



Open Research Online

Citation

Ince, Darrel (2010). The Re-development of a Problem System. Technical Report 2010/04; Department of Computing, The Open University.

URL

<https://oro.open.ac.uk/90279/>

License

(CC-BY-NC-ND 4.0) Creative Commons: Attribution-Noncommercial-No Derivative Works 4.0

<https://creativecommons.org/licenses/by-nc-nd/4.0/>

Policy

This document has been downloaded from Open Research Online, The Open University's repository of research publications. This version is being made available in accordance with Open Research Online policies available from [Open Research Online \(ORO\) Policies](#)

Versions

If this document is identified as the Author Accepted Manuscript it is the version after peer review but before type setting, copy editing or publisher branding



T e c h n i c a l R e p o r t N ° 2010/ 04

The Re-development of a Problem System

Darrel Ince

17 May, 2010

Department of Computing
Faculty of Mathematics, Computing and Technology
The Open University

Walton Hall, Milton Keynes, MK7 6AA
United Kingdom

<http://computing.open.ac.uk>

The Re-development of a Problem System

Darrel Ince
Dept of Computing, Open University
Walton Hall, Milton Keynes
MK7 6AA
d.c.ince@open.ac.uk

Abstract

The Integrated Children's System is a set of computer programs that are used to chronicle the interaction between British social workers and children in need. It has been the subject of many criticisms from social workers, a major trade union and social work academics. This article details a redevelopment of the system that removes the problems with the system and demonstrates the advantages that can be gained in development by employing readily available open-source software. The redevelopment ignored the implementation directive in the specification for a record-based system. The advantages are twofold: first it removes the major problems with the system and, second, does it for a fraction of the £80-100m original cost (the article estimates the full cost of the redevelopment to be *c* £35k). The article is important in that it details a major reuse effort which resulted in very large savings; describes an approach to software development for human-centred applications that is radically different to the forms, relational database, Java/PHP approach; and is one of the first case studies which shows the dramatic effects of employing open-source software.

Keywords: E-government, System Design, System Requirements, System Failure, Human-centered Computing

1 Introduction

The Integrated Children's System (ICS) was a computer system that was specified by the British Government's Department for Children, Schools and Families (DCSF). It was a small part of the fallout from the appalling death of Victoria Climbié at the hands of her guardians. Her death led to a public inquiry [1] which was charged with looking at how such an incident could not happen again. It was led by an ex-chief inspector of social work, Lord Laming. The inquiry resulted in a major restructuring of the way that British national and local government managed child welfare.

ICS was a result of this inquiry; it was intended as a system which helped social workers document their interaction with a child in need, their families, other agencies such as the police and other interested parties such as a child's teacher and childminder. The requirements for the system were defined by the DCSF and were specified in a number of documents:

A Statement of Business Requirements. This was a short (6 page) document that provided an excellent high-level view of what the ICS should do.

ICS Minimum Compliance Criteria. This was effectively an expansion of the previous document. Each business requirement was split into sub-requirements and criteria specified for each requirement. For example business requirement D31 specified that

Each child's record includes a unique identifier and has provision for recording similar identifiers used in other applications as is appropriate to support integrated working across agencies.

with sub-requirements 'Each child's record includes a unique identifier' and 'and has provision for recording similar identifiers used in other applications as is appropriate to support integrated working across agencies'¹ and a set of criteria including, for D31, 'The system should have the capability of recording at least three identifiers' and 'The identifier must be unique to that child within ICS within the LA'.

A set of exemplars. These were forms documents that demonstrated the documentation that should be filled in by the user of ICS and stored in a database. These are categorized as: general records, child protection forms, core assessment records, child or young person in need planning and review documents, assessment and progress records for looked after children and ICS Outputs for Court. There were 32 of these documents. A typical exemplar was the core assessment record for a child aged 5 to 10 years which was 27 pages in length.

ICS Exemplars: Common Principles. This was the shortest document. It effectively stated that the exemplars were not to be modified except in trivial ways, for example changing the order of text fields or check boxes in a form.

A logical data model. This contained 187 pages and documented entities such as 'Adoption Parental Consent Status', 'Family Member' and 'Child Service User'.

A process model. This contained 202 pages and detailed the functioning of processes such as 'Adoption Agency Give Applicant Report on Child', 'Hold Child in Need Review Meeting', 'Review Progress Against Child in Need Plan' and 'Reception and Initial Contact'.

An analysis of this DCSF documentation shows that it mandated a record-based, forms-based relational database-centred implementation; for example documents described the development of an 'IT system', records were freely mentioned, relationships between entities specified using an E-R model and the exemplars were partitioned into individual forms, for example the Core Assessment exemplar was split into 15 forms.

The British Government funded the development of ICS to £36m; a condition for a local authority accepting funding was that the the system they used satisfied the conditions detailed in the document *ICS Minimum Compliance Criteria*. This funding comprised £30m initial funding and £6m for eradicating the problems detailed later in this paper. Each local authority was allocated funding. A typical amount that was initially given was £200,00, either to buy a package or to develop a system *ab initio*²

¹Many of the sub-requirements are cut and paste repetitions of each requirement. They just represent a finer-grained, sub-component based view of each requirements with little or nothing added.

²Only one local authority declined: the Royal Borough of Kensington and Chelsea who

The cost of an ICS is likely to be much more than the government grant, for example Hartlepool Borough Council purchased an ICS system for £750,000. If this was replicated across the whole collection of (150) local authorities it would mean something close to £100m would have been spent.

A number of versions were implemented and, after some delays in about 20% of the local authorities, ICS was placed into service.

2 Experience with the system

There were major problems with the system. These have been documented in a number of articles in social work journals [4, 5, 7, 8, 9, 10, 11]; in evidence to a second inquiry into the death of a second child Baby P by UNISON, one of the leading trades unions in British social service departments [2]; and in a DCSF-sponsored evaluation of four ICS pilot projects by the University of York [4]. A summary of the problems was detailed by UNISON in their evidence to an inquiry that examined the circumstances of the death of a child known as Baby P in Haringey.

UNISON wishes in particular to draw Lord Laming's attention to the seriousness of the problems being experienced by social work staff with the Integrated Children's System. Following a number of reports of difficulties being experienced, UNISON has recently carried out a survey of branches to test how widespread the problems are; whether these are primarily implementation, software or general teething problems; or whether this is a fundamental issue with the whole system. We have found that the problems appear to be fundamental, widespread and consistent enough to call into question whether the ICS is fit for purpose.

The reports we have received show that the direct impact of the system is to delay, frustrate and disrupt the flow of work in busy and over-stretched teams. But in addition it adds to the sense of disempowerment and demoralisation among social workers when a system is imposed that hampers their work so severely, yet there is little management response.

It is worth saying at this point that the implementors of the system should not be placed at fault: apart from some normal teething problems they developed the system detailed in the DCSF requirements documents—documents which mandated a heavy, industrialized system more suitable for data entry clerks than professional social workers.

3 The Problems

The following is a categorisation of problems that have been culled from various DCSF and UNISON documents, together with problems detailed in the academic social work articles cited in Section 2. This document set includes

regarded ICS as not suitable and who developed a more holistic system which was based on the needs of the whole family (Key Decision Report 4 December 2008). This report echoes many of the criticisms detailed in Section 2.

the 2009 circular *ICS Guidance Note: Improving the Usability of ICS Systems* released by the DCSF in which it was admitted that there were major problems. The second and third problems are the most serious and relevant to this article in that the first can be solved technically while the fourth is out of the hands of IT professionals.

- *Technical problems* such as problems with auto-saves, slow response time, problems with undo functions.
- *Lack of flexibility*. The fact that it was difficult to change the system to reflect changes in the environment that it was used, for example between a prosperous rural community and a deprived community such as Haringey.
- *A major mismatch between the day to day tasks carried out by social workers and ICS*. For example, the fact that narrative-based tasks such as preparing a report or trying to get a whole picture of a child took a long time. This was further exacerbated by design issues such as an inability to display multiple windows on the same computer. This was probably the most serious problem and arose from the atomization of data that was designed into the system from the early stages of the project by the DCSF.
- *Extra technical problems*. These were problems that arose from a set of standards known as the Common Assessment Framework that were developed in the wake of the Victoria Climbié enquiry. For example certain events such as an initial assessment had to be completed in a fixed time. This was implemented in various ways in the instantiations of ICS: by sending warning emails or by displaying a traffic light symbol which either indicated green, amber or red.

One of the worrying facts that emerged from the York pilot study [4] and widely reported in the British press³ was the fact social workers could spend an inordinate amount of time at the computer working with ICS—as much as 80%—rather than carrying out core activities such as home visiting.

In 2009 the DCSF realized that the ICS was flawed and, in late 2009, issued a document *ICS Guidance Note: Improving the Usability of ICS Systems*. This document looked at how existing instantiations of ICS could be improved and described a process model for carrying out this improvement. A key property of this document was that it still saw a forms-based, record-based database solution as central.

Many of the serious problems that occurred did so because of a mismatch between the work of the social worker as an author of a chronicle describing their interaction with a child and others and the discrete nature of the ICS system where information was split into chunks. These included:

- The large number of mouse movements that need to be carried out when accessing a number of windows containing atomized information.
- Difficulties with obtaining the ‘big picture’ of a child and their relationship with his or her world.

³See, for example, This social work by computer system is protecting no one. *Sunday Times*, 24th January 2010.

- Difficulties in forming narratives for other agencies such as adoption agencies and family courts.
- An inability to keep a number of windows open at the same time. Not directly relevant to narrative, but an indication that narrative formation is carried out and, for technical reasons, found difficult.
- Free-text boxes did not have enough space, a consequence of a fixed field database implementation. Social workers reported having to spend a large amount of time editing natural language text in order to fit them into fixed-length text boxes. This was a consequence of adopting a forms-based approach which employed fixed-length text fields in a relational database.

4 A Redevelopment

This section describes an architectural redevelopment of the Integrated Children’s System. There were a number of aims in the redevelopment.

The first aim was to see whether a system could be developed that returned the power of narrative back to professionals and, at the same time, was capable of generating essential management information—even as much as the copious information that was generated by the current ICS systems in existence.

The second aim was to see whether a system could be developed in such a way that it could be used in a number of local contexts without going through the painstaking configuration management-based change processes employed by IT developers.

The third was to carry development in such a way that the eventual architecture could be used in similar chronicling applications such as the failed C-NOMIS system [3].

The fourth aim was to see whether a system could be developed using existing software with a concomitant reduction in cost.

4.1 The Architecture

A schematic of the redesigned ICS system is shown as Figure 1; There are a number of components to the design.

Element Descriptors This is a simple text file that contains the names of XML elements, the first line of the file contains an element name that tags a sequence of children in care, the second line contains the element name that tags a particular child that is in need and subsequent lines are the names of elements that refer to events that a social worker faces, for example a referral of a child in need, the development of a court order to place a child in care, a home visit to a child or a case review. Each element is associated with zero or more attributes.

An Word-processed Document Store This is a collection of word-processed documents filed under a folder associated with each social worker. Each document is associated with a single child and contains a bureaucratic header and sections of text corresponding to the events described in the element descriptors. It is worth saying that any word processor could be used, the one I employed was *OpenOffice Writer*.

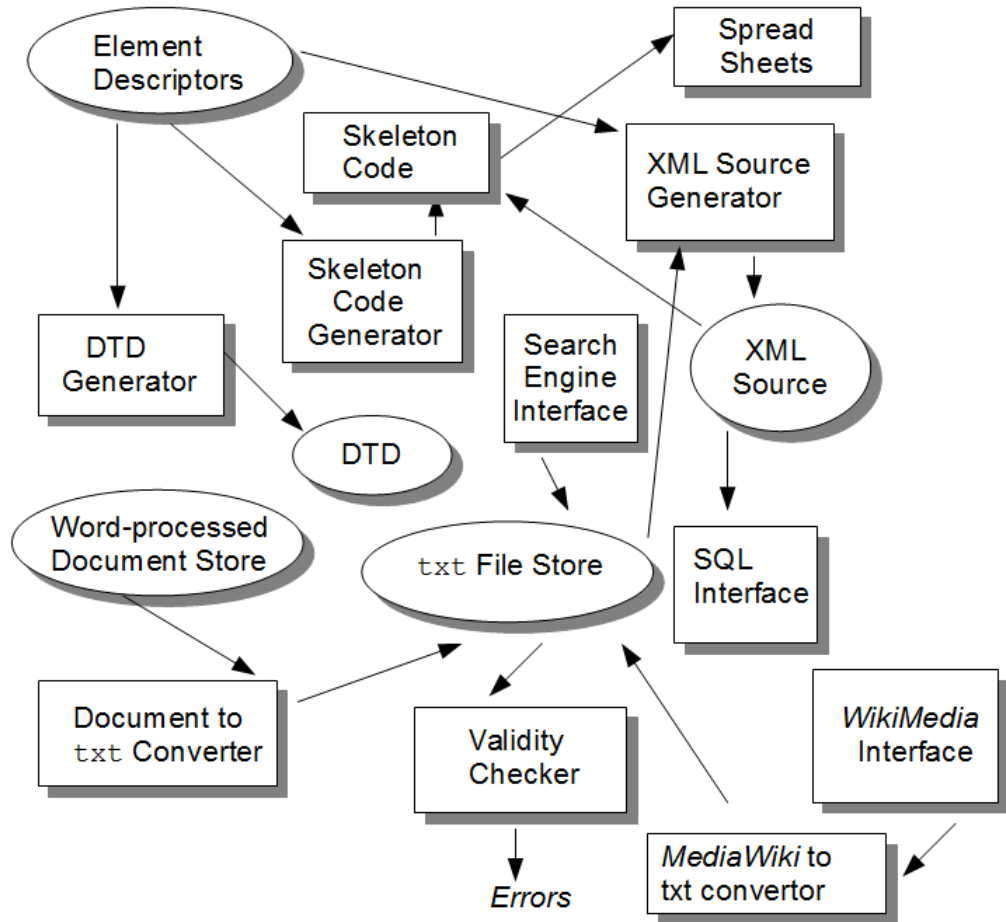


Figure 1: Schematic of redesigned ICS system

The txt File Store This is a store that contains the text versions of the word-processed documents.

The Document to txt convertor This takes a word-processed document and converts into a textual form. The document is stored in the Word-processed Document Store.

The DTD Generator A program that takes the element descriptors and produces an XML Document Type Definition (DTD). This definition defines a file associated with a child as containing a sequence of children with each child associated with one or more occurrences of events. This DTD can then be used for any further applications over and above those described here.

The XML source generator This takes the files that are found in the `txt` file store and converts them to tagged XML files.

The skeleton code generator This takes the element descriptors and produces a partial Java program that makes calls on a SAX XML API. It effectively produces a program that is a finite state machine which examines the XML source produced by the XML Source Generator and allows for code to be inserted to produce reports. The skeleton code makes available variables that: provide boolean values true and false depending on whether the source being processed is currently associated with a particular element; that hold the values of attributes; that count the number of times an event occurred; that hold the value of the string that represents the text associated with an element, for example the minutes of a case review; and, for chronological processing, the date of the previous event and the current event being processed. Code can be inserted into this skeleton to produce spreadsheet reports.

The Validity Checker This program processes the `txt` file associated with a child and carries out a number of validity checks, for example that all the symbolic punctuation marks associated with the form are present and that there are no missing attributes. Examples of these marks are described in Section 4.2. These punctuation marks enable the conversion of the `txt` file into the correct XML source.

A search engine interface This is an interface to the *Lucene* search engine [6]; it would be used if any free text searching was required.

An SQL interface This was an interface to an SQL database. This would enable reports to be generated using conventional query languages.

A *MediaWiki* interface *MediaWiki* is open-source software that is used to implement the *Wikipedia* encyclopedia. It can be used as an alternative to the word-processed document interface. The chronicle of a child's interaction with social services is held as a *Wikipedia article* which would be gradually added to as events relevant to the child occur. There are a number of security features in *MediaWiki* that enable a number of levels of security. For this application overwriting by others apart from the relevant social worker and, perhaps, their line manager can be easily enforced.

4.2 Using the System

Using the system does not entail the use of the large number of windows-based forms associated with current versions of ICS. All that would be required would be for a social worker to open a short word-processor template or *MediaWiki* template that describes an event and copy it to the end of the word-processed file or *MediaWiki* article that contains the chronology of the interaction of social services with the child. The document is then filled in. An example of such a document is shown as Figure 2 for the word-processed version and Figure 3 for the *MediaWiki* version; the symbols `MajorWorry`, `StartOfTextMajorWorry` and `EndOfTextMajorWorry` are punctuation marks used by the XML Source Generator to produce XML source that describes a social services interaction with a

child. Figure 2 also shows three attributes Date, DateOfLineManagerMeeting and LineManagerSeen.

```

----MajorWorry-----
||LineManagerSeen      Name of line manager here
||DateOfLineManagerMeeting dd/Mmm/yyyy
||Date                 dd/Mmm/yyyy
----StartOfTextMajorWorry-----

This form is used to document a major worry that you have about a child, for example after a home
visit. Type your worries here between the two lines in bold that contain dash (-) marks. When you
have a major worry you should talk about it with your line manager or, if they are not available
immediately, the child protection officer. Describe the results of your meeting with them.

You can delete this text and overwrite it. However, please take care not to delete any of the three lines
of the form containing dashes: they are needed for computer processing. There are two dates in this
form: the date you fill it in and the date you met your line manager. Before meeting your line manager
it would be worth partially filling in this form; however, the child's circumstances might be such that
it would need to be delayed. Also fill in the date as a one digit or two digit day, a month using its first
three letters (first letter in caps) and a four digit year. You can separate days, months and years using
the / or - character. Examples are 1/Jun/2008 or 21-Dec-2009.

----EndOfTextMajorWorry-----

```

Figure 2: A word processed 'form' documenting a social worker's worry

The screenshot shows a MediaWiki page for 'Dean Astley'. The main content area contains a form with the following structure:

- A dashed line separator: `----MajorWorry-----`
- A table with two columns:

<code> LineManagerSeen</code>	Name of line manager here
<code> DateOfLineManagerMeeting</code>	dd/Mmm/yyyy
<code> Date</code>	dd/Mmm/yyyy
- Another dashed line separator: `----StartOfTextMajorWorry-----`
- A block of instructional text explaining the form's purpose and usage.
- A final dashed line separator: `----EndOfTextMajorWorry-----`

The page includes a sidebar with navigation links (Main Page, Community portal, etc.), a search box, and a footer with a 'Powered by MediaWiki' logo.

Figure 3: A MediaWiki 'form' documenting a social worker's worry

The other users of this redesigned system would be management at the local children's department and government as part of the statutory reporting process. This would be achieved by creating programs from the skeleton code that produced reports. A simple way of doing this would be to emit a text file with elements separated by commas (CSV format). Such a format is easily processed by standard spreadsheet programs. Typical reports would include those that detail:

- All the cases that have been started but no events have yet occurred for

some time⁴.

- Each child together with a count of each event that has occurred.
- Those children that there are worries about.

The programming required would make use of the variables that are already embedded in the skeleton and would, in the main, consist of Java `print` and `println` statements that deposited the values of these variables into the CSV file; for more complicated reports there would be the occasional `if` statement which, for example, for a report on children with worries would read the boolean variable that denoted that a worry event had occurred.

One potential criticism of course is that a small amount of atomization is evident in the child's chronology by virtue of the fact that the punctuation marks occur; however, there is no reason why a simple program could not be written that removes them and replaces them with simple textual headers for each event.

5 An Appraisal

This section examines the success of the system in terms of the major problems that were encountered with the original ICS in terms of social worker use. It is worth saying, however, that the solution also overcomes the technical problems that have been reported, for example those with undo facilities. This is because it is based on a word-processor and a spreadsheet system which already have the problem functionality successfully embedded in them.

5.1 Narrative vs Reporting

The redesigned system is purely based on a word-processor or *MediaWiki* in which the chronology of interaction is held in a single word-processed document or as a wiki article. A social worker just produces a narrative of all the interactions that they have with a child and others adding to it by instantiating a word-processor template or copying a wiki template and filling it in. However, embedded within each template are as many attributes as are required. The first part of the based narrative is devoted to the event of registering a child after a referral and has necessary bureaucratic contact details which would be implemented as attributes. The remainder of the document would correspond to templates associated with events which have as many or as few attributes that are relevant to the event as is required.

5.2 Sensitivity to Change

The system that I have developed is capable of change which takes cognisance of local circumstances. For example, if a social services department want to use some extra templates all they have to do is to construct them using a word-processor or as wiki `txt` files, change two files and re-generate the XML skeleton

⁴Such occurrences have been the subject of major criticism in a number of inquiries including those that followed the Victoria Climbié and Baby P tragedies.

code. Any report programs generated from the skeleton code that require information from the new templates would need to be modified; fortunately this is a straightforward process involving simple Java statements which issue strings to a file. The same process can be followed for global changes such as government modifications to statutory reporting.

5.3 Generality

The system can be used outside child care. The only reference to the fact that the system is for children's social work is within the Element Descriptors where the elements `ChildSequence` and `Child` are specified and an event such as `CourtOrder` which is associated with seeking a care order are listed as single strings. If the architecture was used for say, probation work, all that would be needed would be to change this file to include elements such as `Offender` and `OffenderSequence` together with events such as `PrisonSentenceReceived`. This would be followed by the development of the word-processor templates that were associated with each event and the report programs.

5.4 Effectiveness of software reuse

The redevelopment of the system so that it matched the DCSF criteria was one of the big successes of the project and one which was not wholly a surprise. The key criticism of social workers was that they were unable to develop a whole picture document. As soon as the decision was made to use a word processor a number of problems were solved: those of a technical nature such as problems with undo and autosave together with the major problem of narrative fragmentation. This had the effect of using implemented functions rather than functions which were developed *ab initio*. I carefully monitored the time that I spent developing the ICS system (around 1300 lines of Java code) and I took just under 15 days to complete this. At a commercial rate for programming of £800 (well above my salary!) this would have cost *c* £12000.

5.5 The use of open-source software

The cost of the system in terms of software is zero. All the major components that were used were free of cost and readily available—many as open-source software. These included: *Open-Office Writer* (the word-processor associated with the *Open-Office* suite), *Open-Office Calc* (the spreadsheet system associated with the *Open-Office* suite), *mySQL*, the *Lucene* search engine, the *MediaWiki* system and a SAX-based API for XML processing.

It is worth saying that the word-processor and spreadsheet that were used could be easily replaced by proprietary software such as *MS Word* and *MS Excel*. The communication between the word processor and the remaining parts of the system is very simple: via `txt` files.

6 Packaging of the Redesign

The architecture that was developed needs no change. However, there is a need for some extra work if it is to be used in a *real* environment:

- CSV files corresponding to reports need to be developed. This would be straightforward in that all that would be needed would, for the most part, be the use of Java `print` and `println` statements together with `open` and `close` file statements.
- The generation of the CSV files needs to be automated, for example by developing a process that is initiated every day which takes these files and constructs the spreadsheets.
- Packaging documentation and user documentation needs to be developed. This would be relatively small: for users it would specify where to find the word-processor templates and instructions on how to fill them in and add them to an individual child record: all standard word-processor activities. Implementation instructions would include: how to install the software on a server and how to provide the templates to users.

All these items are not major. An estimate is that something like c 400 lines of Java and Visual Basic code would be required at most. An estimate of the cost of the industrial packaging is detailed in Table 1. The lineage of code is an estimated upper limit. It is worth making a number of points:

- The costing in Table 1 is approximate just in order to show that a system that cost tens of millions of pounds could be developed for tens of thousands of pounds. What it shows is that by taking care over the human dimension in requirements analysis that it is possible not only to achieve systems that match user needs, but are also considerably cheaper.
- Programmer time is costed at £800 per day. If the development was funded internally in a local authority the staff costs would be reduced by something like 50% to 75%.
- An assumption has been made that programmer productivity would be 40 ELOCs per day. Most of the programming would involve straight line code so it could be argued that this is an underestimate.
- The development of around 30 word-processor or wiki templates would be carried out in the main by social work staff; their time has been costed at £400 per day. The main focus with this task would be the development of guidance text to be embedded in the forms. Much of this text already exists within the DCSF documentation. The figure would need to be adjusted upwards proportionally if more templates were needed.
- The costing of user documentation and implementation directives is based on £600 per day.
- The costing indicates that industrialization of the system would take only a small number of tens of thousands of pounds. This might be seen a huge reduction compared with the £36m that ICS cost the local authorities that used it. What has to be remembered, however, is that each authority was given the option of either buying in a package or developing a system *ab initio*. The true comparison would be with a package that was, say bought by around 20 authorities with the authorities paying around £750,000—still a considerable saving.

<i>Task</i>	<i>Resources</i>	<i>Approx. cost</i>
Generation of CSV files	300 lines Java	£6000
Automation of CSV file generation	100 lines Visual Basic	£2000
Packaging and user documentation	10 days	£6000
Development of word-processor templates	20 days	£8000

Table 1: Costing for the packaging

7 Summary

A major contribution of this article is that it is a case study of the redevelopment of an expensive class of system by considering reuse and that the reuse involved open-source software. The redevelopment effectively ignored a feature of the specification that called for a record-based system and, instead, based it on a word-processor. The software that was used to develop the system was a readily available word processor, a readily available spreadsheet, an open-source XML API, an open-source database system and an open-source wiki API. It is also a lesson to developers—certainly of human-centred systems—not to reach for the UML editor to specify a system *ab initio*, but to pause awhile and cast around for existing software that embeds the functionality required and which is familiar to users *vide* the employment of a word-processor.

References

- [1] The Victoria Climbié Inquiry—Report of an Inquiry by Lord Laming.
- [2] UNISON Memorandum to Lord Laming: Progress Report on Safeguarding. Inquiry Submission, 2008.
- [3] The National Offender Management Information System. Report HC292 2008-09, National Audit Office, 2009.
- [4] BELL, M., SHAW, I., SINCLAIR, I., SOPER, P., AND RAFFERTY, J. An evaluation of the practice, process and consequences of the ICS in councils with social services responsibilities. Report, 2007.
- [5] BROADHUST, K., WASTELL, D., WHITE, S., HALL, C., PECKOVER, S., THOMPSON, K., PITHOUSE, A., AND DAVEY, D. Performing ‘Initial Assessment’ Identifying the Latent Conditions for Error at the Front-Door of Local Authority Children’s Services. *British Journal of Social Work* (2009 Forthcoming).
- [6] HATCHER, E., GOSPODNETIĆ, O., AND MCCANDLESS, M. *Lucene in Action*, 2nd ed. Manning, 2010.
- [7] PECKOVER, S., WHITE, S., AND HALL, C. Making and Managing Electronic Children. *Information, Communication and Society* 11, 3 (2008), 375–394.
- [8] SHAW, I., BELL, M., SINCLAIR, I., SLOPER, P., MITCHELL, W., DYSON, P., CLAYDEN, J., AND RAFFERTY, J. An Exemplary Scheme? An Evalu-

- ation of the Integrated Children's System. *British Journal of Social Work* 39 (2009), 613–626.
- [9] SHAW, I., AND CLAYDEN, J. Technology, Evidence and Professional Practice: Reflections on the Integrated Children's System. *Journal of Children's Services* 4, 4 (2009), 15–27.
- [10] SHAW, I., MORRIS, K., AND EDWARDS, A. Technology, Social Services and Organizational Innovation or How Great Expectations in London and Cardiff are Dashed in Lowestoft and Cymtyrch. *Journal of Social Work Practice* 23, 4 (2009), 383–400.
- [11] WASTELL, D., WHITE, S., BROADHURST, K., HALL, C., AND PECK-OVER, S. The Chiasmus of Design: Paradoxical Outcomes in the E-government Reform of UK Children's Services. In *IFIP Advances in Information and Communication Technology* (2009), G. Dhillon, B. Stahl, and R. Baskerville, Eds., Springer Boston, pp. 257–272.