Audata Limited

www.audata.co.uk

Customised or Commercial Off-the-Shelf Software?

Costs and Benefits of Customising Your EDMS

David V. Bowen and Gary Johnston, Audata Limited.
4 January 2005

Most of the Electronic Document and Records Management systems (EDRMS) approved by The National Archives (TNA) are commercial off-the-shelf software (COTS) packages. This means that the client organisation licences the computer code from the vendor, and then hires consultants (or uses their own staff) to configure the software. Configuring includes (for example):

- Defining retention schedules
- Defining record types and record series
- Defining security models and hierarchies

Configuring software does not require software engineering skills; you do not need to write computer code.

Most clients, however, find that they want one or two features which are not present in the basic software. They have to select among several options:

- Do without the desired feature, or develop a workaround
- Petition the vendor to add the feature to their COTS package
- Write an additional software module, using the software development kit provided with most EDRMS
- Write an external module, independent of the EDRMS

This paper outlines some of the issues involved in this choice, against the background of four actual cases.

Do Without or Workaround

User compliance is a key issue in any EDRMS implementation. If users can be shown that they benefit from the EDRMS (as individuals), they will be more likely to use the EDRMS enthusiastically. A successful, rapid implementation will show clear benefits for the organisation and for individual staff members. The benefits for individuals are especially important since almost every EDRMS implementation requires extra work from those staff members who create documents or records. Everyone creating documents must add metadata (index) terms, and must think more carefully about their document titles. This extra work pays off when people search for documents, but many staff recognise that they are being asked to do more.

If users really need a feature, they will be unhappy if they are asked to do without it, or if an inconvenient workaround is offered. This unhappiness may affect user compliance, and may even jeopardise the implementation of the EDRMS.

The “Do Without” option is really only useful if the feature is “nice to have”.

Petition the Vendor for an Upgrade

Vendors are normally eager to hear any requests from clients for upgrades or additional features. Such a request will be met with interest, but the upgrade will often be issued after two or three years. This delay raises the same user compliance issues discussed above.
Vendors require significant time to build a request into an upgrade for several reasons:

- They need to see if other people want the same change
- They need to plan the change into their software development roadmap
- They need to follow a full software development and testing process

It may take six months or more just to find out if other clients (worldwide) want the same upgrade. Getting clients to agree on the priority of each possible upgrade is even harder. One person's essential upgrade is another person's "nice to have". Indeed, there is usually a third person who hates the suggested upgrade and would never use it.

Most EDRMS vendors operate a formal software development process, with a roadmap of planned changes and new features. Often this will be dictated by expected changes in the IT infrastructure, as well as by planned changes in the EDRMS software. Vendors have to be ready for each new MS Windows® release, and for changes in the major databases or storage systems. This means that even once an upgrade is accepted as widely wanted, it may take 12 – 24 months to go through the software scheduling, development and testing process.

Finally, there will be a delay while the whole release (including the desired upgrade) is tested, and then marketed. This can add a further 6 to 12 months.

Add an SDK Module

Any good TNA-listed EDRMS will come with a software development kit (SDK) or an application programming interface (API). These allow an external application to be created (for example, in Visual Basic) which has access to the commands and the data of the EDRMS. This gives the client organisation the best of both worlds: quick response and support.

There are real advantages to this approach:

- The module is created quickly and can be implemented along with the main EDRMS
- The cost of such a module can be low
- The module will be supported by the vendor or their partner
- New features will be added by the vendor or their partner

Modules, created with a vendor's SDK, can be written in as little as six weeks. This allows for formal project documents, including Requirements Specification, Design Specification, Test Plan and Test Results. It allows for testing in the environment of the client organisation. Note that there is a significant time and cost saving, since the module does not need to be tested for ALL environments of the vendor's EDRMS, but only for the client organisation's environment.

Modules created in this way can be done for a few thousand pounds. This will often be 1% to 5% of the EDRMS project cost, and may be even less. If the benefits have been reviewed and proven to be real, then this is often good value. In addition to the real benefits of the new feature, there will be benefits of improved user acceptance.

Modules created with a vendor's SDK will be written by the vendor themselves, or by one of their consultancy-partners. In either case, they will support the new module. The vendor will ensure that their SDK is kept up to date as the EDRMS software, or the underlying desktop or database software change. They or their partner will update the module as necessary.

The partner (or vendor) will hope to sell the module to other clients, so they will also be alert to new features which may be useful. This means that they will continue to take an interest in a module created with the SDK of the EDRMS.

Finally, as a module shows that it is popular, the vendor may then add that function into its main software. The client organisation gets to use the new feature very quickly, and they see it incorporated into the basic COTS software after a few years. They have also influenced the design and function of the new module, so it is more likely to do what the client organisation wants.

Add an Independent Module

A fully independent module is generally a poor way to achieve a new function around an EDRMS. Such a module will need to be maintained by the client organisation. Neither the EDRMS vendor nor their consultant partners will have any responsibility for it (nor any interest, either). The full cost of modifying the module to fit into changes in the EDRMS, or in the IT environment, will fall on the client organisation.

Also, most new features give rise to requests for more new features. Once users see that X is possible, they wonder whether Y would also be possible (and useful or even essential). An independent module requires that the client organisation take on the full burden of designing and creating these additional features.

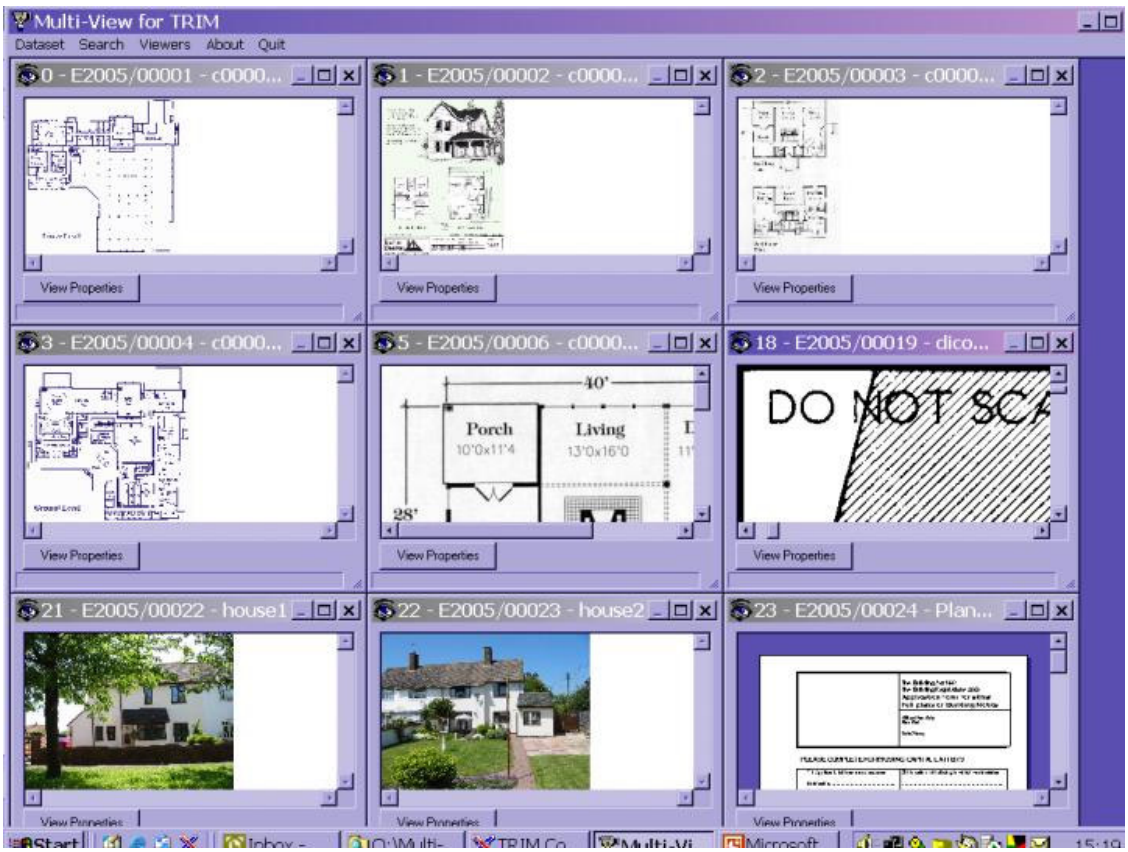
The total cost of ownership of an independent module is rarely justified by any benefit.

An exception to this may be when the database of the EDRMS can be linked to another database, or can be accessed by a readily available tool such as MS Access™. Clients also often use Business Objects ® or Crystal Reports® to manipulate statistical data from their EDRMS.

Multi-view

As part of Audata's implementation of TRIM® at the National Maritime Museum, we discussed searching for images. Some users expressed a clear need to see more than one image at a time. If you searched for "Nelson" (or "junk") you got so many images that selecting the right one required side-by-side comparison.

The TRIM SDK was used to build an application which would display up to 16 images on the screen at one time. The full TRIM metadata are available for each image. The search screen and search syntax are identical to the searches that users do within TRIM, so no new user training is required.



Multi-Select

Audata and Getronics BV worked together to implement an EDRMS for an international court in the Netherlands. This court needed to index documents with mixed metadata items taken from a picklist. For example, a document might refer to three countries: their names should all be selected from the official “Country names” list. Another document might include sections in two different languages. The languages needed to be selected from the official “Languages” picklist.

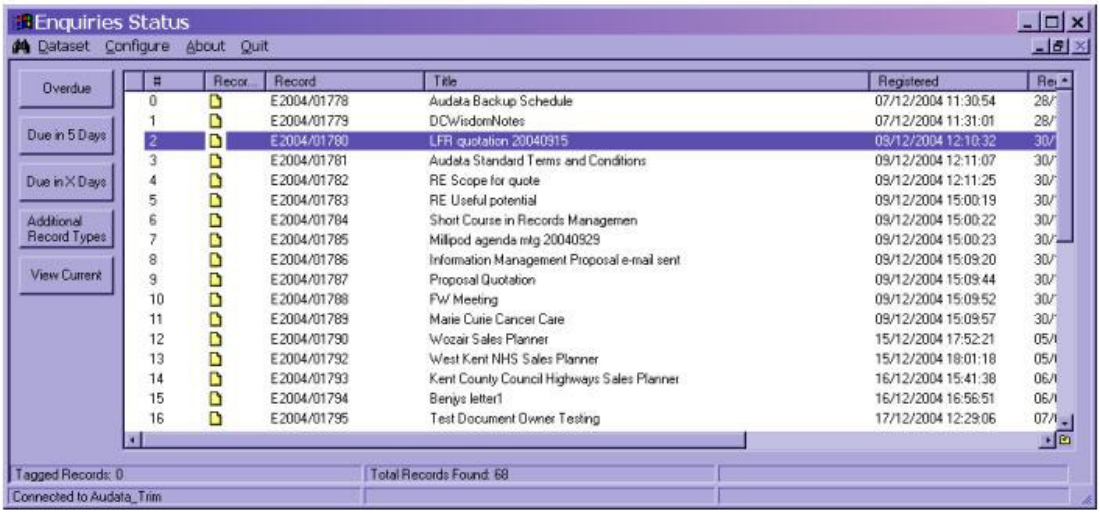
Most EDRMS picklists will only let users select a single entry. We used the SDK to create a modified picklist which allowed the user to select more than one entry.

A more difficult part of the programming was to modify searches so that they would retrieve the document when any one of the picklist entries was used as a search term. (Otherwise users might have to search on the exact combination of languages which was entered in the metadata. This was an unreasonable limitation.) For example, a document might be received in Arabic, with a French interlinear translation. Even if a traditional picklist allowed the user to enter “Arabic and French” as languages of the document, it would have required any search to enter “Arabic and French” as the search term. A search on Arabic or French alone would not retrieve this document. Clearly they should, and Multi-select solves this problem.

Freedom of Information Tracking

Audata and Computer People Ltd. have recently implemented TRIM for the Royal Parks Agency. The RPA wanted to be able to track Freedom of Information (FoI) enquiries, and to be alerted to any which were older than 15 working days.

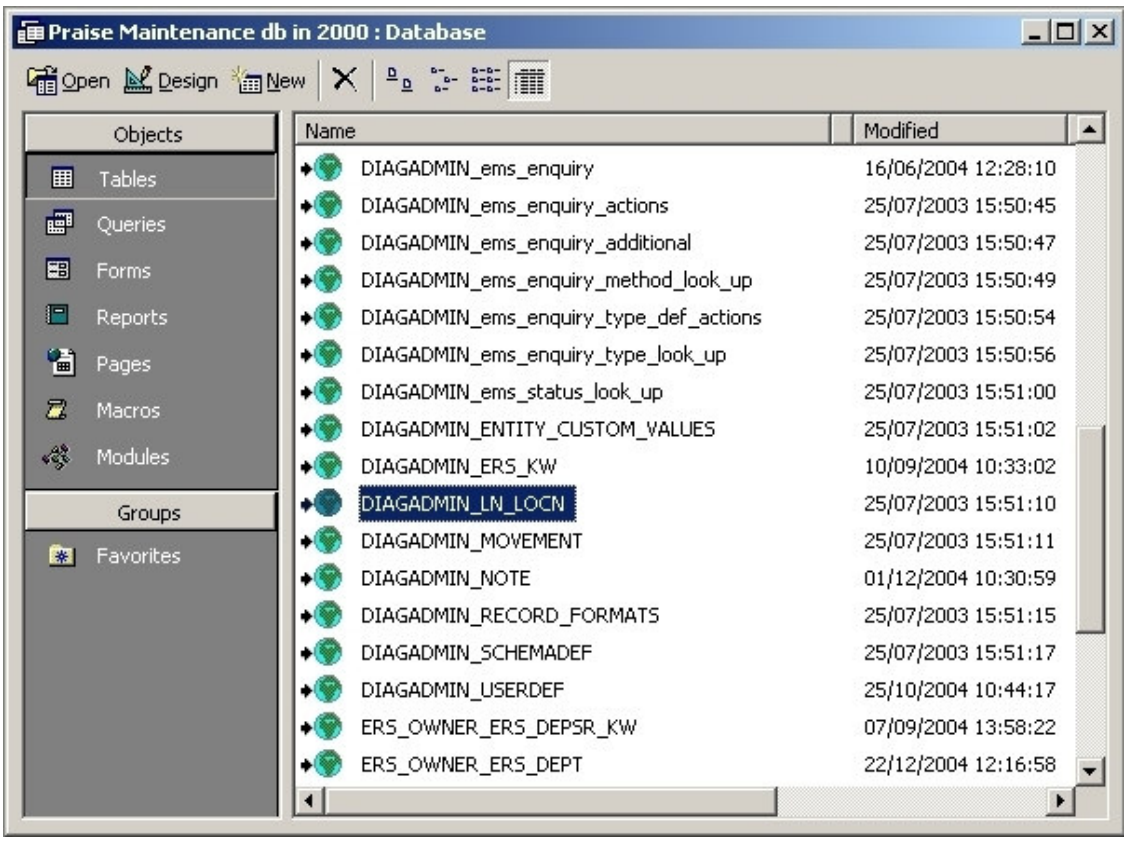
In order to understand “working days”, and to be flexible about the period of warning, an external module was required by the client. The SDK was used to write a simple routine which allows the FoI enquiries to be searched using the number of working days since they were registered (received). The non-working days are held in a simple text file, so it is easy for the client to update this if any unexpected holiday is declared.



Reporting and Location Management

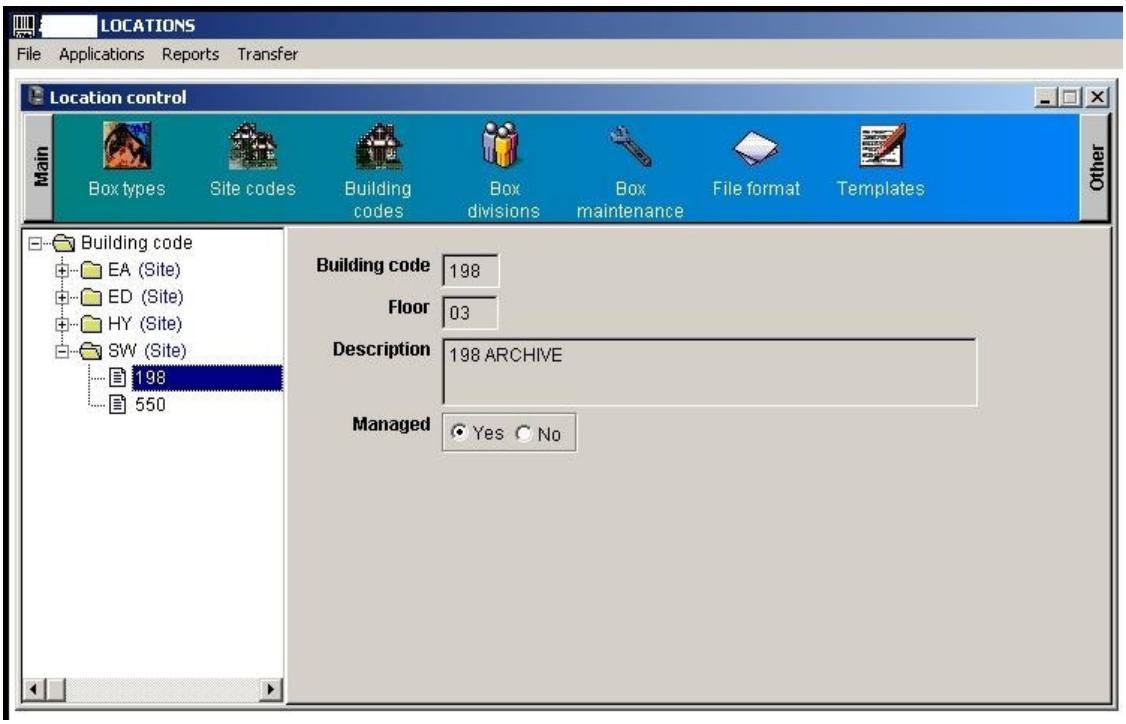
Audata has implemented Wisdom™ from Diagonal Solutions Ltd. for a large pharmaceutical company. Two functions required external links to the Wisdom database: statistical reports and location management.

Although Wisdom has reporting functions, the client wished particular report formats and styles which were difficult to achieve within Wisdom. These were easily implemented by linking MS Access to Wisdom’s database. The required queries and reports could be obtained via MS Access. Although the programming required was difficult, the module has paid for itself already. Also, since Wisdom is closely aligned with Microsoft Office® applications, the module will be easy to keep up to date as Wisdom and the desktop Windows environment change.



The storage provider who looks after the archive of paper records for this site has Location Control software that knows which shelf space each box is stored in. The archive is so big (tens of thousands of boxes) and so active (regular accessions and disposals) that boxes are stored randomly on the shelves.

Each box and each shelf space is barcoded, and the Location Control software knows which box is in which shelf space. (It also knows which shelf spaces are empty.) When a file needs to be collected for a user, or for review, the software knows where the correct box is. When a box is ready to be returned to the warehouse, the software knows where there is an empty space to out the box.



When a group of files is requested by the client, the EDRMS creates a list of the box numbers in which these files are held. A link then allows this list to be shared with the Location Control software. This software prepares a picklist, which is sorted into the correct order so operators do not need to zig-zag about in the warehouse.

Conclusions

Experience of several EDRMS implementations shows that each must be considered as a separate project, on its own merits. However, several EDRMS client organisations have been able to specify and obtain simple modules which give them essential functionality. These can be provided rapidly, at low cost. They are often a cost-effective way of meeting organisational needs and improving user acceptance.