I’ve been sitting here in front of the computer all morning trying to get a custom-built indexing tool to work. The clients who gave me this software want me to use it to index HTML files, and it is supposed to do all kinds of neat tricks — put bookmarks into the HTML files, pull their file names and topic titles out into a spreadsheet for indexing, and then build a nice HTML index for handoff when I am all done.

But it’s not working — no topic titles are being picked up, the bookmarks are not going into the HTML files, and some really strange things are happening to the spreadsheets I build. I’m spending time trying to get this software to work, and not getting much indexing done.

Is this what indexing after the millennium will be like? No more books, only HTML files? And endlessly poking at malfunctioning software? Let’s hope not!

I don’t think books will ever go away, myself — they are too much fun, too satisfying, and no one wants to read a computer screen in bed before drifting off to sleep. (I can see it now — you close your eyes, drift off to sleep, and drop the laptop on your head — sweet dreams!)

**Standalone indexing**

As long as we have books, making a living by creating traditional standalone indexes from page proofs is a possibility. Many people will still be able to work the way they always have. But I see real changes coming — much as embedding indexing codes is a royal pain, I feel it is what will be asked of indexers more and more. Publishers have to cut their costs and reduce their time to press, and they see embedded indexing as one way of doing that, especially if pieces are going to be updated, translated into another language, or converted into online interactive books. Those codes allow them to manipulate their materials and still have them indexed, albeit painfully and at times not with the best quality indexes.

Even now, in my own indexing business, I get to do standalone indexing only one-tenth of the time. The rest of my time I spend with indexing software modules and tools, and focusing on getting indexing into files as well as making it good. I don’t see any way of avoiding it, and actually, I don’t want to avoid it! It challenges me to build a good index regardless of the tool and the displayed index interface. Each one is new and different.

Perhaps because I enjoy a good challenge, I see the next few years as an exciting time for indexing.

**Embedded indexing**

We are in a transition period in terms of embedded indexing, and yes, it is painful now. The software packages we are asked to work with to embed codes are much more primitive than we would like. One important role we can play is to make it less painful. I would like to see indexers assisting software engineers to create indexing modules we can really use. We can definitely make these tools easier and allow better indexing, and should be making our voices heard on this subject.

Software engineers need to understand how good indexes are built, what the process of indexing is really like, and need to redesign indexing software modules so that we can produce good indexing quickly and without all the silliness and extra effort we now have to go through. They need to help us by building tools that fit into electronic book production seamlessly — for instance, creating modules that allow us to build an index outside of the book’s files, and then have the modules push the finalized index codes into place. They need to understand that the current form of embedding-solely-in-the-file indexing modules hamper the publishing process.

Everyone from editorial to production needs to have access to the book’s files when the project is nearly complete, unfortunately right at the time that the indexer must have the files tied up to embed codes for several days. This simply does not work well in the publishing process. The process of reviewing indexing for updated book versions also needs to be changed, as that is where the real cost savings to the publisher will be realized. Currently, it is tedious to review existing entries — if we could get better indexing tools, we could make this process much easier as well. These kinds of feature changes must be made before embedded indexing can be at all cost-effective, and indexers are the ones who can help design the ways the tools can work.

**Interactive indexes**

Online and interactive indexes are another intriguing avenue to follow. Indexers are used to the interface of the printed page, and understand how to make print indexes work well for readers; now we need to look at how readers use indexes online, and how to design them for the best retrieval. How do we make them easy to use? How do the indexing rules change? For example, readers get tired and confused after looking through subheads online — it appears that five subheads are about the maximum before they give up and just choose anything.