

**Editorial:  
Evaluating library software and its fitness for purpose.**

**Abstract**

<i>Purpose of this paper</i>	To adapt general principles used for evaluating software quality to more specific requirements characteristic of information retrieval and educational applications in library environments.
<i>Design/methodology/approach</i>	A conceptual paper based on existing software evaluation models, but one which tries to add some important refinements to key aspects of these received concepts.
<i>Findings</i>	That the application of engineering concepts to technologically mediated tasks where crucial system outputs consist of the facilitation of specific mental events in the user of the system (e.g. 'learning outcomes') can be problematic. In such scenarios it may be valuable to avoid incorporating the production of specific mental events into the 'objective functionality' of the software in question. Rather, it is suggested that the creation of these outputs should be left to the user, not to the mechanised routines of the system (that is, these mental system outputs should be facilitated by the system as part of the 'subjective functionality' of the overall design).
<i>Research limitations/Implications</i>	These are purely conceptual approaches to library and e-learning software that should be tested by practical case study investigation (e.g. comparative user evaluations both of software designed along general quality models standards, and of software designed to the model suggested here).
<i>Practical implications</i>	You can get better results in terms of human outputs and benefits (i.e. better learning outcomes, or a higher level of engagement with effective information retrieval) by means of simple, minimalist implementations of information technology and educational technology-based systems.
<i>What is original/value of the paper?</i>	This library and information science paper attempts to use ideas from software engineering and software evaluation to create an original perspective on real-life problems in the sphere of information retrieval technology and educational technology.

**Paper type:** Conceptual paper

**Keywords:** Libraries; information services; computer software; evaluation.

## **Introduction**

This paper will attempt to give the practitioner librarian some useful concepts with which to evaluate the effectiveness of library or information retrieval software designed for use in educational contexts, and more broadly, the effectiveness of a range of educational software. These concepts also apply equally well to the creation of effective specifications for writing software for informational or educational purposes. So if you want to know how to create a good spec for authoring your own library package, or how to evaluate such packages prior to purchase, these ideas are intended to help you.

Firstly the concepts are out in the abstract, and they are then followed by a discussion. Finally, some practical examples are given. The initial exposition of these concepts may seem a little detached from the sphere of immediate practice, but hopefully their relevance will become apparent in the latter parts of the paper.

## **Objective and subjective functionality**

One of the core ideas underpinning software evaluation is the concept of functionality. This concept can be defined in a number of ways. Technologists tend to emphasise a view of functionality as a set of properties residing inherently in the technology under consideration. In this approach, functionality is objective and software evaluation consists of seeing whether a package has the right properties 'contained within it' for the task in hand. Indeed the material and straightforward nature of this objective functionality leads some to dismiss any need to think abstractly about functionality. Here is one such dismissive definition of functionality:

'Waffle for "features" or "function". The capabilities or behaviours of a program, part of a program, or system, seen as the sum of its features. Roughly, "the things it can do". Generally used in a comparative sense, e.g. "The latest update adds some useful functionality".' (Howe, 1997)

Other hard, technical views of functionality also emphasise this objective conceptualisation of functionality (Advisory Group on Computer Graphics, 1988 - 1998; DSDM Consortium, 1994-2006 - see comments in the References below).

However, there are other views of functionality which show that it is not in reality a straightforward concept. For example, in a marketing context, functionality has a different emphasis:

'What a product does for the buyer and user; the utility it offers the user; what he or she can do with it.' (Yellow Pencil, 2006)

This definition is quite different from the technologists' objective view of functionality. The subjective element is emphasised in the marketing definition: the function of the product lies in 'What a product does for the...user', not in what features are inherent in the product or engineered device. This is another view of functionality, one which is subjective rather than objective. Note how the nouns in the objective definitions of functionality are all 'things', whereas the nouns emphasised in the subjective definition of functionality tend to be 'people':

<b>Objective Functionality:</b> <i>Emphasis on tool used</i>	<b>versus</b>	<b>Subjective Functionality:</b> <i>Emphasis on User of tool</i>
Hardware		Buyer
Computer package/Software		User
Program/System		He
It		She

Fig. 1: Two types of functionality distinguished.

We need to bear this distinction in mind when turning to evaluations of library software.

### **Hard and soft engineering**

The second type of distinction we should make is between two types of engineering: the 'hard' engineering of tangible products, devices and mechanisms with determinant features (e.g. a car with fast acceleration), as opposed to the 'soft' engineering of virtual creations, whose material existence - as a collection of electrons moving about at the nano-level - is less significant than their conceptual function in the mentality of the user (e.g. an effective piece of e-learning software which aids learning outcomes). This is not in any way an original distinction, and it is implicit in the use of the terms 'hardware' and 'software'. It is simply a distinction which is useful for the applications we are studying in this paper.

One of the characteristics of 'hard' engineering is that, because it involves creating machines for people to use, it must, like any sort of engineering dependent on an operator's involvement, involve the user->interface->tool relationship. For this reason, some of the functionality of 'hard' engineering is subjective as well as objective. In other words, a well engineered aircraft must have a decently designed cockpit with good instruments capable of being used by a skilled operator. Some of the functionality will therefore be defined as 'what the product does for the ... user' - an output of the hard engineering interface design process will be 'that the pilot understands the data given to them by their instruments.' Most of the functionality of an aircraft remains objective - e.g. its speed, fuel consumption, resistance to corrosion and so forth.

Most importantly, it is theoretically possible and practically desirable in hard engineering to eliminate the subjective functionality of the product - the ideal plane is one that does everything right on automatic pilot, so that 'what it does' consists of nothing but hard engineered objective functionality. And this is the crucial difference between hard engineering and soft engineering.

In hard engineering, certain mental events may happen in the mind of the operator, but they are incidental to the objective functionality of the mechanism, and indeed they should be eliminated and absorbed into the machine functionality of the technology. With soft engineering, the events happening in the mind of the user are the explicit aim of the technology and should not be eliminated by the technology's objective functionality.

Thus, in soft engineering, some of the quality of the final product must definitely remain a phenomenon in the mind of the user. If such subjective functionality was to disappear, the essential function of the engineered product would have been lost. This is particularly true of information technology, and even more so, of educational

This is a peer-reviewed author accepted manuscript of the following research article:  
 Joint, N. (2006), "Evaluating library software and its fitness for purpose", *Library Review*, Vol. 55 No. 7, pp. 393-402. <https://doi.org/10.1108/00242530610682119>

technology. If the user of information technology does not become informed then there is no functionality at all, and if the student using educational technology does not become educated, then again, there is no functionality at all – and how! The essential part of this ‘soft’ technology’s functionality therefore resides in the mental events which it must be designed to engender or facilitate.

### **Discussion**

An interesting historical aspect of the relationship between objective and subjective functionality is that, when a technology is primitive, be it soft or hard, then the subjective functionality it engenders can be seen largely or purely as a failure of objective functionality. This is obviously the case with aerospace or mechanical engineering – it took a lot of admirable operator skill to fly a World War One bi-plane, but the requirement for such a high level of skill was hardly a blessing (although the aviation genius Saint-Exupéry, a devotee of subjective functionality, would have disagreed – Chadeau, 2000).

But to some extent this has applied even to information technology and educational technology in the dim and distant past. For example, for all its scriptorial splendour, medieval manuscript parchment was a very primitive form of both information technology and educational technology. It was expensive to make and it had to be written on by hand, meaning it had to be stored very carefully because of its enormous production costs. To make up for its inadequacies, human memory had to be developed to a very high degree – medieval universities to a great extent were places where trainee ecclesiastics and priests were ‘educated’ purely by spending a large part of their ‘education’ memorising holy texts, prayers and liturgical formulae. Education was indeed reduced to little more than spoon-feeding by a lack of technology with enough good objective functionality. However, medieval manuscripts were as well designed as possible in terms of much of their subjective functionality: for example, they were skilfully designed to be much more memorisable than post-Renaissance books and journals (Maharg, 2006)

Thus, the advancement of I and ET which resulted in the ‘cheaper memory’ of the printed book was a great educational step forward. Rather than memorising chunks of text that could not be circulated in sufficient number due to cost, the process of intellectual storage passed from the domain of subjective functionality to that of objective functionality. It became accepted practice not to memorise all the texts that one needed to read, the texts could be allowed to be stored dormant inside the pages of a much more affordable book rather than stored in human memory. The machine took over from the human mind (the printed book being a sort of ‘memory machine’).

And obviously the costs of memory have been falling ever since Gutenberg in the 15<sup>th</sup> century, meaning that as memory is absorbed into the objective functionality of our IT, our information and educational technology can be dedicated to facilitating the right sort of subjective functionality. That is, we have less to remember so that we can think more – or so we hope. Even then there have been opponents of advances in IT who think that indiscriminate progress in information and educational technology which exempts us from the ‘chore’ of memorisation is culturally destructive – these advances abolish the memory of both non-technological cultures as well as important parts of our own (Suaiden, 2003).

This is a peer-reviewed author accepted manuscript of the following research article:  
Joint, N. (2006), "Evaluating library software and its fitness for purpose", *Library Review*, Vol. 55 No. 7, pp. 393-402. <https://doi.org/10.1108/00242530610682119>

So, the impact of the changing relationship between objective and subjective functionality is not a simple one. Although memory storage has become more efficient in the last 500 years (an improvement in objective functionality), other aspects of the transfer of objective functionality to objective functionality have been more problematic. Copying text is one such example.

Throughout history, paper-based copying by students has been laborious, and, in some ways, like memorising medieval text by rote, has simply been a response to a failure to develop helpful IT. Once you can copy easily, by using control-C/control-V, you are free to read copied texts and gain the full benefit of their wisdom, subjecting them to analytic scrutiny without character by character hand reproduction of chunks of text. However, with the distinct gain in functionality of IT-based copying and pasting, something valuable in subjective functionality is lost: the slowness of hand-copying itself allows for scrutiny. Learning takes place by using the act of manual reproduction of an original masterwork as a way of meditating upon the content and wisdom therein. J.S. Bach was famed for his international style of composition, but never travelled: he simply copied original manuscripts (above all the Italian style of Vivaldi's concerti) and learnt in the act of copying – as in fact did many other composer-copyists (Boorman, 2006).

Unfortunately, the learning benefits of copying have largely been lost today, due to the ease with which we can copy. Librarians have long noted the effect of photocopying on student reading, where the machine reproduction of a text seems to absolve the student from the chore of reading the work – library photocopier custodians reflect sadly that, if students had to transcribe bits of their texts by hand, they might take in a bit more of the content. This is at the heart of the contemporary assessment dilemma of plagiarism: a derivative hand written or typed essay in the 1970's might well have been the result of a student reading and reproducing with reverential accuracy a learned essay by an original thinker, and it could have been marked generously on trust. These days the result is more likely to have been an act of plagiarism facilitated by the pull down Edit menu in Word or Internet Explorer (Joint, 2003).

This transfer of subjective functionality (learning something important by copying without IT) to objective functionality (learning nothing by copying easily with IT) is an example of how the intended development of one type of functionality into another is not necessarily an unalloyed benefit. Technological progress is largely a story of advances in hard engineering technology in which products and devices increase in objective functionality while the human does less and less. For example, improved transport technology means that the humans served by it do less ('Let the train take the strain').

The irony of advances in educational technology is that progress has to be measured in different terms. Advances in soft engineering mean that subjective functionality increases: but the humans served by it have to do more ('Now your brain takes the strain!'). The goal of increasing subjective functionality means facilitating an increased level of positive mental events, i.e. those of an educational nature, in the user of the technology – or it should do. Deliberately improving the wrong sort of functionality in a software package can actually be a retrograde step in terms of mental outputs and learning outcomes.

### **Practical applications**

This is a peer-reviewed author accepted manuscript of the following research article:  
Joint, N. (2006), "Evaluating library software and its fitness for purpose", *Library Review*, Vol. 55 No. 7, pp. 393-402. <https://doi.org/10.1108/00242530610682119>

### *Writing information retrieval and educational software*

Good examples of these concepts of functionality presented themselves in embryonic form during the writing of the spec for the GAELS information skills courseware at the Universities of Glasgow and Strathclyde in the late 1990's (GAELS/Glasgow **Allied Electronically with Strathclyde**, 2001). The GAELS e-learning package was designed to help academic staff and researchers learn the electronic information handling skills needed to use the information services of a joint electronic library maintained across both Universities' campuses. But the GAELS e-learning package was in fact the second such package to be developed in the West of Scotland. The preceding TILT information skills courseware development project had developed a similar package focussing on information handling skills *just* at Glasgow University Library (Creanor et al., 1996). This package was extremely well developed and nicely designed, and had a fine pedigree in terms of its user testing and software evaluations.

However the TILT package had not been adopted by librarians at Glasgow University and had deteriorated quite rapidly. This needed explaining lest we were to duplicate the same problems.

One of the characteristics of the spec for the earlier TILT package had been its insistence on a high level of objective functionality in the educational interactivity of the courseware. The developers were very aware of a model of interactivity whereby a process of exposition, activity and feedback on activity, should be "hardwired" into the software. This meant that the package did use highly mechanised multiple choice exercises with quite elaborate feedback menus authored to provide comment on the various choices a student might make when completing the exercises.

The problem for the librarians who needed to inherit and feel ownership of the package was that the mechanics of updating these exercises involved quite a high level of software authoring skills. Moreover, research students themselves found the multiple choice approach a little repetitive and even sterile.

The choice for us was to redefine the functionality we wished to author into the package: we knew that 'interactivity' was important, otherwise the research students using GAELS would remain passive, unengaged observers of textual instructions on how to use information tools. On the other hand the highly mechanised interactivity of the TILT package had not succeeded. We decided to reconceptualise the interactivity of the package as subjective rather than objective functionality. Rather than making the learning events of 'exposition-activity-feedback' into literal events in the software of the courseware, we would let these activities reside more at the mental level of the user than in the overt workings of the package. This also meant the courseware authoring skills needed to maintain the package would be easier to learn by jobbing LIS practitioners, and the package would be easier to take into ongoing library practice.

In fact, the 'exposition-activity-feedback' paradigm (which differentiates educational software from the simpler 'query-answer' model of interaction underlying information retrieval software) did have to be mechanised to a certain degree by some simple javascript routines. These allowed the student to read the exposition on a full screen, while moving through a simple sequence of two further distinct and discrete pop-up screens, one of which set the activity based on the content of the preceding exposition, followed by another pop-up which gave feedback on the likely results of the activity. This required an absolutely minimal level of hard engineering skills. The

This is a peer-reviewed author accepted manuscript of the following research article:  
Joint, N. (2006), "Evaluating library software and its fitness for purpose", *Library Review*, Vol. 55 No. 7, pp. 393-402. <https://doi.org/10.1108/00242530610682119>

larger part of the engineering was 'soft', and resided in the quality of the educational writing skills of the text authors, which created the primarily mental events of successively 'understanding an exposition,' 'doing an activity,' 'receiving and grasping feedback'.

Ironically, practitioners were initially unimpressed by the GAELS courseware and said that it had 'no interactivity' and was therefore 'dull', presenting too much passive text. The previous, highly mechanised routines of the TILT package were looked back on nostalgically as being virtuoso displays of programming skill, with 'lots of interactivity' and therefore 'high levels of functionality' which stimulated lots of activity in the user and gave lots of feedback. This was in spite of the fact that there was no record of any librarian having used the courseware in a class subsequent to the TILT project ending.

It was interesting therefore to use the GAELS courseware with real users, who in fact praised the 'high levels of interactivity' of the package. Intuitively, the students realised they were being stimulated to high levels of educational interactivity by the simply engineered exercises of the package, despite the fact that the programming of GAELS was much less busy than its predecessor. How can a package that does less stimulate more activity on the part of the user? If you accept the marketing definition of functionality (emphasising 'What a product does for the ... user'), then the level of functionality in the GAELS courseware was high. If you wanted a high level of mechanical functionality, then the package failed. But educational technology aims at provoking mental events by means of physical activities (electrons passing through a pc), and better mental events are not necessarily linked to high levels of mechanical activity.

Can these concepts of functionality be worked into existing patterns of software evaluation? If we turn to the British/International Standard for software evaluation, BS ISO/IEC 9126 (BSI, 2001), it provides a model of software quality which embraces a number of factors, which in turn can be broken down into more specific quality attributes. These top-level factors are as follows:

- Functionality
- Reliability
- Usability
- Efficiency
- Maintainability
- Portability

These factors do describe quite usefully some of the quality issues we were faced with in evaluating existing courseware and re-specifying new courseware in GAELS. In trying to increase the 'functionality' of the TILT courseware by over-mechanising it, the 'maintainability' of the package had suffered. By rethinking what was meant by 'functionality', we managed to increase the genuine functionality of the package while at the same time increasing its maintainability – the GAELS courseware, in contrast to the TILT package, was adopted by library practitioners who managed to master the scripting routines underpinning its interactivity and maintain its content and formal language (e.g. adapting it from html 4.0 to xhtml 1.0). It is still in use by practitioners to this day.

#### *Evaluating off the shelf packages*

This is a peer-reviewed author accepted manuscript of the following research article:  
Joint, N. (2006), "Evaluating library software and its fitness for purpose", *Library Review*, Vol. 55 No. 7, pp. 393-402. <https://doi.org/10.1108/00242530610682119>

This may be fine as an approach to authoring such packages, but how does it work in practice when the library practitioner has to choose and evaluate different types of already designed, of the shelf library and information software?

One of the present problems confronting LIS practitioners is choosing packages for a range of new purposes. The creation of digital libraries involves choosing and using a variety of possible digital library packages. But, depending on the type of digital collection you are creating, you may need one type of tool and not another. These tools are in many cases so unfamiliar that the names of the software may not mean very much, let alone the descriptions of what the software is supposed to do.

But if you were to deduce from the GAELS courseware example above that the guiding principles of our approach can be summed up by phrases such as 'low-tech isn't no tech', and 'less is more', then you'd not be far wrong.

For example, there are a range of 'digital asset management' packages on the library software market at the moment, which can be used for a range of purposes: you can use them to create learning object repositories, or institutional research repositories, or even traditional catalogues. One helpful idea to bear in mind when looking at these products is to ask: what sort of functionality does the task to be fulfilled actually require?

In other words, are the essential outputs of the system a type of mental event or a mechanical process? If the essential outputs are mechanical processes then you need to have this functionality hard-wired into the system as well as possible. If the outputs are mental events, then it may not be necessary for the system to generate these outputs mechanically (remember the over-mechanised interactive routines of the TILT package?) In fact a mechanical approach to these mental outputs may undermine the effectiveness of the package, if it is an educational package or even an information retrieval system designed for an educational environment (all such packages trigger mental events).

It is interesting to note that the self-same digital bibliographic item may very appropriately be found in a range of digital packages: in a learning object repository, in a virtual learning environment (VLE), in an institutional research repository and in a library catalogue (for example, an academic research paper can be catalogued in an opac, used to teach in a VLE and be repackageable via a learning object repository, and if it is a significant piece of research, it can also be lodged in an institutional research repository). The main difference between the function of the item in its four different homes, is conceptual rather than mechanical: the researcher searches an institutional research repository for an item to stimulate their original research ideas, the student searches for a learning object to understand a received piece of accepted knowledge.

So the same 'knowledge object' remains technically the same entity in both contexts (it's the same bunch of bits and bytes) but conceptually it fulfils two entirely different functions: therefore the most important particular functionalities of learning object repositories and institutional research repositories are in fact their highly differentiated types of subjective functionality. The essential (but different) purpose of each these packages is to facilitate certain mental understandings of the use of the item of the knowledge objects which they store.

This is a peer-reviewed author accepted manuscript of the following research article:  
Joint, N. (2006), "Evaluating library software and its fitness for purpose", *Library Review*, Vol. 55 No. 7, pp. 393-402. <https://doi.org/10.1108/00242530610682119>



Of course, in some cases the understanding with which each knowledge object is used may have such legal significance that you don't want to leave the matter to chance; for example, digital learning objects in VLEs can be shared legally only amongst the students in the relevant class, while institutional research objects in institutional research repositories can only be used for the personal research and study of any individual researcher.

You may wish to have this distinction hard-wired into your software as objective, mechanised functionality, that is, by means of password control or authentication. However, if these information tools are to be used as educational technology (because IT and ET can also imply different types of functionalities!) then you may wish to educate the user in digital rights issues by simply telling them on a technically less elaborate help-screen what is the correct usage for the object in question in the given context. If the power of machine authentication controls the whole process then there is no danger of the student making a mistake, but there is also no need for the student to understand what they are doing or learn about intellectual property issues. (Educators and digital rights holders will probably have different views on this!)

So in many cases, you can get better informational and educational outcomes with simpler technological approaches. Of course, the difference between objective functionality and subjective functionality echoes other issues in web theory and thinking, such as the emergence of the idea of the semantic web from the purely technical network that comprises the 'non-semantic' web, and also the difference between technical interoperability and semantic interoperability (e.g. Miller, 2001). All such concepts fundamentally differentiate the technical existence of networked information from its semantic or mental dimension, and are aware of the vital need to distinguish between the utility of the internet as a technical medium and its cognitive, human dimensions, wherein lie its real value.

Library and information practitioners have to tap into this line of thinking as well, simply in order to do their job properly. These concepts provide them with ways of evaluating the true fitness for purpose of educational library software, and help distinguish between appropriate functionality, and over-elaborate or misapplied machine functionality. This can be unnecessarily expensive to purchase and burdensome to maintain.

So the pragmatic traditions of our profession, with its emphasis on the user-friendly rather than the intimidating, should prove a suitable and effective medium for the implementation of such ideas. We trust that our readers will use them in that way!

Nicholas Joint  
Editor  
'Library Review'

## References

Advisory Group on Computer Graphics, (1988 - 1998) "AGOCG Activities and Resources" < <http://www.agocg.ac.uk/cdroot.htm> > [Their definition of functionality:]

'A measure of the capability of, or range of facilities available in, an item of software or hardware. Modern computer packages and equipment tend to have much better functionality than older ones, and expensive ones greater functionality than cheaper ones.'

Boorman, S. "Holograph", IN: Grove Music Online ed. L. Macy (Accessed 5 May 2006), < <http://www.grovemusic.com> >

'The circle of scribes who worked for major 19th-century composers frequently included other, younger composers, earning their living while learning their craft. The musical style of the younger man often shows traces of what he learnt while working as a copyist.'

BSI, British Standards Institution (2001). "BS ISO/IEC 9126-1:2001 Information technology. Software product quality. Quality model (PD ISO/IEC TR 9126-2:2003 Software engineering. Product quality. External metrics PD ISO/IEC TR 9126-3:2003 Software engineering. Product quality. Internal metrics PD ISO/IEC TR 9126-4:2004 Software engineering. Product quality. Quality in use metrics)"

Chadeau, E. (2000) "Saint-Exupéry". Paris: Perrin. [Novelist and aviator Antoine de Saint-Exupéry knew that the subjective functionality of flying - personal skills and artistry - was lost by advances in the objective functionality of aircraft technology: in fact such learned artistry was never designed in as part of the system outputs of aircraft engineering and cannot be called 'functionality' as such. Yet he himself still managed to be an energetic proponent of hard engineering in this field and helped design out these incidental but desirable learning outcomes from the process of flying! Some of his 'hard engineered' patents are listed at the url below:  
< [http://www.saint-exupery.org/biographie/inventeur/inv\\_brevet1.htm](http://www.saint-exupery.org/biographie/inventeur/inv_brevet1.htm) >]

Creanor L, Durndell H, Primrose C. (1996) "Library and Study Skills using Hypertext: the TILT experience" The New Review of Hypermedia and Multimedia, Applications and Research, Vol. 2, pp121-147.

DSDM Consortium, (1994-2006) "Dynamic Systems Development Method Version 3" < <http://htsa.ie.hva.nl/~helpdesk/DSDMManual/default.htm> >.  
[Their definition of functionality:]

'What the computerised part of a system does to automate the business processes'

GAELS (2001) GAELS Project report.  
< <http://gaels.lib.gla.ac.uk/gaelsprojectreport.pdf> >

This is a peer-reviewed author accepted manuscript of the following research article:  
Joint, N. (2006), "Evaluating library software and its fitness for purpose", Library Review, Vol. 55 No. 7, pp. 393-402. <https://doi.org/10.1108/00242530610682119>

Howe, D. (1997) "DefineThat". < <http://www.definethat.com/default.asp> >

Joint, N. (2003) Editorial ["Plagiarism, copying and learning"], *Library Review* 52 No. 2 pp. 53-55.

Maharg, P. (2006) "'Borne back ceaselessly into the past': Glossa, hypertext and the future of legal education". Draft paper presented to the British and Irish Law, Education and Technology Association (BILETA) Conference, 2006 (unpublished).

Miller, P. (2000) "Interoperability. What is it and Why should I want it?" *Ariadne* 24 < <http://www.ariadne.ac.uk/issue24/interoperability/intro.html> >

Suaiden, E. J. "The social impact of public libraries", *Library Review* 52 no. 8 , pp. 379 – 387.

Yellow Pencil (2006)

< <http://www.yellowpencil.co.nz/brand%20sharpening/brand%20glossary> >

(N. B. all urls accessed May 10<sup>th</sup> 2006)