ISSN 1744-1986

Technical Report N° 2010/ 17

Social Work, Chronicling and the Internet Age

Darrel Ince

19 October, 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

Social Work, Chronicling and the Internet Age

Darrel Ince, Department of Computer Science, Open University

Abstract

Many of the IT systems that are used by social workers are typical of those that were developed before the Internet became a part of everyday life. This article examines a number of developments—in terms of users, technical facilities and environment—that have been enabled by the Internet and describes a system for social work chronicling that reflects them. The context is that of the computer systems used to support the British Integrated Children's System (ICS); this was a set of business processes that were supported by computer programs that reflected an industrial, pre-Internet approach. The article looks briefly at ICS, examines a series of Internet-affected developments and finally describes a chronicling system suitable for any branch of social work that reflects these developments. Although ICS is used as an exemplar in how *not* to develop a modern system, the major points made in this article are relevant to all forms of social work and are also not just relevant to a British context..

1 Introduction—ICS

Since 2006 British social workers have had to cope with the Integrated Childrens System (ICS) [5]; this consisted of a set of inflexible procedures which govern the interaction between a social worker, a child in need, the child's family and other agencies such as the Accident and Emergency Department of a local hospital.

The set of procedures that made up the ICS were supported by a number of computer-based packages that the children's departments of local authorities could purchase. Local authorities could also develop their own software. The then Labour government supplied funding and all the local authorities in England, save one, purchased or developed a system.

ICS attracted a large amount of criticism from social workers, social work managers, academics and unions. The literature that addresses the faults of ICS is comprehensive and it is not the intent of this paper to dwell

too long on the failures of the system—a good overview can be gained from [1], [13] and [15].

The major faults were:

- It concentrated on data gathering, for example for the Children in Need Census that local authorities have to regularly complete. This concentration was at the expense of useful chronicling facilities for the social worker: in effect ICS required the social worker to fill in a large number of forms of which only a small part detailed the interaction with a child in need and others.

- It atomized the chronicle of interaction with a child in need and others. It split up the the various interactions into small discrete chunks. Nowhere was a whole picture created. The effect was to develop something akin to a soap opera where there was little relationship between short episodes and where the episodes were in the wrong order. This is not just a feature of ICS, but has occurred in other areas of assessment [16].

- Social workers reported that the majority of their time was spent in filling in forms rather than carrying out front-line work. Many felt that ICS had reduced their work to that of a data entry clerk [1].

- That it was difficult to share information within teams and between teams and others such as pædiatric staff at a local hospital.

- There were technical problems with the ICS computer systems, for example, social workers complained that they spent hours working on a form only for the server that they were interacting with to crash and lose their work [1]. Many of the technical problems arose from the fact that a heavy-duty industrial system was being run on computer systems which did not have the processor power and storage capacity to cope with the demands of the software: British local government has, traditionally, employed cheaper and older hardware than that found in the commercial arena.

These, then, were the main failures of ICS. Eventually, in 2009, the British government effectively admitted that it had failed and stopped mandating its use. Clearly ICS is a historical entity, however, it does have some value as it represents a technical, managerial and cultural failure that can be used to judge future ICT-based approaches.

This article will not rehearse the relationship between government and the social work profession that has been motivated by targets and was a major influence on both the processes and IT elements of ICS—[6], [7] and [12] are good introductions—but it will have a more technical take. In subsequent sections I shall show that a the Internet and one specific technology—the Wiki—are capable of providing a much richer communicational infrastructure for social work than industrially-informed computer systems that implement ICS and other chronicling systems.

2 IT Development and Structured Data

2.1 Commercial computer systems

One of the characteristics of the ICS is that it was based on an architecture found in a variety of commercial systems such as those implemented for banks, building societies and manufacturing companies. The reason for this is that the specification developed by the then department responsible for children's social work—the Department for Children, Schools and Families (DCSF)—focussed on data collection that could be used for performance management, local auditing and for government-based inspections of children's departments by OFSTED, the Office for Standards in Education, a British government inspection body.

Little attention was directed towards chronicling facilities for front-line workers apart from the insertion of short text boxes within a number of forms known as exemplars. This lead to one of the major complaints about the ICS computer systems: that social workers would spend too much time editing the text to fit the windows-based forms [1].

2.2 Commercial software development

Many of the problems with ICS arise from the database systems that were used: relational databases. In such a database the data is held as tables. An example of a table, taken from a banking application, is shown as Figure 1 (only three records are shown and only a very small subset of data is shown in each record); it holds data on each account held by a customer: their name, their current balance and how much they are allowed to be overdrawn. The key property of relational databases is that they are oriented towards data which has a finite fixed length, for example a surname, the amount of stock in a warehouse, the balance of an account and the number of times a customer has phoned a call centre. They are very poor at storing the variable-length,

Account Name	*Balance*	*Overdraft Limit*
D Roberts	200.34	1000
W Williams	76.22	500
D Roberts	22.01	1400

Figure 1: Bank data in a table

free text required for chronicling. If an application uses a relational database system and requires textual content then the approach normally adopted is to have a finite-length text element in a record; normally only a few hundred characters in length.

Another property of industrial systems using technologies such as relational databases is that they are difficult to change while in operation in order to respond to changing requirements; this is a point expanded on later in Section 3.2.

So, the computer systems supporting ICS atomized text by placing it in tables, ensured that not too much narrative text was stored by the social worker and were structured in such a way that change would be very difficult to achieve. The ultimate change has, of course, happened: the previous Labour government stopped mandating ICS in 2009, leaving local authorities to simplify the way that they carry out children's social work, but with the support of extant, baroque ICS computer packages. This is analogous to a farmer using a Ferrari to tow a truck containing the manure—it works but something simpler and less expensive would do the job.

As a postscript to this section it is worth saying that the developers cannot be blamed for the problems with the ICS IT systems: they were given documentation by the DCSF that effectively told them to develop a system which was similar to that of a data entry system for a bank or some other commercial enterprise; a system which would rarely change and which enforced a large degree of standardization on social workers.

There is a good *technical* reason for standardization: it simplifies the development of an IT system. An example where a system development failed badly with a loss of *c* £600m because of an insufficient attention to standardisation was the NOMIS system for tracking offenders; a major reason for failure was that little effort had been put into simplifying the processes involved in interacting with offenders before implementing the system [2]—the implementation tried to produce a system that covered all local practice. However, there is a potential major tension here between IT systems development and the standardisation that can adversely affect day-

to-day work of a professional, as it did with ICS; this article explores this issue later and provides a solution that embraces a degree of standardisation but also returns useful functionality to the front-line user.

3 The Internet and its Effects

The computer systems that implement ICS are typical of an approach to development and systems architectures that was extant in the eighties and nineties; it was forced on the developers by an over-rigid specification that the DCSF insisted on. The aim of this section is to examine major effects—both in technical terms, user experience and organizational environment—that the Internet has given rise to.

3.1 Technical effects

The first effect that the Internet—or more specifically the World Wide Web—has had is in terms of the nature of the data that it holds. Pre-Internet most data was structured and held on relational databases. Web pages contain large amounts of unstructured text: sentences in some natural language, with only a small amount of structured text being stored. Usually the latter is implemented as forms, for example the form that a user has to fill in when ordering an item from an online retailer.

What this has given rise to is a large body of research into the processing of free text, for example research on retrieving web pages from keywords using search engines, research into the translation of one language into another, research which aims at providing tools and techniques for understanding free text and research into carrying out summary processing in such a way that the key points of the text are presented and unnecessary verbiage eliminated [9].

Moreover this research has given rise to powerful tools and technologies for the processing of natural language. A good example here is that of the programming language PERL that, in a few lines of code, is capable of processing free text that would require many tens, if not hundreds of lines, of conventional coding using industrial programming languages.

A second effect has been the increasing amount of standardization that has occurred because of a huge increase in the transportation of data round the Internet: a process known as electronic data interchange. In order for the data to make sense to any recipient it must be expressed in a standardised form. One of the major technologies for this is the eXtensible Markup Language (XML). This is a language in which the syntactic structure of a

piece of text—small or large—can be defined. There are now thousands of XML-based standards ranging from those in banking to those in education. There are a large number of tools for defining, constructing and editing XML documents.

A consequence of the increasing standardisation is that companies and organisations expose their resources to the users of the Internet in order that systems can be developed that make use of their resources and benefit both the enterprise and those who carry out the development. A good example of this is where the online book company *Amazon* provide a programming interface to the catalogue of items that they sell. Developers can then construct Internet-based programs that, for example, display specialist catalogues on the web and allow browsers of these catalogues to order items; for each item that is bought via the specialist web site the developer is given a proportion of the profit *Amazon* makes.

There are now a large number of enterprises that provide such standard interfaces and there are a huge variety of web sites based on taking advantage of one or more of these interfaces. Where a web site is based on a number of interfaces it is known as a mashup. An example of a mashup is where a developer produces a map of dwellings for sale by accessing facilities at the *Google Maps* web site and the property listings found on the web sites of local property agents.

A third effect is cloud computing. This term describes the fact that data is stored in the Internet rather than on some local computer or a network confined to an organisation. There are a number of advantages to this:

- for a company that offers cloud computing facilities security will be stronger than for some organisation such as a children's department; economic returns to scale ensure this.
- for a customer all the effort required to carry out processes such as backup and those associated with disaster planning are hived off to a third party;
- individual computers that access cloud data require less storage and, as a consequence, will be cheaper.

A fourth effect associated with cloud computing is that of software as a service (SaaS). This is where computer programs are held on servers situated on the Internet with users accessing these programs using a network connection. A good example of SaaS is *Google Docs*, a set of office programs that are similar to the suite of programs provided by *Microsoft Office*. The

word-processor associated with *Google Docs* is not stored on your local computer, but on a remote computer, and any documents that are written are stored remotely.

Google Docs is an example of utility software; there are however, numerous examples of application software being treated in the same way, for example `salesforce.com` is a large company that provides customer relationship management software that, for example, keeps track of sales and leads, with the software being held centrally.

Cloud computing follows the path that electricity generation followed in the nineteenth and early-twentieth centuries in that companies used to have their own generators but eventually power generation was centralized and became a utility [4].

The advantages of SaaS are similar to those of cloud computing as they are close cousins: cheaper local computers; expertise that is often lacking or in short supply within the customer's organisation being able to be provided effectively at the SaaS company because of economies of scale; and the removal of the requirement to update every local computer whenever the software is changed: all that is necessary is to update one copy of the central software.

The final effect is that of the emergence of the plug-in or add-on. Software varies from the specific: for example a system to manage the business of a particular bank, to the general: a word processor or an Internet browser. The users of general software will range in intent and in terms of their computer skills. For example, a social worker may just require basic facilities from a word-processor, while a scientist may require mathematical typesetting facilities over and above the rudimentary ones provided by most word-processors. Because of this, systems developers have developed a number of techniques and technologies that allow software to be bolted on easily, some of these are based on the XML technology detailed above.

Many of the effects detailed above have occurred not because of the Internet *per se*, but because of the availability of high-speed, broadband connections. They form part of the next generation of the World Wide Web known as Web 2.0. The term 'Web 2.0' was invented by the publisher Tim O'Reilly in 2005, for a somewhat technical description of the concept see `http://oreilly.com/web2/archive/what-is-web-20.html`.

3.2 Environmental effects

A major problem with many industrial systems is associated with the process of maintenance. The term 'maintenance' describes the activities that involve

the modification a system after it has been released to its users and is in operation. There are a number of reasons why change occurs in operation. Immediately after a system is released users will discover errors and they need to be fixed; however, one of the major determinants of maintenance change that occurs after this initial teething period is that of requirements change. Here a company might merge with another company, new legislation might mean that a company has to keep extra data or a competitor offers services that a company also wishes to offer and which needs to be supported by a current system.

There have been a number of studies of the amount of effort that is required to carry out maintenance that show it can be considerable, with some researchers estimating that as much as 80% of a systems developer's resources is often spent on modifying a system after release [AuthorOwn2]. The problem is that an IT system will usually consist of tens of thousands, if not hundreds of thousands or millions, of lines of program code, and that one change—even a small one—made to the code can effect the remainder of the system and introduce major errors. This means that when a change is made the developer has to expend considerable resources reading large swathes of program code and then running all the original tests of the system to check that nothing untoward has happened.

Two early pioneering studies of this problem are [8] and [11] which confirm the large amount of resources expended on maintenance; what is noteworthy about these studies is that they were carried out at a time when systems were a small number of thousands of lines of code. Many IT systems now contain hundreds of thousands of lines of code; it is highly likely that the problem has not reduced.

Another environmental effect is that of variability. The more general a software system the more it is required to cope with local conditions. The computer systems supporting ICS were a good example of this. In terms of generality ICS systems should have been somewhere in the middle of the application spectrum bounded by general software such as a word processor and a highly specific software system such as that used to manage a car-hire company: each local authority in England had a version of ICS, but each authority differed in terms of how their children's departments were managed, the client base and the way that local reporting was carried out. Because ICS computer systems were inflexible it took a long time for change to occur and very little variability was achieved.

3.3 User effects

There have been a number of Internet-related effects that have not only affected businesses throughout the world but also the individual user—not just the social worker but their clients. The first effect is that of the greater availability of mobile devices such as the *iPhone* and a variety of phones powered by the operating system known as Android. Such devices not only act as phones but as electronic diaries, mp3 players and web browsers. There are an increasing number of lighter computer-based devices that contain the same facilities; for example the *iPad*—a tablet-like computing device—and a large number of min-laptops known as netbooks. Such devices can access the Internet either via WiFi in buildings or the home or via 3G communications facilities provided by mobile phone companies.

A second effect has been the increase in multi-way communication. Mobile phones provide SMS facilities whereby text messages can be sent from one phone user to another and back. Email carries out the same function but allows for longer messages. Web sites such as *Facebook* and *Twitter* enable Internet users to communicate with each other effortlessly and, in the case of the former instantly.

The one technology that embodies the drive to multi-way communication is the Wiki. This is a web site that acts as a virtual white board; it provides a shared space in which users can collaborate by writing, modifying and deleting text on the white board. The most famous Wiki is *Wikipedia* an online encyclopædia that contains articles that users can create, modify and edit. Although *Wikipedia* is the most well-known application of Wiki technology there are many other applications including collaborative creative writing, developing marketing plans and carrying out collaborative research [14]. It is Wiki technology that lies at the heart of the second part of this article.

4 MediaWiki

MediaWiki is the software system that supports *Wikipedia* [3]. The key element that this system is built around is the article: a collection of text describing a particular topic. It contains a large number of built-in facilities:

- It enables the user to develop text associated with an article and enter it into a text box; the text is then displayed in a web page.
- It enables the user to edit text in an article.

- A variety of media: faxes, images, web links, email links, word-processed documents etc. can be embedded within an article.
- It provides a variety of security mechanisms, for example forbidding other users to change an article that has been created by another user.
- It enables other users to comment on an article written by another user.
- It provides a mechanism whereby plug-ins known as extensions can be developed.
- It has a software interface that allows the developer to develop other types of plug-ins.
- It has extensive facilities for tracking changes to articles.
- It provides automatic indexing facilities that require no work from the user.
- It provides a variety of special web pages useful for the administrator, for example a list of pages that are not referenced by other web pages.
- It enables articles to be given a category which can then be used for searching; for example, the category `Fostered` might be used for a child who is in foster care.

The remainder of this article looks at how Wiki software can be used as an effective plug-in based technology for chronicling—focussing in on children's care. It uses a *Wikipedia* article to hold a chronicle. The eventual system was validated by examining the requirements for a chronicling system that were developed using by carrying a latent error analysis [AuthorOwn1] against the Victoria Climbié Inquiry report [10]. The system was also validated by checking its functions against what were known as the 'business requirements' specified by the DCSF at the beginning of the ICS project. These latter requirements were eventually expanded into a highly constrictive set of requirements that were one of the root causes of the problems with ICS. The system implements all these requirements and, moreover, supplements them by a rich Internet-based communicational infrastructure.

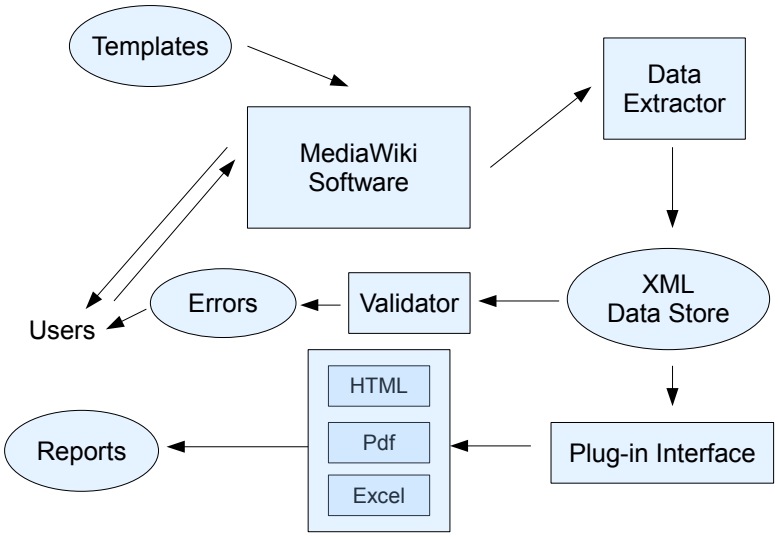

Figure 2: The Architecture of a Wiki-based Chronicling System

5 A Chronicling System

5.1 Architecture

The chronicling system that was developed is based on holding a child record as a *Wikipedia* article—the chronicle. It is worth stressing that this article is not advocating the storage of confidential, child-related data on a public web site run along the lines of *Wikipedia*; rather it is about the use of the Wiki software for running a secure system within a protected network. Each social worker that has responsibility for a number of children is associated with an article, with each social worker article containing hyperlinks to the children they are responsible for. Clicking on a child link will result in the browser displaying a record of events that are associated with that child, for example an initial assessment event. Each event is associated with a sequence of structured data—for example the date of the event—and narrative text that describes the event that can be of any length. The structured data is similar to the data that is conveyed by forms in a conventional system.

A model implementation of a chronicling system was developed. A schematic architecture is shown in Figure 2. There are four components to

this. The first is the *MediaWiki* software as implemented by the Wikipedia Foundation—no changes were made to this component. This provided the vast majority of the functions of the chronicling system.

The second component is a data extractor that takes the structured text associated with an event and then creates a file that relates children's events and the structured data associated with an event. The file contains XML data that describes the structured data typed in by social workers.

The third component is a validator. This checks the structured data in the XML database and issues a number of warnings if there are errors in the data, for example that the name of a child has not been entered or data has been incorrectly entered. The validator is executed periodically checking all the structured data that has been input by users.

The fourth component—or more accurately set of components—is an interface that allows plug-ins to be written to access the XML data so, for example, if a children's department wanted to commission a program that provided certain warnings about the progression of a case then it would be this interface that would interact with these programs. Other typical functions that could be implemented include producing summary spreadsheet information, web pages or printed reports. The plug-in interface has been developed in such a way that it is relatively easy to develop programs that implement such functions. Currently the interface enables reports to be written in the common pdf format and as Microsoft Excel spreadsheets.

5.2 Using the system

Using the system is straightforward. A social worker has to navigate to the web page that holds the name of the children they are responsible for, click on the child whose record is to be added to. They then select a template that describes the event they are documenting, for example a template for a referral (an example of a template can be found in Section 5.4). They then copy the template into the child's record and fill in any structured data; in the case of a referral template who made the referral, their contact details, the reason for the referral etc. The user can then type in any length of free text that they wish to. The *MediaWiki* editor enables the user to type in bullet points (an asterisk precedes the bullet point text) and paragraph breaks (two returns on the keyboard).

5.3 Implementation

A children's department that wanted to use such a system would need to carry out the following tasks if they were to use the system *ab initio* without copying existing records across. First, the *MediaWiki* system needs to be implemented on a web server, this is a technical task which normally takes no more than an hour.

Second, the events that a childrens' department want reported on would need to be defined within the *MediaWiki* database—typical events would include an s47 inquiry, the taking of a child into care, the original referral of a child to a children's department etc.. Many of these would be common across children's social work departments some, however, would be developed based on local circumstances such as a different form of reporting. If the system described here was implemented as a package, common events would come pre-written with a children's department adding any events that reflected local reporting and management practices.

Third, the *MediaWiki* database needs to be populated with the names of the social work staff; third any programs for local and governmental reporting need to be implemented as plug-ins. Most of these programs would be relatively small, for example a program that reported on delays to the progression of a case would be no more than twenty or so lines of program code and would take no more than half a day to program; a program that displayed n data times on a spreadsheet would consist of around $n + 15$ lines of program—trivial by today's software standards. A program that produced a printable pdf version of a a whole corpus of child records with hyperlinked indexing to social workers, children and events such as a referral took the author about 6 hours to develop and test.

5.4 Drawbacks

There are two problems to the system that has been described. The first is that when a user makes an error when entering structured data, for example typing a month greater than 12, then feedback is not instantaneous. The validator program is executed frequently—with the frequency being determined by a parameter in the system. This can vary from a few seconds to a few days although the latter is unlikely. When the validator discovers an error on the article that implements a child record it writes an entry onto an error web page that can be viewed by any users.

This is a relatively minor problem that arises from mixing structured data and free text on the same article. It is a minor problem as there is

no requirement for immediate feedback on an error; chronicling is not like aircraft navigation where invalid data would affect the flight of a plane and would need to be flagged and re-input immediately.

The second problem is that entering structured data is a little awkward. The text for a template of a typical event: that of a referral is shown below. The user copies this from a web page that forms part of the system and then fills it in.

```
{{Referral_Record
| Date_Referral_Event_Entered    =    Text here
| Name_Of_Referee                =    Text Here
| Referee_Address                =    Text Here
| Referee_Phone_Number           =    Text Here
| Date_Of_Referral               =    Text Here
| Method_Of_Referral             =    Text Here
| Reason_For_Referral            =    Text Here
}}
```

The text on the left-hand side of the = symbols is the name of the data item; the user places the text that is required on the right of the = symbol replacing the text 'Text Here'. This is slightly awkward; however, we would contend that it is no more awkward that using complex forms.

6 Wikis and Connection

This section examines the implementation of the chronicling system in terms of the factors that were detailed in Section 3. Figure 3 shows each of the user, development and environment factors associated with the Internet and how the system described in this article implements them. All the factors that were detailed are implemented. We would content that not only does a Wiki approach cover all the functions required of a chronicling system, but also provides a rich environment that is suited both for social workers and their clients.

7 Summary

This article has described an approach to chronicling which deviates greatly from a structured data, forms-based approach. It is based on the use of a web browser and actions that are familiar to any user of such a browser. It

Internet factor	*Implementation*
Mobility	Use of the Internet
Free text processing	The use of a Wiki and standard search engines
Standardisation	The use of XML
Cloud computing	Since it is Internet-based it can be placed in the cloud
Software as a service	Can be placed on the Internet and offered as a service
Plug-ins	Based on accessing an XML database
Maintenance	Use of plug-ins means that extension is easy
Variability	Use of an XML database
Mobile access	Achieved via the Internet
Email	Achieved via the Internet
SMS	Integrated with the use of the web via mobile devices
Social networking	Achieved via the Internet
Multi-way communication	Achieved via the use of a Wiki

Figure 3: Internet factors

has the major advantage that it is capable of coping with both change and with the variability that occurs across children's departments.

One feature of the approach is that, although it has been described in terms of children's social work, it is more general in that it is suitable for any activity where chronicling takes place: for example there are no assumptions that a child is the client. It could be used, for example, within probation work and adult social work: the only relationship that is built in to the system is that a professional is responsible for a collection of clients, each client is associated with a sequence of events (stored as a *Wikipedia* article) that consists of structured data and narrative text and that the sequence of events represents the chronicle of interaction between the professional, the client and others. It is also general in that it an be configured to any set of front-line and managerial practices extant in a social work department.

References

[1] UNISON Memorandum to Lord Laming:Progress Report on Safeguarding. Inquiry Submission, 2008.

[2] The National Offendor Management Information System. Report HC292 2008-09, National Audit Office, 2009.

[3] D.J. Barrett. *MediaWiki*. O'Reilly Media, 2009.

[4] N. Carr. *The Big Switch: Rewiring the World from Edison to Google*. Norton, 2008.

[5] H. Cleaver, S. Walker, J. Scott, D. Cleaver, W. Rose, H. Ward, and A. Pithouse. *The Integrated Children's System: Enhancing Social Work and Inter-agency Practice*. Jessica Kinglsey Publ., 2008.

[6] P. M. Garrett. The Electronic Eye: Emerging Surveillance Practice in Social Work with Children and Families. *European Journal of Social Work*, 7(1):57–71, 2004.

[7] P. M. Garrett. 'Social Work's Electronic Turn'; Notes on the Deployment of Information and Communication Technologies in Social Work with Children and Families. *Critical Social Policy*, 25(4):529–553, 2004.

[8] V.R. Gibson and J.A. Senn. System Structure and Software Maintenance Performance. *Communications of the ACM*, 32(3):347–358, 1989.

[9] D. Jurafsky and J. H. Martin. *Speech and Language Processing: an Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Pearson Education, 2008.

[10] W.H. Laming. *The Victoria Climbié Inquiry*. Dept. of Health, 2003.

[11] B.P. Lientz and E. Swanson. Problems in Application Software Maintenance. *Communications of the ACM*, 24(11):763–769, 1981.

[12] J. Seddon. *Systems Thinking in the Public Sector*. Triarchy Press, 2008.

[13] I. Shaw, M. Bell, I. Sinclair, P. Sloper, W. Mitchell, P Dyson, J. Clayden, and J. Rafferty. An Exemplary Scheme? An Evaluation of the Integrated Children's System. *British Journal of Social Work*, 39:613–626, 2009.

[14] D. Tapscott and A.D. Williams. *Wikinomics*. Atlantic Books, 2008.

[15] D. Wastell, S. White, K. Broadhurst, C. Hall, and S. Peckover. The Chiasmus of Design: Paradoxical Outcomes in the E-government Reform of UK Children's Services. In G. Dhillon, B.C. Stahl, and R. Baskerville, editors, *IFIP Advances in Information and Communication Technology*, pages 257–272. Springer Boston, 2009.

[16] S. White, C. Hall, and S. Peckover. The Descriptive Tyranny of the Common Assessment Framework: Technologies of Categorization and Professional Practice in Child Welfare. *British Journal of Social Work*, 39(7):1197–1217, 2009.