Working Paper Series
ISSN 1170-487X

COMPUTATIONAL SENSE: THE ROLE OF TECHNOLOGY IN THE EDUCATION OF DIGITAL LIBRARIANS

Michael B. Twidale and David M. Nichols

Working Paper: 10/2006
October 2006

Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton, New Zealand

Computational Sense: the role of technology in the education of digital librarians

Michael B. Twidale[1] & David M. Nichols[2]

[1] Graduate School of Library and Information Science
University of Illinois at Urbana-Champaign, IL, USA
twidale@uiuc.edu

[2] Department of Computer Science, University of Waikato
Hamilton, New Zealand
dmn@cs.waikato.ac.nz

INTRODUCTION

The rapid progress of digital library technology from research to implementation has created a force for change in the curricula of library schools. The education of future librarians has always had to adapt to new technologies but the pace, complexity and implications of digital libraries pose considerable challenges. In this article we explore how we might successfully blend elements of computer science and library science to produce effective educational experiences for the digital librarians of tomorrow. We first outline the background to current digital librarian education and then propose the concept of *computational sense* as an appropriate meeting point for these two disciplines.

DIGITAL LIBRARIANS

There is an ongoing discussion on what it means to be a digital librarian (Coleman, 2005; Marion, 2001; Mostafa et al, 2006; Pomerantz et al, 2006). It is similar to being a traditional librarian in terms of ethos and applicability of core guiding theories, including access, cataloguing, collection development and teaching people both search skills and general information literacy. It can be viewed as simply integrating one set of digital information resources into a set of access services provided around a pre-existing set of paper-based resources. However it also requires additional technical skills, extending the librarian's role into new areas. For example, DL creation can be more akin to publishing than collection development, involving aspects of editing, revision and aligning to data and metadata standards.

The topic of digital librarianship is subject to rapid change (Pomerantz et al, 2006), as it is partially defined by the availability and functionality of appropriate software. The ease of building a digital library or an institutional repository has been radically changed over the past decade by the emergence of software such as Greenstone (Witten & Bainbridge, 2003) and DSpace (Tansley et al, 2005). For example, the courses presented in library schools are often based around similar material presented in tutorials at digital library conferences. Education based around digital library software inevitably brings with it techniques and concepts from computer science; especially as most of these courses combine theory and practice (Ma, Clegg, & O'Brien, 2006) Issues that can cause problems for library students vary from the basic (such as selecting the appropriate software version, downloading and installing) to advanced topics involving customization and extensibility (Nichols et al, 2006).

COMPUTATIONAL SENSE

As libraries have become more computerised, librarians have had to learn more and more about computer systems, file formats and web servers. Over time, commodity software can hide away certain levels of complexity as a basic infrastructure is developed. But innovative computer applications will always require a degree of explicit support and tinkering. For example, the Greenstone software development team has, over several years, developed a variety of tools to abstract away from technical details to simplify the tasks of creating and maintaining digital collections. Despite this work the experiences of using digital library software can still be disconcerting for some students (Nichols et al, 2006). This state of affairs is more a reflection on the state-of-the-art in content management systems than a criticism of any particular software applications.

From the perspective of software developers the power of digital library software derives from the flexibility of computer programming languages. Indeed, the Greenstone digital library suite embeds programming language constructs into its customization features; providing considerable flexibility for collection design but at the expense of learnability (Nichols et al, 2005). A practical example is the nature of conditional statements in Greenstone macros, which:

- have a unique syntax (rather than re-using an existing language such as JavaScript or PHP)

- have a different syntax to conditional statements in Greenstone formatting statements
- have no error checking at design time (such as the red underlining of spell checkers or Visual Studio)
- have no integrated documentation (such as found in modern programming development environments)

At one level it is simple to say that these can be corrected through more software development. The larger question is whether this is the right direction for systems that are to be used in digital librarian education. The resulting system would be very similar to an IDE (Integrated Development Environment) of the sort regularly used by computer science students and programmers the world over.

This raises the question: should future librarians be forced to become programmers in order to graduate? We believe that digital librarians should not have to become programmers but that they *do* need to acquire a fluency with information technology beyond that of a traditional MLS degree. We coin the term *computational sense* to describe this level of fluency. It covers a range of issues that remain in flux and are clearly open for debate. We outline a few of them below for illustration.

Comfort and fluency with computational systems

Of course a digital librarian should be comfortable in using the latest technologies available. However, although necessary we do not believe this is sufficient for librarians to be able to take active roles in exploiting the potential of a rapidly changing set of technologies.

Metacognitive skills in learning about new computational resources

Possessing training in a technology is insufficient because technologies change so rapidly. New versions of applications contain new functionalities, new interfaces and hence new possibilities for how they might be used. New applications become available to complement or replace existing applications. Online resources and computational services can enhance or disrupt what users expect or want to be able to do with existing resources. Patrons' changing reactions to library catalogues in the light of their experiences with Amazon and Google are just one indicator of this. Consequently, practising librarians need to be able to learn about the latest versions, applications and new ways of combining applications quickly and maybe without formal training.

This can be done, but probably requires the teaching of the metacognitive skills that can facilitate more efficient and effective learning both by individuals and communities within and across organisations.

Fluency in incremental tailoring and combining of applications for evolving needs

Even if they do not directly program new digital library applications, librarians are likely to be involved directly or indirectly in the installation and use of those applications, which invariably involves tailoring work. Modern applications typically come with a host of options, and the defaults are often less than ideal. Making a resource useful and usable in a particular context typically involves a range of tailoring activities ranging from selecting between options, choosing modules to install, including or linking the application to the resources it will use or provide, integrating the software with other applications or provisions and 'skinning' the interface to create an integrated, consistent and coherent look and feel for end users as part of a focus on usability. All these require varying levels of technical expertise. They also require the ability to interact productively with others, both stakeholders with far less technical skill, and also technical experts, lacking local knowledge of use-in-context. Although somewhat daunting-sounding this information intermediary role is one firmly within the bounds of the ethos of traditional librarianship (Erlich and Cash, 1994).

A sense of applications as ongoing co-designed artefacts rather than technological givens

Computational sense should include an understanding and an expectation that digital librarians be involved not just in the selection, analysis, and facilitation of learning and use of computational artefacts, but that they should also be involved in the *design* of those artefacts. At the very least it should empower digital librarians to take a more direct role in larger-scale design processes. Applications should not be taken as technological givens, to be coped with by them and in their information intermediary role in helping others to learn and use. Rather these applications need to be regarded as provisional, changing and amenable to change by librarians. In many existing cases this still might not actually be feasible, but we believe that it is worth considering what it would take to give more librarians the skills necessary to not merely critique inadequate systems but to get involved in actually improving them. Design does not necessarily require programming. As noted above, tailoring is one aspect of design, but there are others. As

well as in-house design, there are various approaches to wider involvement in larger co-design processes, including participatory design (Kyng and Mathiassen, 1997) and the involvement of lead users (von Hippel, 2005) as well as open source software development. Modern systems are frequently in the process of near-continual re-design as new versions and modules are released, which at least raises the possibility of greater involvement by people with appropriate skills

A sense of the feasibility of potential design options

To be fully involved in ongoing co-design, it is very helpful to have a good sense of what is feasible in current systems development and what is not. A qualitative sense of the relative costs of different design options is also helpful as the design space is collectively explored. Typically computer science students are expected to acquire these rich qualitative skills just by lots of practice. They are rarely explicitly taught. We speculate that it might be possible to teach these skills of judgement without requiring considerable practical programming experience. If so, it would allow digital librarians to engage in far more productive interactions both with vendors and systems development teams, as advocates for stakeholders and use-in-context as different functionalities and possibilities are uncovered. If you don't have a sense of what it is reasonable to ask for, you are inclined to select just from the options on offer. All design involves trade-offs and compromises, but when these are done in ignorance of whole categories of costs, benefits, and opportunities, truly innovative design possibilities are overlooked.

Spreadsheets as a guiding metaphor

To illustrate our vision, we use an example from an earlier round of technological innovation coinciding with the advent of the personal computer in the 1980s. Prior to that time, people who wanted to do mathematical calculations on computers (including scientists and financial experts) would have had to write a program in a language such as FORTRAN, or commission someone else to do it for them. This was for many daunting or prohibitively expensive in terms of money, opportunity cost or effort-risk-reward calculations. With the advent of the early spreadsheet programs on PCs, many people started using them because they were far easier to both learn and use than programming to achieve basic numerical calculations. Although intended for accounting calculations, the ease of learning and use encouraged innovation and appropriation by many others with needs to manipulate numbers. Nevertheless, spreadsheets were not completely trivial to use. One needed basic numeracy to be able to use them productively and

not generate nonsense – what we might call a numerical sense. One just did not need to know how to program in FORTRAN. Ability to program certainly helped in avoiding certain classic errors, and was essential for more complex calculations that either the spreadsheet software of the day could not manage or where the programming of macros was needed. But programming ceased to be a prerequisite and the skills of sophisticated spreadsheet use, planning, design and debugging could be taught to people who had not first learned to program.

In a similar manner, we believe that it is possible to extract some of the skills traditionally acquired (explicitly but more often implicitly) from long experience of programming, and teach these to librarians under the label of computational sense. In so doing we can widen the number of people able to productively engage with information technologies, just as spreadsheets widened the use of computers by numerate people far beyond the range of those willing to invest time in learning FORTRAN. This approach also challenges the developers of digital library software to produce tools as useful and useable as the spreadsheet.

CONCLUSION

Computational sense extends beyond the notion of programming to encompass an ability to understand the broader notions of the capabilities of software and the socio-technical issues of usability, system deployment and maintenance. In this article we have proposed some characteristics of computational sense and how they might influence the design of the curriculum for digital librarians. In doing so we don't claim to have a complete list but simply to contribute a new perspective to the ongoing debate on the nature of digital librarianship.

REFERENCES

Coleman, A. (2005) Interdisciplinarity, interactivity, and interoperability for educating the digerati. *Education for Information* 23(4) 233-243.

Erlich, K. and Cash, D. (1994) Turning information into knowledge: information finding as a collaborative activity. *Proceedings of Digital Libraries '94*. College Station, TX. 119-125.

von Hippel, E. (2005) *Democratizing Innovation.* Cambridge, MA: MIT Press..

Kyng, M. and Mathiassen, L. (1997) (eds.) *Computers and Design in Context*. Cambridge, MA.: MIT Press.

Ma, Y., Clegg, W., and O'Brien, A. (2006) Digital library education: the current status. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'06)*. ACM Press, New York, NY, 165-174.

Marion, L. (2001). Digital librarian, cybrarian, or librarian with specialized skills: Who will staff digital libraries? In H. Thompson (Ed.), *Crossing the Divide: Proceedings of the Tenth*

National Conference of the Association of College and Research Libraries. Chicago, IL: American Library Association. 143-149.

Mostafa, J., Brancolini, K., Smith, L.C. and Mischo, W. (2005) Developing a digital library education program. *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'05)*, 427. ACM Press.

Nichols, D. M., Bainbridge, D., Downie, J. S., and Twidale, M. B. (2006). Learning by building digital libraries. *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'06).* ACM Press, New York, NY, 185-186

Nichols, D. M., Bainbridge, D., Marsden, G., Patel, D., Cunningham, S. J., Thompson, J., Boddie, S. J. and Witten, I. H. (2005) Evolving tool support for digital librarians. Theng, Y.-L. and Foo, S. (eds), In *Design and Usability of Digital Libraries: Case Studies in the Asia Pacific*, 171-189. Information Science Publishing, London.

Pomerantz, J., Wildemuth, B. M., Yang, S., and Fox, E. A. (2006) Curriculum development for digital libraries. *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'06).* ACM Press, New York, NY, 175-184.

Tansley, R., Smith, M. and Walker, J.H. (2005) The DSpace open source digital asset management system: challenges and opportunities. *Proceedings of the 9th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2005)*. LNCS 3652. Springer.242-253.

Witten, I.H. and Bainbridge, D. (2003). *How to Build a Digital Library*. San Francisco, CA: Morgan Kaufmann.