upgrade would generally demand the immediate reinstallation of a new client on all users' PCs. This is a time-consuming exercise, particularly if the client is physically located in another State. As the Citrix WinFrame solution requires the various application clients to be installed on a single processor, applications, software updates and patches can be rapidly and easily deployed.

Utilizing WinFrame, users can still cut and paste text between local and remote applications, and local printing is still possible. WinFrame allows each user to have a unique profile recorded on the server. User access, screens and printing options can be fully customized. In effect, a "virtual" PC desktop is configured for each user.

One notable advantage of a WinFrame solution is a significant extension of the life of existing hardware. For example, an organization with a large collection of old 286/386 PCs, and heritage cabling infrastructure, could install a single Citrix WinFrame Pentium server and simulate the processing speed of the Pentium computer across all WinFrame client workstations. Users could access the latest 32-bit resource-hungry applications from their existing 286 PCs, without memory or processor upgrades.

## **Publishing selected information to the Internet**

The next phase of the project, pending approval from the Court, will involve making selected information from the case management system available over the Internet to any person with a web browser. If implemented, this will doubtless prove of great utility to parties and interested members of the public, and should reduce the number of routine enquiries about the status of various cases presently dealt with by Registry staff.

Web access to this information would be made possible by regularly copying, to a proxy server, selected information from the databases which make up the case management system. A Notes Domino server on that proxy server would respond to connections from web browsers by dynamically producing hypertext markup language (HTML) documents.

One of the strengths of Notes is its ability to publish nominated objects from within a Notes database dynamically to the Internet. Unlike many competing relational database products electronic publishing to the Internet can be done seamlessly and without the need for third party products. Notes allows Internet publication down to individual database field level.

Subject to the Court's approval, the Notes server would be programmed to replicate nominated fields to a second web server on an hourly basis during business hours. The use of a proxy server to publish information to the Internet would guarantee the integrity of the primary Notes server: no direct connection to the server from the Internet would be possible. At worst, a hacker could gain access only to the proxy server. Any fields on this server that had been damaged or changed would be restored at the next replication.

Whilst the decision whether to proceed with this phase of the project is yet to be taken, Notes does include native web publishing and server replication facilities. The ability to publish selected fields, including embedded objects, is an added advantage.

## **Conclusion**

The High Court's new case management system has been operational since the beginning of 1998. The system has already assisted workflow and improved management of cases within the Registry. It has also greatly simplified the generation of sophisticated correspondence, which has increased the amount of information regularly supplied to parties explaining the status of cases and the procedures that have to be followed.

The system contains data, migrated from the old case management system, for cases filed in the Court as early as 1980. The cases database has been developed to allow three types of case record: *full* (for cases active since the new case management system became operational), *partial* (with a restricted set of possible events, for cases whose details were stored in the old case management system), and *minimal* (for cases for which the Court has no on-line information). These minimal case records could be used to store limited information on every case filed in the Court since its establishment in 1903 for which there is not already information stored in a full or partial case record. In this way, the new case management system could encompass the Court's entire past.

Of course, the system has also been designed with regard to the Court's future. The decision to develop the new system using Notes was significantly motivated by a belief that Notes will be available, and will continue to be improved, for many years to come, and that people skilled in the maintenance and improvement of Notes applications will continue to be widespread. Notes has rich functionality and, to the extent possible in today's rapidly changing world, provides a "future-proof" environment within which to operate a powerful and complex case management system.

---

<sup>1</sup> Available on-line at <<http://www.highcourt.gov.au/AREP96-7.HTM>>.

<sup>2</sup> "Groupware" is the IT infrastructure that allows work groups to work as a unit toward common goals. It is premised on the assumption that productivity can be improved in an environment of frictionless information exchange and seamless communication throughout the entire organization.

<sup>3</sup> In this figure, the blue diamonds indicate *full* cases; the green diamonds indicate *partial* cases. The difference between *full*, *partial* and *minimal* cases is explained in the "Conclusion" section of this paper.

<sup>4</sup> The Tolly Group (1996).

<sup>5</sup> WinFrame has been developed as an authorized extension to Microsoft Windows NT, under license from Microsoft. See <<http://www.citrix.com>>.

<sup>6</sup> An Intel Pentium Pro 180 MHz, with 64 Mb RAM.

<sup>7</sup> The Australian Federal Police.