The myth of the reusable index

Bill Johncocks

Index reusability is one of the claims made for embedded indexing but the technique itself provides evidence that no human index can ever be divorced from its source document.

Embedded indexing is still a minority enthusiasm among publishers, but indexers who regularly practise embedding may have been struck by a strange irony. One of the claims made for the technique is that it allows an index to be reused without further human intervention even after changes to the document, yet the mechanics of embedding themselves demonstrate that index reusability is completely unachievable. This has implications well beyond the narrow world of embedding: not only should the non-portability of an index hold true however it was made, but misunderstandings about reusability illustrate where our relationships with publishers and systems people have gone wrong. Understanding this might help us resist detrimental changes wished on us in future.

A brief outline of embedded indexing

The technique needs a machine-readable document, which is no real limitation now even parish magazines are drafted on PCs, so we are never likely to meet new material without a machine-readable equivalent. The indexer inserts each term beside the text to which it relates as part of a hidden text tag, instead of building a separate index file with terms accompanied by their locators. Then, once the indexed document has been laid out and paginated, the software automatically associates locators with those terms and generates a sorted index. As an example, here is the start of the astronomical song from *Monty Python's Meaning of Life*, together with some Microsoft® Word embedding:

> Just remember that you're standing on a planet that's evolving
> And revolving at nine hundred miles an hour {XE "Earth:rotational velocity"}
> That's orbiting at nineteen miles a second{XE "Earth:orbital velocity"}, so it's reckoned,
> A sun that is the source of all our power.
> The sun and you and me and all the stars that we can see,
> Are moving at a million miles a day.
> In an outer spiral arm at forty thousand miles an hour,
> Of the galaxy we call the Milky Way {XE "galaxies:*See also* Milky Way;!" /t}{XE "Milky Way:Sun's position and motion within"}.

and here are the index entries it might lead to:

> Earth
>   orbital velocity, 1
>   rotational velocity, 1

> galaxies
>   *See also* Milky Way
> Milky Way
>   Sun's position and motion within, 1

The example includes operators for subentries (:), forced alphabetization (;) and cross-references with suppressed locators (/t). Other word processing and desktop publishing packages have different ways of coding these essential operations, but the embedding is usually as hidden text. So, in Word, just typing {XE " will not result in a usable embedded entry: hidden text insertion is a system function. Indeed, turning off the 'Show hidden text' option will make all these entries disappear.

Claims of reusability

Embedded indexing offers some genuine advantages over conventional, stand-alone working, although unfortunately its proponents claim much more. The principal advantages claimed are time savings, resilience and reusability.

Time savings are achievable because each text section can be indexed as soon as it is stable, rather than only after final pagination. Instead of having to wait until the last chapter of a 20-chapter book is typeset, then sending the entire text off for indexing, a publisher can have each chapter indexed once the text is finalized so that, when the last chapter does arrive, the rest of the book is already indexed. As we shall see later, such time savings are partially time *shifts*, but the advantage, in an industry where schedules are being squeezed, is genuine enough.

Resilience means that any deletions and rearrangements of the indexed text will still result in the right locators being allocated. They are only associated with the entries at the time of index generation, so if the text changes, the software is simply run again. So repagination of a UK hardback as a paperback, or with US page sizes, or deletion of numbered pages of illustration, will not necessitate re-indexing. And rearranging chapters or deleting an indexed section will usually work, although there are dangers.

Reusability is a bolder claim. Because embedded index entries move with the text to which they were allocated, repackaging a document – as a web page, a briefer extract or part of a larger compilation – should in principle still yield a coherent index automatically, without further human involvement. This is the myth I want to explode. And it is one to which we have ourselves contributed, uncritically supporting systems specialists' sales talk. For example (Mauer, 2001):

Future revisions of the text will not require re-indexing of the entire text since index entries will be retained in the text that remains unchanged . . .

> If a university press wants to create a textbook that just includes chapters 1, 3, 5, 7, and 9, the corresponding index can be generated quickly

It's an attractive idea: index once – publish many times. But I don't believe these claims are sustainable. In fact, I think that by accepting them, we are contributing to the under-valuing of our profession. What is generated under these scenarios may well look like a convincing index, but it will contain deficiencies in proportion to the extent of the changes made.

Simple and strict embedding

One last embedding technicality: in my SI workshop, I distinguish two variants of embedding which (never having seen them discussed before) I call 'simple' and 'strict' embedding. Crucial in the real world, these are practical alternatives adopted after the terms have been embedded and a draft index generated for editing by the indexer.

In *simple embedding*, editorial changes are made to the generated index but not to the embedded mark-up, so that the document itself remains uncorrected. The corrected index is then typeset and added to the source document just like a stand-alone index file derived from MACREX or CINDEX.

In *strict embedding*, changes are made more laboriously by modifying the original embedding: a second run of the program then generates a corrected index and leaves a document that incorporates the indexer's changes.

Simple embedding tends to be chosen when saving time is all-important, and leaves a document whose embedding is not reusable. Strict embedding is preferred when the aim is reusability of the indexed text, although at the cost of an additional indexing stage.

The need for editing

The index generated automatically from embedding can never be anything like satisfactory without further work. Two major problems compromise its quality:

- Indexers cannot see the index developing as they add terms.
- To save time, they are often expected to work without a complete view of the source document.

So consistency becomes elusive even if one has a photographic memory. In practice, material indexed early in a project (the first chapters of a book, say) is often returned to the client and thus unavailable to the indexer for later comparison, while later material may not have been released. The inadequate index first generated will need thorough editing, cancelling out some (though not all) of the claimed time savings. Of course indexes always need editing, but with conventional, stand-alone indexing, one's first thoughts can be recalled, reconsidered and if necessary

amended while new index entries are being made. It is a peculiarity of embedding that corrections must be held over until the first index has been generated, then carried out as a distinct stage. That at least allows embedded indexers to imagine the effects of their changes on reusability.

Components of the editing stage

While editing indexes resulting from initial embedding, an uncomfortable conviction forced itself on me: the changes I made to match the index to its parent document were all progressively undermining its portability. A few examples of conventional indexing practice should convince any sceptics. If we can agree that the editing of a draft index comprises the following stages:

- imposing consistency
- ensuring accuracy and 'disambiguation'
- optimizing the number of subheadings and locators
- choosing the direction of cross-references
- choosing between cross-references and double posting

let us look at each in turn and assess its effect on reusability, beginning with the most important: imposing consistency.

A draft index (given only normal inconsistency by author and indexer) might contain, well separated by the alphabetic sequence and thus not always apparent, such sequences as:

> accidents, motor vehicle, 26
> motor vehicle accidents, 57, 90
> road accidents, 301-304, 310
> traffic accidents. See road accidents

In a stand-alone index, we might simply have corrected these in passing.

Making this index consistent now clearly requires document-specific decisions. One concept is being expressed in four different ways, but which term is preferred and which are relegated to cross-references depends on factors like usage elsewhere, probable reader expectations and perhaps the indexer's own prejudice. There is no reason why these unavoidable decisions should be reproducible. In a different book, the choice might be very different.

The next editing operation is to provide 'disambiguation', for example:

> mobiles (sculptures), 57
> mobiles (telephones), 12, 168

and to ensure accuracy:

> Millenium bug, 18
> millennium bug, 127
> Millennium bug, 210–230, 301
> Y2K problem, 82

But we would only qualify 'mobiles' in documents where the term was used in both senses and there was a genuine possibility of confusion. The second example reminds us that

there is a time component of reusability too. Though I don't intend to discuss it further, many of us recall a happy time when few Britons could even spell 'millennium', and perhaps look forward to one when nobody even remembers what Y2K once signified. All indexes eventually date (though less rapidly for guides to English cathedrals, say, than those about the damp squib that was 1 January 2000) and this limits their reusability too.

The final stage of editing is almost cosmetic: optimizing the number of subheadings looks for a happy mean between superfluous subentries:

> golden eagles
>   persecution by gamekeepers, 25

and long strings of undifferentiated locators:

> golden eagles, 25, 76–79, 83, 103, 105, 116–119, 123, 134, 149–152, 170 . . .

But again the balance we seek is right only for the document actually being indexed: it depends entirely on the total number of significant mentions. Finally we need to decide on the direction of *see* cross-references; whether *see also* references should be reciprocal, and between cross-referencing and double posting before finding and extirpating the feared blind links and circular references. All these decisions are document-specific. If they were not, perhaps we would not have to repeat them every time.

To summarize, editing manifestly reduces portability; its purpose is to make an index consistent, usable and attractive within its present setting. But an unedited index is not the answer. That would just be chaotic without being more portable: a ragbag of uncontrolled and unrelated synonyms.

The context-sensitivity of all indexing

If reusability already looks shaky, things are about to get worse. Because is the initial selection of index terms not also sensitive to context? It certainly takes account of the overall document subject, reflects the anticipated readership and any treatments elsewhere, and involves assessments of informativeness and uniqueness. It is also subject to stylistic variations between indexers. All these influences either depend directly on the physical form of the document (book, conference proceedings, website, periodical article or whatever) or else their effects are unpredictable if reuse is attempted. Let us again look at each operation in turn.

Imagine we have a paragraph pointing out that road accidents caused by kangaroos are increasing significantly and giving detailed annual figures. In isolation – perhaps also in a general periodical – we might make three entries to cover this:

> kangaroos
>   road accidents involving, 25
> road accidents
>   numbers involving kangaroos, 25
> statistics
>   road accidents involving kangaroos, 25

but this would seldom be appropriate in the context of a book or website. In a book about kangaroos, we would not include the first; a road safety manual would find the second too general; while the third would certainly look odd in a set of statistical tables. Conceivably if the document were about the wildlife of Northern Queensland we might use the first two, but it is debatable.

The point is that the most basic indexing decisions are heavily influenced by overall document subject, and thus by the physical package in which our sample paragraph appears. We avoid using concepts already expressed in a document title as main entries, so that a book on growing oranges will not have half its index under 'O', or a biography of Mark Twain under 'T' (sorry, I mean 'C' of course!). This is an admirable principle, but one with grave implications for reusability.

To show how grave, imagine the index to a book on Antarctic exploration. This could contain the odd reference under 'Arctic', but probably none at all to the main subject, the Antarctic. If its publisher decides to extract a chapter (perhaps one dealing with the problems of coping with low temperatures, isolation and protracted darkness) into a sister volume on polar exploration generally, the text may fit seamlessly and (assuming the index was embedded) the locators will survive the move unscathed, but the new index will still be seriously deficient: any references to the Antarctic in the transplanted chapter (but only in that chapter) will be missed.

A more dramatic effect would result if our neutral-seeming chapter were incorporated into a book on *Arctic* exploration. This would reduce the index to nonsense, not only by omitting key mentions of the Antarctic but by introducing completely unhelpful ones for the Arctic, all strictly confined to the transplanted chapter. In case this seems far-fetched, I have met just this situation, involving a chapter from a textbook using mainly database A, which made occasional (and quite properly indexed) mention of differences from database B, being transplanted into a book entirely based on database B. So Peg Mauer's assurances about publishing extracts don't always hold in practice.

Our second aspect of index term selection involves assessing an entry's informativeness and uniqueness. Re-examining the extreme example where I suggested subentries were required:

> golden eagles, 25, 76–79, 83, 103, 105, 116–119, 123, 134, 149–152, 170 . . .

an alternative approach suggests itself. In just such a book, I recently found the first mention (say, page 25) estimated the number of breeding pairs of these birds on Skye and pointed out that they can occasionally be seen from almost any location. Isolated mentions on pages 83, 103, 105, 123, 134 and 170 might therefore simply repeat the same information when discussing individual bird-watching locations. If so,

they could safely be deleted as adding no further information. But if the chapter containing pages 103 and 105 is then removed and incorporated into a more local guide, it would obviously have lost all index mention of golden eagles. Also technical literature is full of such obviously non-transplantable sections as:

> the sorting procedure uses exactly the same approach we met in Chapter 7, so we'll not repeat it here

followed perhaps by several pages of code listing!

Problems caused by individual indexers' styles probably need no elaboration, since exercises where a passage is given to several indexers (or where an index begun by one has to be completed by another) show that term selection, like the editorial changes discussed above, aims at only a personal consistency. Paradoxically, embedding actually curtails some of the indexer's supposed freedoms, for example by imposing its own preferred filing order.

These examples demonstrate that an index is not a sorted list of text descriptors; it is part of an organic whole with the document of which it forms part. It is designed to fit between the covers of a particular book, say, as specifically as the rest of the content. And, in view of this context-specificity, we can revisit the remaining advantages claimed for embedding. Even resilience now seems less well founded: reformatting an unchanged document will indeed preserve the coherence of any existing index, but modifying it at all is risky, and transplanting sections perilous. Swapping Chapters 4 and 5 around, for example, will leave an index making apparent sense, but risks having a concept applied before it is introduced (a sad echo of that old quip about indexes being the one place where people die before they marry).

Problems with computerized term selection

A final example, though not specifically connected to embedding, illustrates just how subtly context can affect human indexes, and contrasts with how automated indexing packages have tended to approach text. Imagine that page 174 of a 400-page biography of David Lloyd George contains the following statement:

> During this wilderness year, David was often solitary, enjoying little company aside from his faithful Yorkshire terrier, Shorty.

This is clearly unremarkable and provides no useful information, so few of us would even consider indexing it. But a small change might make a huge difference:

> During this wilderness year, David was often solitary, enjoying little company aside from his faithful *giraffe*, Shorty.

Though ludicrous, this illustrates a point gleaned second-hand from Richard Dawkins (Dawkins, 2003) on surprise as one determinant of the information content of a message. It is surprise that lends the significance to Lloyd George's giraffe, and surprise is an element to which indexers have been responding almost instinctively for many years. But surprise too is context-dependent. In the biography of a British politician, a pet giraffe would be remarkable and bizarre; it would be less so (to the extent that it might no longer even merit inclusion in the index) in the reminiscences of a zookeeper. Context is everything.

These two examples can illustrate another aspect of human indexing we may take for granted. Getting back to the more plausible of Lloyd George's imagined pets, Shorty the Yorkshire terrier, suppose that some 30 pages later, the text records Lloyd George holding forth to a dinner party, or even to the House of Commons, comparing a political opponent wittily to his pet dog? For a human indexer, recall is triggered ('We've had this Shorty before'); a search reveals the (quite properly unindexed) former reference that now, since it gives additional information about Shorty, has acquired more importance, and its locator is added retrospectively. We can modify our indexing in the light of a later reference, revising our relevance assessment upwards just as, in the case of the golden eagles, we revised it downwards. And machine-readability is a huge boon here.

Parenthetically, let us examine how the giraffe example might be indexed. I suspect a computer would probably use the headword 'giraffe'. A human indexer probably wouldn't – or certainly not as the only headword, as 'giraffe' is unlikely to be sought – but perhaps would enter something like:

> pets, LG's taste for exotic

However the word 'pets' is not even mentioned in the source text, and a computer might have a hard job coming up with it unaided. This illustrates my last point about the modelling of indexing operations generally. Whereas models for automated indexing have traditionally seen an index as something to be extracted from the existing text, it is not. It is rather something brought to the existing text by the indexer.

I freely concede that my speculations about automatic indexing may be out of date, but the claims made on behalf of embedded indexing are certainly based on this kind of misapprehension about what indexers do, and also perhaps what readers expect from indexes.

Misapprehensions about our craft

The first two of the following quotations are around a decade old but – sadly – might equally have been written today:

> It would also be helpful if designers would speak with professional indexers and find out how we work.
>
> (Mulvany, 1994)

> [Embedded indexing modules] are designed by people whose idea of an index is an alphabetical list of words extracted from the text plus their locators.
>
> (Wellisch, 1996)

> We have let production managers and computer technicians decide what an index is and should be. Our only choice now is to help them make what they want to do work even better.
>
> (Prout, 2004)

It is noteworthy that all three of these commentators

complain *about* publishers – but *to* fellow indexers. As a profession of lone freelancers, we are hopelessly disadvantaged in attempting to influence multinational publishing corporations. We have so little status that indexers often are not even consulted over decisions about indexing. If anything can be done, it must be done through our professional societies, but first let us try to see how things can have got so bad.

Imagine a situation that may be commoner than we think. A publisher has a sure-fire best-selling title ready, the text finalized and paginated on the company's computers. Everything is ready to go to print, with a good chance of hitting the bookshops a few days ahead of commercial rivals with a directly competing title. Then, everything stops while proofs are laboriously prepared, bundled up and sent by special messenger to their usual indexer in Chipping Sodbury. Three weeks later, a fine index comes back, is typeset, and the book finally joins in unequal combat with its now well-established rival – and loses. If such delays can be fatal, are they even necessary? It is easy to see how publishers might be susceptible to salespeople and other siren voices urging them to adopt embedded indexing or fully automatic indexing ('just press one button!'); to outsource the indexing to India, or even to omit the index altogether.

As a profession, we are not Luddites. Some technology has been unequivocally welcomed and embraced. Most indexers use email, word processors, and either MACREX, CINDEX or SKY because they all help us to do more efficiently what we were doing anyway. Many of us accept our documents in machine-readable form as email attachments or FTP downloads. But sometimes we are being asked to do pointless, unproductive things, so in spite of our weak bargaining position, we need at least to prepare ourselves to refute claims made by systems specialists and repeated by publishers. We are not Luddites, but sometimes we might resemble ostriches, so an awareness and informed discussion of emerging technologies should be a key function of professional bodies like the SI. In our unequal working relationships, forewarned may not quite be forearmed, but the converse is certainly true.

Personally I think automated indexing – or automating the grind of indexing – is unexceptionable, provided a human indexer edits and refines the result. And much of what we do is surely capable of being modelled and replicated with modern computers, with their phenomenal capacity for statistical and syntactic analysis (or even artificial intelligence approaches, using the huge archive of indexed text accumulated by publishers), although I am no expert. On the other hand, we *are* all experts in indexing. we must make ourselves available and keep ourselves informed; we must subject embedded indexing systems and the other trends that threaten quality indexing to merciless professional scrutiny as they emerge, to avoid having them imposed on us in the future as we have in the past.

This article is based on a talk given to the SI Conference 'Connections: working in the present, learning from the past' held at the University of Exeter, 8–11 July 2005.

References

Dawkins, R. (2003), The 'information challenge', in *A devil's chaplain: selected essays,* London: Weidenfeld and Nicolson.

Mauer, P. (2001), `www.asindexing.org/site/Mauer_EmbeddedIndexing.pdf`; see also Mauer, P. (2000) Embedded indexing: pros and cons for the indexer. *The Indexer* **22** (1), 27–8.

Mulvany, N. C. (1994) Embedded indexing software: users speak out, in *Changing landscapes of indexing. Proceedings of the 26th Annual Meeting of the American Society of Indexers* (also at `www.bayside-indexing.com/embed.htm`).

Prout, D. (2004) Why embedded indexes are different, not better. *Key Words*, **12** (4), Oct–Dec, 136.

Wellisch, H. H. (1996) *Indexing from A–Z*, 2nd edn. New York: H. W. Wilson.

Bill Johncocks is a freelance indexer, specializing in embedded indexing using MS Word and XML, and living on the Isle of Skye. Email: `bill.johncocks@tinyworld.co.uk`