

HANDBOOKS IN 2001

I review the handbooks saga for 2001. The software is working pretty well, but there were – as always – new fires to fight. Information of various sorts didn't get to me when I needed it, and that led to some frantic catching up late in the process. The lack of a real department database is a real difficulty.

BACKGROUND.

I've been doing this since 1997 now, so it should be easy. It isn't.

This is not because of any deep flaw in the system; the original scheme for the operation¹ still works pretty well, so far as it goes. That isn't very far, but it accounts for the most serious delay in this year's process. Further delay is caused by some inconsistencies between parts of the system designed for the annual handbook update and others directed towards immediate updates for web pages. These, along with some other consequences of haphazard development, can be ironed out with more thought and less reflex action.

Apart from that, the difficulties are mainly caused, in one way or another, by people. Some of those are in the department, and make life hard by not wholeheartedly cooperating; others are elsewhere, and impose complications by changing things (URLs, regulations, ways of doing things) unilaterally. There's nothing that I can do about that, except try to foresee possibilities for confusion and guard against them.

CALENDAR (approximate).

June 1	Send requests for people information.	
June 12	Send requests for handbook information.	
August 13	Unix locking problems with mailx. I decide to eliminate mailx	because this is the second time, and the problems cripple my system.
September 10	Start using notmailx.	
September 30	Faculty deadline for handbooks.	
October 1	I receive lecturing assignments spreadsheet.	
October 2	Send requests for courses information	because I didn't know who was doing what before.
October 15	Almost the final people reply arrives.	
October 26	The final people reply	because the E-mail address had changed, and nobody told me.
October 29	The final undergraduate handbook reply arrives	because the functionary didn't know how to tell inco that nothing had changed.
November 1	The final courses reply.	
November 1	First version of HTML handbook available.	
November 9	I receive the list of theses completed	because Penny had to get it from faculty (?).
November 13	Final replies for the graduate handbook	because some jobs had changed, and I didn't know.
November 14	ANOTHER courses reply	because not all the courses were in the first version of the handbook, causing unnecessary requests for resubmission.
November 15	I receive the course diagrams	because no one had asked Andrew to do them earlier.
November 19	Print handbooks complete.	
November 20	Authoritative version of HTML handbook available.	

The calendar lists some of the significant dates in constructing the handbook. It exists because I now have software to analyse my logs and inco's mail. It's approximate, because it relies on finding things in my mail

(using Eudora's search) and inco's mail (only saved systematically since I started using notmailx), and on understanding the headings in my logs (which I didn't write with a view to information retrieval).

WHAT HAPPENED.

We muddled through, as usual. Things did not go as smoothly as I'd hoped, because some of the improvements I'd made took a little while to settle down (= I did it wrong), and there were consequent interruptions in service. As against that, service, when it came, was in some ways better than before.

PROBLEMS ADDRESSED.

These are some of the higher spots. I haven't recorded any activity in January, February, or April, but apart from that I've done something or other every month. (I didn't know that before.) This list is therefore restricted to the larger exercises.

Unix troubles, particularly with locking files, started last year and continued from time to time. The most common effect was to refuse to lock the mailx output files, whereupon the run would finish prematurely, and nothing would ever leave the input mailbox. The effect was that inco's mail interface didn't work properly; the subsequent stages either received no input, or received the same again and again, and in either case didn't do all it should. The system is robust, and nothing broke, but many things were done repeatedly.

The problem was associated with mailx, which I'd used as the mainstay of inco's automatic input because it was there, it had been quite reliable in the past, and it had a number of functions which I could bend (approximately) to inco's requirements. The problems seemed to be solved when the overall mail system was changed, and mailx stopped dealing with the system mail; then inco acquired a standard mail file via fetchmail, and could use mailx locally. Then they came back again. Yes, it would be mended (and eventually was, they say, though it took a long time), but before then I'd decided that mailx was not sufficiently reliable, and determined to change. Now inco uses his own software to manage mail – which, in fact, is much better, because instead of bending mailx facilities to something like what I wanted I could do it all properly, but that took some time.

This led to some other improvements. Now inco is not dependent on mailx, it sends less cryptic messages to acknowledge receipt of responses, and also reports some errors (a new feature). It also sends better reports to me, so I have a much better idea of what's happening, and have been able to mend quite a few things – typically those caused by people who send syntactically invalid responses – very quickly.

The university web site has produced a significant crop of new complications with its widely admired (but not here) new design. There were several links to it from the HTML handbook, all of which were broken. One might have expected some sort of redirection as an interim measure. As against that, I had no complaints from people frustrated by the broken links, so it's possible that no one ever tried to use one. I had to spend quite some time trying to find them again in the new system, and not all are there.

It has not helped that the new university URLs include ampersands and question marks, both of which interfere with the inco text processing software. They don't now, but it took some time to identify the problem before I could mend it.

University information seems to come in a different form every year. Presumably they think it doesn't matter because everyone else uses typewriters. The most dramatic event this year was the absence of any easily accessible table of times and rooms for courses. They are available, after a fight, from the ludicrously complicated (and illiterate) "ndeva" system, but I quailed at the prospect of extracting the real information from either the displayed text or the HTML.

Andrew got me a table eventually, very cryptic, but at least reasonably compact. But will the same thing come next year ? This is part of a general problem with interfaces with the world outside. So far as I know, the registry computing operation offers no specification of data which it provides, and doesn't feel obliged to conform to any rules which would help people to use it. This is unprofessional.

Documentation is improving, and becoming more structured. This is partly from inclination, and partly from necessity. I have always had a list of software and what it does, and for some years I've had a list of what to do when certain events occur, but they have been less than systematic. As the system has grown,

they have proved essential to remind me what I should be doing, and I've been expanding the scope of the documentation to include more detail of what depends on what. This is incomplete, but already useful.

PROBLEMS REMAINING.

SYSTEM PROBLEMS.

I've mentioned some of these already as inconsistencies consequent on exploratory development.

There is much duplication of code; the whole system should be reorganised, and more effective modules designed. As it is, many of the changes required have to be made in two or three places, and inevitably consistency is lost. Often, it's because of the system's history, but it shouldn't be.

It is still much too hard to change individual files – sometimes harder than rewriting the whole handbook. That's because in the early years the HTML handbook did not loom large, so all the design went into the full handbook preparation. What I lack is a clear definition of the dependencies within the system. It's getting better. This is one reason for the documentation improvements mentioned above.

PEOPLE PROBLEMS.

Oh, boy. This is the big one, and it comes in several flavours.

Getting information to people is hard. First I tried to make sure that the information sent with the mail to be amended was complete, so they complain that there is too much instruction in the inco mail messages. Then I cut down the mail messages, assuming that most people in a computer science department will understand the idea of mark-up, and provide links to the web material, and they don't read it, and mess up the mark-up. (Some of the non-technical people do much better; some don't.)

In fact, there's a fair bit of documentation. I told them about it in the first message. I didn't repeat it thereafter because of the moans about long messages, and as I'd already told them it seemed redundant. Clearly it wasn't, and people forget (for example, there was a very late query on how not to change a file) – or they simply don't choose to look, assuming that I'll sort it out. Which, of course, I will, because we're already well past the deadline and I want to get the thing done.

Getting information from people is perhaps a bit easier, once they've got past the comprehension step. Most people do reasonably well at it most of the time, provided that it's no trouble. Once anything beyond simple turns up, though, some people don't try very hard.

- People don't use the markup symbols – particularly for URLs and mail addresses. It's fiddly (the curious format of some of the codes comes from the days when everything was done with Word macros, and is scheduled for change), but not impossible – and they're all documented.
- Bits of LaTeX appear in the returned stuff.
- Several people removed, or changed, the line which says "PLEASE LEAVE THIS LINE HERE".
- Several put messages to me – or the text which was supposed to be the reply – in parts of the reply which are automatically thrown away.
- A few simply don't reply. Some of those turned out to be no longer doing the jobs I thought they were, but they neither told me nor sent on the requests to whoever is doing it now. One job had apparently disappeared (Stage II coordinator).
- Misplaced helpfulness : After finding that some replies had information in inappropriate fields of the record, I tried to help with the course files by including blank entries for fields which supervisors might find useful; several people filled them all in, *as well as* listing another course from which it could all be copied.

Conflicts : People supposed to collaborate don't always talk to each other.

Expectations : Some people assume that I'm going to rush around looking for things – "You might be able to get this from Penny" – despite my clear (if hopeful) statement beforehand that I wasn't. It seemed to me that that several people in charge of X didn't know much about X.

INFORMATION PROBLEMS.

If inco doesn't get the information, he can't do anything with it. Some examples from this year's activity :

Functionaries : I assume that I know to whom I should send items for checking. The process is all automated from a list of aliases linking people to responsibilities. If I have the wrong names for the jobs, or the people decide that they're not going to do the jobs, disarray ensues.

Novelties : There were quite a number of new people whom, and some new courses which, I heard of anecdotally. The courses were eventually sorted out by the lecturing responsibilities spreadsheet.

Courses supervised outside the department (specifically 225) are a mess. Getting the course information is beyond inco's scope; I've tried to get stuff from the mathematics department, but it's very hard. It would be sensible to have someone in the department, preferably associated with the course, whose job it is to find the information. This year I invented "Contact"s for courses, who are not supervisors but deemed to be authoritative sources, and appointed them myself. (The contacts also turn out to be handy for courses without supervisors – generally project courses.)

Miscellaneous missing information : No one knows who of our graduate students have written theses, and what they're about. I'd assumed that the PhD coordinator and Dosgbody would know, but they didn't; eventually Penny got it from the faculty. And I've already commented that the timetable and room allocations no longer come to us in comprehensible form.

This is a clear indication that my original intention for inco – to coordinate the department's information² – was a good idea, and still hasn't been done. Vincent is proposing to address one part of it with his "Theses and Projects" database, which is good (though I think he's underestimating the difficulties), but there's a lot more to do.

FUTURE.

Is there a future ? Faculty are pushing a centralised handbook. Is it worth going on with our own ? I assume for purposes of argument that it is, but the problem should be addressed. On that assumption :

The system should have a thorough overhaul. The underlying structure is sound, but there's a lot of room for rationalisation and improvement. An interactive interface would probably be better, though there are questions of authentication and editorial oversight to be solved.

I think there should be an executive editor; experience suggests that it's silly for me to do it all, as I have done so far. I have drawn back from this suggestion, because an editor would almost certainly want a different sort of system, and that would plunge me into yet more software development which I don't want to do. (This is only a problem because I see inco as a research project, and wouldn't want to spend time on a purely administrative function; at the moment, I'm doing mediocre administration and getting nowhere with the research because of all the problems. From inco's point of view, a wholeheartedly administrative approach would probably be the better way to go.)

The goal for inco is that the whole handbook process should be reduced (for inco) to pressing one button, which might be labelled "make". We're not there yet, but this year we moved significantly closer. Most processes work smoothly when all is well, but I'm not as confident as I want to be that I can catch and contain errors.

REFERENCES.

- 1 : G.A. Creak : *Background for a document generator*, unpublished Working Note AC115 (September, 1997).
- 2 : G.A. Creak : *Information structures*, unpublished Working Note AC111 (May, 1997).

