

An Interview with
MARK P. McCAHILL
OH 328

Conducted by Philip L. Frana

On

13 September 2001

Minneapolis, Minnesota

Charles Babbage Institute
Center for the History of Information Processing
University of Minnesota, Minneapolis
Copyright 2001, Charles Babbage Institute

Mark P. McCahill Interview

13 September 2001

Oral History 328

Abstract

In this oral history Mark P. McCahill, Assistant Director of Academic and Distributed Computing Services at the University of Minnesota, recounts his role as leader of the team that created the popular client/server software for organizing and sharing information on the Internet. McCahill also describes his work in the development of Pop Mail, Gopher VR, Forms Nirvana, the Electronic Grants Management System, and the University of Minnesota Portal.

This is an oral history conducted by Philip Frana under auspices of the Software History Project of the National Science Foundation. The interview date is September 13, 2001, in Minneapolis, Minnesota.

Frana: Let me start with my burning question which is central to how I've been thinking about Internet Gopher, and that is Tim Berners-Lee, in his new book, - I don't know if you've read it...?

McCahill: No.

Frana: He says, "A computer typically keeps information in rigid hierarchies and matrices, whereas the human mind has the special ability to link random bits of data. The human mind usually sees organizing structures all the time but can also break out of them and make intuitive leaps across boundaries." And the question is, did the Web and Berners-Lee have a more accurate model of how human cognition works than Gopher? Was Gopher easy to use because it was natural or was the Web more natural and more familiar?

McCahill: I don't know. Some of its engineering trade-offs are some of why I think Gopher was an early success, versus Web had everything to do with what the environment was at that point and maybe less to do with how human cognition works. Certainly when Farhad and I, and the rest of the guys were putting together Gopher, we were more than aware of what the limits of what the technology could do were. That a lot of this was going to be access over incredibly small links by today's standards, and so we couldn't send that much information and we wanted it to feel fast. That was the main thing. It has to feel really really quick, so that is going to limit you in what you can do some. The issue we always had - we still have to an extent- with hypertext or hypermedia as the only way of getting at things is that sometimes I do want some structure because it helps me find things. I know that it is in that room and down there. As opposed to this feeling of wandering all these great connections and serendipitously stumbling into interesting things, but not being able to get to what I want. So Tim and I sometimes disagree. The best information system I ever saw when things advanced to the state where CPU cycles and bandwidth wasn't the issue was the guys out at the Technical University in Graz, in Austria. Hyper-G, in my opinion, rocked the house. That was, and is, really cool, because it combined both the idea that you could have some hierarchical organization to organize this stuff with hypermedia. So I think a melding of the two is probably what I, personally, like the best- or what I think is the best model. I want to use the hierarchy to get in the right neighborhood and then I'll look for- jump it around and having all sorts of interesting links.

Frana: Was it like Gopher VR? Was it through a 3-D multimedia ...?

McCahill: They could do some 3D stuff. But they had a complete authoring system where you could take a page, say these are links, they link to this thing over here, put it in by hierarchical organization of documents in that place in the hierarchy, and the links were two-way links. Much better than anything Web has, so that if a document was moved it didn't break the links. It had a big object database underneath things. But technically the University of Graz had been looking at hypermedia and information systems for a long long time and so it is not too surprising that they

kind of synthesized some of what was going on in Gopher and some of what was going on in Web. They had the luxury of taking a look at those things and saying, 'Here is what you've got wrong guys.' I think they did it better than either of us. They also worked way too late in the game to make a big impact, and the problem that they also had is it is hard to do authoring that way. It takes an authoring system and some self-discipline. Where the early days of Gopher you could slop information up like that [snaps fingers]. It was the same thing with Web. After they got some better tools, you could slop up something with almost no effort. The 'Net'? Is that a good model of human cognition? Probably. [laughter]

Frana: You weren't really thinking of cognition? You were sympathetic to the librarians and how they thought in terms of folders and files in the manner of the UNIX directory?

McCahill: I think we were most sympathetic to the library model of, 'Let's put together the things that go together, together in the same place' - some notion of organization and structure. This idea that 'Oh, you are just mapping the file system of the underlying machine over there' - uh uh [no] - that was never in the intent. Any of the halfway good Gopher sites were doing things to subvert what the underlying structure of the file system was to make what appeared in the Gopher menus up to the site administrator, as the right hierarchy for the design of the information space, versus how we happen to store it underneath.

Frana: Okay.

McCahill: There was a way of overriding. By default, things would show up in hierarchy that would not to your UNIX file system or your Mac folders. But you could override that and say, no, actually I happened to store it over there, but please make it show up in the Gopher menus in this place.

Frana: Okay. That's an important point.

McCahill: Yes. Because again, I think it's maybe my background and Farhad's that we both like libraries a lot. My idea of a good time is to just go wander around where there is a big collection of books, where some of the books are on the shelf next to each other and I find the one I was looking for and then I look at some others really close to it.

Frana: Shelf reading- grazing they call it.

McCahill: Yes. Browsing.

Frana: And what about the NeXT Step operating system? Does that figure prominently into some of the decisions you made?

McCahill: Yes. NeXT's were really important because I was in love with full-text searches, but I didn't want to write a full-text search engine because that would have taken a long time and I didn't really want to do it if I could avoid it. I happened to have some NeXT machines sitting around. And one of the things NeXT was flogging with the NeXT machine was, 'Hey, there's a

digital library.’ In other words, full text searches over a body of work. So one of the key things for Gopher was- Farhad had got a hierarchy that you could navigate going, but I knew we’d never have a saleable product. Saleable inside the university to our bosses is something that was good, unless there was some kind of way of searching. At that point everyone was obsessing on search on keywords or do some kind of SQL database to search this stuff. But I’d seen full-text searching from some stuff I’d seen at Apple. They were doing some things with their technical information library where you could put in words and it would find all the articles that had those words in it. So we had got exposed to full-text searches and then had seen a little bit of WAIS stuff- Brewster Kahle’s full-text search engine. I said, ‘Great, if we could weave full-text searches onto a hierarchical structure’ - the hierarchy gives you a way to browse and organize the information and do the shelf grazing that you are talking about. The full-text search engine lets you find things that you wouldn’t find by shelf grazing. ‘Where are we going to get the damn full-text search engine?’ And the answer was, ‘Well, gee- NeXT has one. I wonder if there’s any API, if there is any way for me to use it for something it hadn’t been intended for?’ I happened to know the NeXT salesman [Greg Smedsrud] for this region really well. He’d previously been our Apple salesman and the university had sold a ton of Apple’s to students, so he liked us. We were some of the guys that had helped with that. So I twisted his arm really hard and said, ‘There has got to be some kind of library or API in your digital library, so I can use it for something else. It’ll sell NeXT machines Greg, give me this stuff.’ They gave me a library and I found a way to kind of make it compile and work, in spite of no documentation, and then glued that into Gopher and then we combined the two. So we ended up having to do some pretty funny stuff where we’d want to have a collection of documents full-text searchable, but we only had one or two NeXT machines and we couldn’t afford to buy more. So we started telling the salesman we’d buy more, but I couldn’t afford any so we’d have to move the collection of documents onto the NeXT machine to run the library builder – NeXT’s full-text search engine made a pass through the documents to figure out what words were in which one, index them on the machine, then you’d throw the copy of the documents on the NeXT machine away, and tell the NeXT box that the references to the documents it returned should point to the other server (where the document’s permanent home was). So that the NeXT machines end up just holding the indexes- the inverted index of what’s in these things on other Gopher servers. All driven by, we were cheap and didn’t have any money to do this. But combining those two things together, then I knew we had something that I’d want to use myself. You could do it both ways. Search when you didn’t know where it was, but also when you did have an idea of where it was.

Fran: So, when did the University of Nevada, Reno people get involved?

McCahill: A little bit later. First, what I’d consider the first real iteration of Gopher, was when we had NeXT stuff working and we had servers and browsers on Macs and UNIX and something running on PC DOS, which dates us a lot- as a DOS client. At that point, we had something that was viable and you could do great demos and get people really excited. And that was in April, May, June, of what, ‘90-’91. The Nevada guys showed up about six or eight months later- out of the blue. We’d been making jokes a little bit about, ‘Well, if there ever is some equivalent of an Archie for Gopher, then it should be called Veronica.’ And, because it would be a good goof on Peter Deutsch and Alan Emtage. And the discussion in Nevada just popped up and they sent us an e-mail saying, ‘Hey, check this out.’ ‘Oh, that’s really cool! Here’s what you should call it-

you should call it Veronica. That would be a good goof on Archie.’ [laughter] ‘Oh, yes, we should!’ And they thought of an acronym to make it make sense. So there wasn’t any real need for a Veronica until you had a lot of Gopher servers out there- a lot being fifty or one hundred. In the early days, there were like two or three: us and then a couple other schools like Michigan State and then you were getting to the point where there were fifty or one hundred. At which point it could make sense to go and walk the tree, find the stuff, build an index and have a global Gopher index.

Frana: Now were you working on this before the campus-wide information system idea came up?

McCahill: No. This was a reaction to the campus-wide information system stuff.

Frana: I talked to Paul Lindner a little bit about this. Farhad called the first proposal for a CWIS “a design by committee monstrosity”. He was upset that the university saw Gopher engineers as “lowly Mac/ PC/ UNIX support monkeys,” as he put it. So was this a culture clash?

McCahill: An organization clash. Here’s the deal - at that point there was not a permanent Chief Information Officer boss of the Computer Center for the U of MN Minneapolis campus. There were a couple of people who were interested in the job. Mark Luker, who had been doing that kind of thing for the University of Minnesota up at Duluth, was down in the Twin Cities in an acting position - not a good situation for any person, Acting Chief Information Officer. CIO’s are already tenuous, if you are ‘Acting’ you are really tenuous. There had been a lot of talk among the CIO level people at all of the big universities about, ‘We should be doing campus-wide information systems’ - this is a cool thing. The mainframe people were looking at campus-wide information systems as their salvation. People had been fleeing mainframes like crazy at this point. Everybody was getting their own pc or Mac or something and doing computing on the desktop. And there were questions looming like, ‘Why do you even have a Central Computing facility?’ and ‘What are those mainframes good for?’ So with these two ideas loose in the world: campus information system- of being neat because that would show the computer is being used for something other than computation- it would be good as publishing vehicles, ‘Gee we’ve got to do something with the mainframe and central computing infrastructure.’ The mainframe guys looked at this as, ‘Oh, well, that’s simple. We’ll just write an application and you’ll come to the mainframe, log in and do the VT100 terminal sessions.’ We had the distributed computing religion in a big way, and my group were the people who were doing development and answering questions- doing help lines for microcomputers. We’d already done an e-mail application to run on Mac’s using the POP protocol - POPmail.

Frana: Oh, okay, this is the second one?

McCahill: Yes, and we did others like it. Pop Mail though was cool because you could have a Mac user-interface over the top of your UNIX e-mail. And it talked TCP/IP over the network, or over a SLIP dial-up connection. So we had some infrastructure in our development workshop for writing IP applications and we had already done one, so we knew we could. At that point, the mainframe guys were saying, ‘The campus-wide information is mine.’ And we were saying,

'You are crazy, stupid, - everything should be done on PC's.' We were PC radicals at that point - PC and workstation radicals. The acting chief information officer did about the only thing he could do to mediate the dispute, which was, 'Well, let's study it in a committee and let's get lots of users to say what they want.' And the committee then made a big hairy design. And because it was a committee - and I was the only person representing the little computers- pretty much every time they wanted to do something I had one vote and everybody else had all the other votes. So what they wanted to do was something that would be very difficult to code, and we had all the work to do. Because there were supposed to be clients for DOS and Mac and UNIX and- it was just not a good scene. So Gopher was definitely a reaction to that whole campus-wide information system thing. But when Farhad came up with the very basic protocol of, 'Here, we can do it in a very simple way. This would be easy to code.' Then we saw a ray of hope that we wouldn't be stuck being the monkeys coding stuff up ...

Frana: That's where that comment comes from!

McCahill: Yes. Because otherwise it would have been these other guys, who don't have to write the code, dreaming up what you should do, throwing it over the fence and saying, 'Get going there.' When we got the NeXT full-text search engine working with that hierarchical browsing thing, we said to ourselves, 'We have just won' - because these guys haven't even figured out that what they want is full-text searching. They are thinking people are going to go and put keywords in documents. There is no way people are going to keyword this stuff. This lets you search in any way. And what is really ironic now, for me is, we were (the radicals) what - ten years ago- saying 'full-text searching is great. SQL and relational databases suck. Forget that.' Now, the bulk of what we are doing is SQL and relational databases.

[laughter]

McCahill: But back then, full-text searching was not a well-understood or well-known thing. It was one of the weird realms of, you know, library science and that stuff. So those few things, we said, 'We've got something better than any CWIS [campus wide information system] that those guys are going to do. If we just do it, they won't have a choice but to fall in line and do it the way we want.' And that's where some of these titanic internal battles about, 'what's the CWIS going to be? What campus wide information system - who's right- who's wrong?' - came from.

Frana: And so as late as 1990, these mainframe people really weren't buying into distributed computing at all?

McCahill: No, it threatened their jobs. Why would they want to buy into it? They had become comfortable and were good at centralized big-box computing. And that's where the Computer Center's money came from too. So, there were big entrenched interests for continuing to do that if at all possible. And we were the guys who were threatening to upset the apple cart in a couple of ways. Or at least, we were the agents of change, in that we were helping people use the little computers, and we were moving our stuff off the mainframe. Bang- they can do their computations fast enough. So mainframe as a computation engine for most people didn't make sense anymore -do it on a spreadsheet on your PC.

Frana: Sort of jumping ahead a little bit here, but how then does Gopher get distributed across campus and how does it escape? How did people find out about it? Did they download the software and set up their own servers?

McCahill: Well, we put out an announcement. There was a campus-wide information systems mailing list or newsgroup on Usenet. So we put out an announcement saying, 'Hey, we've got this thing you might be interested in!' And we had the software up for anonymous FTP, in the grand Internet tradition. 'Here it is- come and get it!' And a few people grabbed it and played with it. And typically what was happening then, there were individuals who were interested in this kind of stuff or had heard about it from their friends, they would grab the software, set it up and get it running. It wasn't typically an institutional decision that this is what we are going to do. Nearly always, there were a couple of unofficial sites that would start doing this themselves, and then after a while, the official guys would come in and say, 'Now I'll run this.'

Frana: That would explain why in '92-'93 in the state where I was located, there was no official documentation. It just sort of permeated through.

McCahill: It just happened.

Frana: Okay. This was distributed by a GPL license?

McCahill: In effect, for server, anyway. The clients - we didn't hand out source for- which was interesting and actually was a strategic decision. Partly because we, in our minds- I put too much work into figuring out how to write really good TCP/IP applications for DOS and Mac at that point. It's not easy. I don't want to give that away. That was the thought process. Also, I want to control what is in those clients because then if I control the client, I can drive changes on the server end and move stuff forward. If you don't control the client, you are in trouble. And you see that reflected in today's computing environment where Microsoft controlled the desktop and the rate of change is dictated by Microsoft. Microsoft can decide it doesn't want to put Java on Windows XP. It's an add-on, and Sun is shrieking now. Right. Sun has got a real big problem. We knew controlling the client was a big big deal. So that one we held on to. Also, the number of people that couldn't read and compile it successfully weren't that high. On the server side, we put source out partly because it was easier to distribute than putting out binaries. And the server guys tend to be suspicious when you put source and you are not compiling some, or running some Trojan or something, on their server. So, effectively, the UNIX stuff was the equivalent of GPL, the clients we weren't giving out source for. But we were giving the client away for free. But, I learned important lessons from that. Control the client. If you control what Joe Average User is doing that's the hardest thing to change. It's relatively easy to upgrade the server and change that- there are only a few of those. There are zillions of users. It's hard to change them. If you lose control of that end of the equation then you don't really control the environment at all. And that- in the end - is what really threw Gopher over. What killed the protocol off was when people stopped using Gopher clients specifically and started using Mosaic. Mosaic had a shitty implementation of Gopher, and they had zero interest in improving their implementation of Gopher, or supporting any of the new stuff. But once the bulk of the people were running Mosaic

we could think of new things to do with the protocol and improve the client all we wanted. It didn't matter. I learned a good lesson. Control that client.

Frana: So what happened? I don't remember ever using Gopher over Mosaic. If you typed in the box, you know, boombox.micro.umn.edu, what happened?

McCahill: You would get the same information with the name. It's the hierarchy and you could scroll down it.

Frana: Oh, okay.

McCahill: In fact, one of the really funny things about the days when the Web was trying to get its feet underneath it and get going, is that Gopher had got there first, and there were, at that point, a lot of servers- hundreds and hundreds. Thousands even. In the early days of the Web people would put up a couple of Web pages and typically after two or three pages of Web you'd be on the Gopher sites. You might not really notice it, but you were on the Gopher sites searching documents in Gopher.

Frana: Okay, so I did use it a lot after all?

McCahill: You did. You just didn't notice it.

Frana: That's a problem. So why does Shih-Pau Yen get so much flack for shutting off Gopher if that's the real reason? Now it seems to me that the chronology is that - I mean, if I remember right - there was a memo that you guys produced that said, 'Look, this is costing us money, and if you are a commercial user you need to pay.' And then Yen said, 'We are going to start licensing.'

McCahill: Yen gets a lot of flack because Yen is our boss, so he always takes it.

[laughter]

McCahill: Well, let's get the chronology right. So this CWIS stuff comes up. We don't like what the committee is doing. We put back a minority report back to the committee saying, 'Here's what we are doing. It's better than what you thought of.' Which they didn't like, but ultimately had to deal with. And we let it loose in the world. At the same time this is happening is when the budgets are getting chopped big time for the Computer Center. So there are painful things happening and the potential for layoffs. Or comments like 'Maybe we should outsource the entire Central Computing operation at the university' 'Everybody doesn't have a job but you can all apply to the outsourced operation' 'It's someone else who will maybe do this function instead.' There's a bunch of that going on.

Frana: This is all tied to the growing irrelevance of mainframes?

McCahill: This was due to the fact that the university, at that point, was tight on money. It's not a good time at the university and maybe we could do it more efficiently. There might be a better way of getting along, buying with less bucks, because the university needs to get more efficient, blah blah blah.

Frana: Okay.

McCahill: So, that's going on. Gopher is now starting to get rolling and getting some people using it and people are even saying it's a good thing- the university is doing this cool thing. The budgets are still really tight. And then we start seeing some commercial places going, 'Hey I can use this.' We don't have much choice but to say, 'You know, budgets are super-tight here and we are putting a ton of effort into this. There needs to be some hope of some kind of payback at some point or we can't put so much effort into this.' So that was where the - 'If you are commercial you're going to need to license this stuff if you are going to use our implementation.' The commercial guys who had been getting a free ride didn't like this too well, of course. A free ride was much better than thinking about paying. Although, you got a fair number of licensees and there was enough of equal licensing to keep the internal politics at bay that we got to continue to work on this stuff. But Yen got a bunch of flack for, 'Oh, he killed Gopher - he did this terrible thing.' Yes, but if we hadn't done something about making some moves towards trying to recoup some money for the development we were putting into it there was no way that I could keep having my entire development staff and me working on this essentially full-time.

Frana: How many are we talking about, at this point, on the team?

McCahill: Let's see ... Farhad, me, Paul Lindner, Bob Alberti, Dan Torrey, Dave Johnson, - so five to six people. It's not cheap.

Frana: Yes, oh, it's not.

McCahill: And at that point we are also doing the UNIX client and server, Mac client and server, PC DOS client - we never did a server for DOS because we didn't like DOS very well and it's probably not the best OS for doing an Internet server on. So we were doing that stuff. Plus, trying to advance the technology - that took a lot of people and a lot of effort. So people wanted a convenient scapegoat- they used Yen. But maybe the real truth had something to do with ultimately losing control of the desktop and the client when Mosaic came out. And another problem was, with success, things change. Gopher became the big deal. The visionaries at all the universities and all around were trying to sell this cool new thing they've got- this thing called the Internet. They needed some poster children to show how cool the Internet is. Conferences with librarians or potential Internet users and that kind of stuff. So, there's a never-ending - for about a year and a half- set of road shows, that Farhad and I went on. When you are on the road you are not writing code. You are doing talks. And that's distracting. So, that's another issue: it is hard to both sell that concept to technology and advance it and put on Gopher conferences and keep the corporate guys who are now spending some money reasonably happy with new releases and all that.

Frana: So, are you saying then, that over that year and a half, you didn't really see the sea change coming? I mean, in a way, you didn't worry about it?

McCahill: I saw it, and I didn't think it was that big of a deal. Because, again - I like the idea of some kind of structure, you know put the books that go together in the same shelf please. It shouldn't all be in this big jumble and you'll just find it stumbling through the big jumble. That never sold with me. The other thing was that when I first saw it it was too high-end. It was only running on UNIX boxes and we were saying, 'Well, that's nice. That's cool. But the bulk of the users have crappy little Mac's or PC's, the bulk of the users are using dial-up slip connections and if they are lucky, have 9600 baud connections 'Man, that's fast!' Given those constraints, here is what we've engineered that runs reasonably fast. Your stuff -you don't have the desktop- it wasn't until they were able, until the PC's and Mac's got a lot faster and the bandwidth got quicker, that you could really run a Mosaic browser on something that an ordinary user could afford.

Frana: Now that took more than a year and a half to implement?

McCahill: Yes. So we said, 'Well, it is an interesting high-end kind of thing. I think it is missing some stuff: full-text search engines and hierarchy. In our mind, this was a very interesting type of document. And we're about the shelves in the library and the search to find those documents. So the right thing to do would be to treat this as another kind of document. And when we hit an HTML document, what we, Gopher browser wanted to do, was hire up something to display that HTML; show them the page and let them look at the thing. But that's a document-view, as opposed to a complete information system. So where we were trying to go with this is have a lightweight thing for browsing and searching that we'll call helper applications or plug-ins or something, to view different document types.

Frana: Okay. So it would be just like any other file type.

McCahill: Let's say you've got a QuickTime movie. Cool. Let's run the helper application or some plug-in thing to do that. I don't want to have to write something to handle every document type in the world. I don't have enough developers to do that.

Frana: Okay.

McCahill: That's how we always wanted to handle the HTML stuff and that's probably also how we got blind-sided. It started out as too high-end, and the Mosaic guys have different approaches to writing browsers, which was put it all in one, which we hated. We liked the idea, 'Do one thing really well.' It's our best-of-breed intercooperating applications. One of the fundamental arguments in software has always been, 'Do it all in one. It's all there - this big integrated glob' - or no - 'Do a bunch of individual things that are good at one thing and let them cooperate.' I've always come down on the small best-of-breed individual components side. The market place is always excited for mediocre, crappy, big, stuff that's got it all there - done slowly and poorly.

Frana: Like the System/360.

McCahill: Yes. That's been another one of those continuous themes I keep seeing over and over again. Apple with its OpenDoc stuff had this fabulous idea of like little individual display units that can snap together. But it came at the wrong time and it took too long to get it together, and it never flew. That direction is what we were thinking.

Frana: Okay. Now, the pretty pictures- the inline graphics- did they amaze you? One of the early reviewers of the article I am working on wondered if that was one of the things that escaped your attention at the time? That you didn't see that the images were the same as content, and that people would view them as just as important as the stuff, the text, that was in the Gopher selectors.

McCahill: I don't think that was missing at that point. We were thinking, okay, well, that is in the document on the shelves that the books go on. We build the shelves and organization for the books.

Frana: You still stored all those images, right- just because it wasn't an image-based system directory.

McCahill: So there was some of that. The other thing in our minds was 'Shit, I've got as many developers as I'm going to get for this stuff.' I had really a great great development team for its size and that time. People could write IP applications at work, they were fast, and I was lucky as hell to have assembled that kind of team. But we had all we could do with just keeping the current stuff going. So the notion of, 'Now we've got to support some kind of page description language too? Shit.' The strategy at one point, in our minds, was, 'Well, okay, this HTML thing is okay,' but - my girlfriend Wendy Jedlicka is a graphic designer, which is why we had T-shirts and stuff. Since Wendy cares a lot about exactly how things look and placement- I mean, what do graphic designers do except get the font right and line everything up perfectly? The idea of HTML, which said, 'Well, I'll give you some rough hints about kind of the size and you kind of figure out how it gets laid out on the page,' was a little bit offensive to the graphics designers' sensibilities. If you really care about pretty graphics, HTML is the wrong, wrong, wrong place to do this. At least early incarnations were completely wrong. We really liked PDF, because you got absolute control over layout, location and all that stuff. We ended up working with Adobe, beating them about the head and shoulders, saying, 'You guys are sitting on a gold mine. Put URL's in this and have a way to call an external handler to go resolve the URL and you'll have something that kicks the butt of the current Web.' It will be prettier; it will be easier to author stuff. Now, it's not addressing some of the issues of, 'But I want to know the semantics,' at all. We were saying, 'But if you want pretty, do it in PDF.' But the Adobe guys didn't get it.

Frana: It still might not be too late for this.

McCahill: Well, it is sort of happening now because they do have link handlers in there and that sort of thing. But unless they control the client and the desktop, it's too late. And they don't control that. So, it's too late. However, that was the take we had on this: (a) I don't have the

people to do a competitive page description language. I've already seen one that is a markup language- HTML. Okay, great. I shouldn't reinvent that wheel; (b) If I want graphic fidelity, PDF looks like the way to go, but I don't have the people to compete with that. All we can do is beg and plead with those guys, 'Would you please make your stuff smarter?' But they didn't get it or weren't paying enough attention at that point. And probably they understood that without control of that desktop, without control of the default thing you're dead- it's not going to happen. For a little while we were in the enviable position of having our clients be the *de facto* standard that would go out on Internet tool kits the ISP's would give to their customers and that stuff. So we did control desktop. Microsoft hadn't bothered at all about bundling standard stuff with its OS for Internet. Apple hadn't thought about that at all really. So for a while we controlled that. Later when the big boys got in the game, we didn't.

Frana: Did you ever think about leaving the university when you were completely overworked and had too small of a staff to support all that you were working on?

McCahill: Well, I think that the timing wasn't quite right. Farhad and I made a bunch of jokes about how - 'You know, we were too early.' The wave of venture capital wasn't there. That all happened later. Quite a bit later. So, yes, we thought about it a little bit and were approached by a couple of guys with bad offers of, 'Here, I'll own everything, you can be an engineer in the corner.' So I have less control. No. It was at times frustrating - but also fun. Partly, having a really good little development team is like having a sports car. They are super maneuverable. You can quickly dart into little holes you find. So if you are into software engineering having the equivalent of a little sports car is a lot of fun. You can't take it off road, meaning you can't go head to head with the behemoths like Microsoft - but being able to dart in and out of traffic is a lot of fun. So that's been enough to keep me engaged. Yes.

Frana: The university environment- I know it has become more and more of a big business. Does university life appeal to you more than commercial activity?

McCahill: So far.

Frana: I mean, selling stuff.

McCahill: Yes. Yes. I think part of what I like about the university is- it sounds stupid- but once in a while I get to work with, really smart faculty on projects. I get a real kick out of working with really, really smart faculty. And it is sometimes about advancing technology or advancing knowledge, as opposed to, 'How can we make a buck?' Anybody that has worked at the university for twenty years or so has to buy into a little bit of: 'I want to advance knowledge.' Making a buck is also important, but it is not the most important thing. And anybody who kept working at a university through the dot-com hysteria definitely has to think that.

[laughter]

Frana: Maybe we'll get our revenge.

McCahill: We still have jobs.

Frana: That's true, that's true. How did the High Speed Computing Act affect all of this? It came out in what- '92 as I recall? And it made all those pronouncements.

McCahill: Yes.

Frana: Did that affect your work? It kind of comes at a critical time.

McCahill: Yes. It changed some things. Acceptable use policies were a big deal in the early days. And when things started moving away from Internet traffic having an acceptable use policy prohibiting any commercial activity, then it was possible for the commercial guys to actually start doing commercial stuff without twisting or bending the rules a little bit. In the early days I had to explain to the networking guys around here that it was okay to be doing some of this stuff because we can use it for the students; it is not bad to do this. But the early AUP's were getting to be a real constraint on what was okay. You had to explain why you were doing something. So moving out of that realm to 'It's going to be commercialized,' 'There's going to be commercial players publishing stuff on here,' was a big deal. The early days of Gopher there were people who were putting out basically little pastiches of magazines. Here are some articles from this magazine placed with that magazine, and places to subscribe. Some of the, actually, Larry Masinger and that crowd, no not Larry - some other guy who was involved in that a bit.

Frana: Ed Vielmetti?

McCahill: Yes, maybe. How we had an inkling that maybe someone would pay for Gopher server was that early commercial publishers started doing that kind of stuff. So that made a big difference and that led to more bandwidth, which led to you could do a lot more, and it fed on itself.

Frana: These acceptable use policies- you're talking about the old NSFnet, and even on Arpanet, e-mail was sort of frowned upon unless it was for official business. But at the time you are talking about- the late eighties and early nineties- what were the acceptable use policies? No commercial content. Were there other prohibitions?

McCahill: No commercial content.

Frana: That was really the big one. Personal communication was okay as long as it was related...

McCahill: to research or teaching.

Frana: Okay. And after '92 that was all sort of wiped away.

McCahill: That started going away because how the university and other universities were getting their backbone connection started changing a lot.

Frana: Now, today can the university put logos of ...

McCahill: sure.

Frana: ... Sun Microsystems on their web pages?

McCahill: Sure.

Frana: As advertising or as sponsorship?

McCahill: By the way, what's the difference?

[laughter]

McCahill: Sponsorship is high-class advertising isn't it?

Frana: There are better commercials on PBS now I guess.

McCahill: Yes, what's the difference? Either way. Yes, we were catching the tail end of that, 'It's just for research, don't be doing commercial stuff here.' That stuff was just starting to die with a whimper. And that's- again, the timing was right for us- maybe a little bit early, but just about right. Our bandwidth was a big deal. I think also that Al Gore hyping the information highway resulted in me getting a lot of frequent flyer miles. Because they were doing road shows.

Frana: So this fed into your new lifestyle.

McCahill: Well, lots of people were putting together library groups- 'Here's the future, a Brave New World.' If you listen to Cliff Lynch talk about the Brave New Library World- and see Peter Lynch, the guy who did Archie and Mark McCahill and all those Gopher guys, talk about some of the systems. This is what is happening! The world is changing now! There were a ton of those kinds of things going on.

Frana: So you became sort of a rock star- I mean, at least that was how they wanted to present you.

McCall: In a little niche, yes. I got to meet- what was good about that is I got to meet some really interesting people. And there were some very interesting trips. A good friend of mine now, a guy named Anders Gilner, out of Stockholm, was working for KTH, I think, at that point- one of the universities over there. And they were putting together NordUNet conferences, which were kind of Nordic University network thing. There was a really cool conference that had Peter Deutsch talking about Archie, Tim Berners-Lee talking about Web and that stuff, me talking about Gopher... so at that point, there were a lot of those sorts of, 'Let's get three or four people who are doing information systems together, they'll lay it out for you guys who are interested in networks, you go back to your institutions and sell this idea to the locals, and say this is why we

should be building network infrastructure.' It was fun. I got to see Archie demos to the point where I could do them...

[laughter]

McCahill: ...enough of World Wide Web demos where, I could do those too. And I am sure they could do mine!

Frana: There were a lot of familiar faces when you did these things?

McCahill: Yes. Yes.

Frana: Now, perhaps this is a personal predilection, but I'd like to be able to say that Gopher isn't dead. That's Marc Andreessen's great pronouncement. These Web-enabled cell phones they look just like a Gopher menu.

McCahill: They look just like a Gopher menu because they are under the same sorts of constraints that Gopher was under when we engineered it. We don't have that much bandwidth in the current cell phones, right? 9600, or 19,200 is about what you've got for data bandwidth out of current generation cell phones. You don't have that fast of a CPU, you don't have that much screen resolution or space. Given those constraints, you are going to end up with about the same kind of thing. It is just going to be kind of text oriented, because that's what you can do with those limitations.

Frana: But so far as you know, people haven't said, look this has already been done and this works and we're going to bottle it for cell phones?

McCahill: No.

Frana: Because it's simple and stripped down that it makes sense.

McCahill: Yes. But I keep going back to, I think if you are under those constraints and trying to do it simple- which you have to under those constraints- you will end up with the same kind of look and feel whatever you did in the back end. And if you are trying to minimize what you are sending over the wire you are going to end up with, if not the same solution, something very close, where there's a bit or two that says what is this object? Is it a document or a menu? It's name, where to get it- that's about as low as you can go for information to be sent.

Frana: Okay. Now I've run into a lot of librarians who tear up when I mention Gopher because they liked it so much and because it made sense to them intuitively. Have they found a way to keep using Gopher, in any fashion, in any way?

McCahill: I don't know.

Frana: I know a lot of librarians just really hate the Web.

McCahill: Well, because librarians are about organizing information. The Web is not about organizing information. Or if it is, the organization is embedded in those documents in a way you can't abstract it out and show it some other way. One of the reasons that we did the Gopher VR stuff, the 3D Gopher thing was, we liked 3D and we thought it'd be cool. But also we wanted to make a point while we still could, you know, if you would separate the organization from the content you could display the organizational structure a bunch of different ways. Web is only belatedly getting around to handling with some of the new extensions, a way of saying what a thing is.

Frana: Oh, all those tags and XML.

McCahill: Yes, all that tagging and XML stuff finally gives them a way to have some meta-information and finally gives them a way to say, actually this thing is say, the equivalent of a directory or part of a hierarchy. That hasn't been there until very very recently. So, of course, librarians are going to hate something where you embedded the organizational structure into the content with no way to retrieve it, other than a human sitting there. You are building some kind of structure like librarians do- that's crazy to us, don't do it that way. What we found out, though, was while librarians like organizing things and while we like it- and I think Joe User actually kind of likes that- you see it in some of the search engines now we are starting to overlay these sort of hierarchical organizations over their search results. People actually- that's one reasonable form of cognition, not the only, but one - building that is some work if you are going to do it on more than a small scale. If what you are doing is a brochure ad you don't want to bother with that, there's no point in it. There it is all content. So for slopping stuff up onto the net, and for commerce, where it's a brochure, you don't need that. It's not the right thing. If you are a document oriented information systems, fine, Web's great for that. So, the big onslaught of the Internet going commercial, in retrospect - of course that would swamp a library kind of approach out because brochures aren't in libraries. It's ephemera. It was like all librarians are about- some stuff is worth keeping- other stuff is just ephemera. So the original question is, 'Well, what is it?' 'Is there still any sign of life?' Well, there are some guys...

Frana: Cameron Kaiser, do you know him? You mentioned his name in a brief interview for Minnesota Public Radio.

McCahill: Yes. So there are still some guys poking at this and you know, doing that is fun. Now the UNIX Gopher stuff is really officially GPL licensed, we put it out there. We have a new GPL license on it, and there were some guys who were looking at cleaning it up a little bit. So presumably, some people were working on it. That's nice. I know that other projects have been going on for a long time, so it's not one of those things we go, 'Huh, they are using it, that's interesting.' Every once in a while I'll go back and run a client to remember how fast that stuff was.

Frana: I bet it is still much faster than browsers.

McCahill: Because you are not moving much information.

Frana: And it always will be.

McCahill: Yes. Yes. So, in the grand scheme of things it is probably a case of, the problem we were trying to solve, which came out of that- the tail end of the Internet as a research world thing. We're of course going to come at this from a library research information system point of view. What happened after that was, the rest of the world stumbled into it. And when the rest of the world gets there then there is going to be a lot of commerce, there's going to be a lot of brochures, and that sort of stuff- ephemera- and that is going to be the bulk of what's out there, because that's commercial activity. Libraries are not that big of a part of everyday life. They are a nice thing to have around, but that's not where the bulk of the action is in the world. Probably it never will be.

Frana: That's kind of a niche, yes. Do you have other amazing revelations? You said at the outset that you'd share never before reported anecdotes and stories.

[laughter]

McCahill: At the first IETF Farhad and I went to-

Frana: What does IETF stand for?

McCahill: Internet Engineering Task Force.

Frana: Oh, okay.

McCahill: We ended up going to this because- we put together Gopher and some people were starting to use it and we got this e-mail from Joyce Reynolds who works at the Internet Assigned Numbers Authority, IANA the guys who decide which port number you are supposed to use for e-mail or Web or Gopher. Her e-mail was worded in kind of a funny way. It gave us pause... It basically said, 'Hi, I am from the Internet Assigned Numbers Authority and you are using a port number that you shouldn't be using for your protocol. I encourage you to...' - she didn't say it this way, but- '...get your act together and maybe document this a little bit and then ask for a port number so you can get a real one instead of squatting on wrong numbers.' And we went, 'Oahhhh...'

Frana: Broadcasting on the wrong channel!

McCahill: They really liked that, yes.

Frana: You were using port 70?

McCahill: No we weren't. I can't remember what port we were using. Some port we picked out of the air. So, we wrote up the protocol a little bit and sent that in. And then we were encouraged by Anders Gilner, whom I hadn't met at this point, this guy from Sweden who's a good friend of

Joyce who was very interested in this Gopher thing saying I think to Joyce, 'Hey we should hassle these guys a little bit and get them in the fold.' We were encouraged that, 'You know you guys could go to this Internet Engineering Task Force meeting and do a little, you know, twenty minute presentation about what this thing is.' So, I ended up going- basically, it was the equivalent of something like the FCC which was saying, 'Guys, I don't think you should be doing that, maybe you'd like to get a license first.' So I got to go to that meeting, and right before the presentation I was like, 'Well, Farhad, do you want to do this talk or should I?' Farhad wouldn't. So I had to go and do say, twenty minutes of, 'Here, Gopher protocol looks a lot like the finger protocol, only kind of pumped up- it is carrying more information, you know, ...thank you.' They gave us a nice low port number- that was cool. And I got to meet John Postel - that was cool, and meet Anders Gilner, who's a good friend now, and Joyce Reynolds and that whole crowd. Late in that meeting, Tim Berners- Lee was there too, and he showed Farhad and me how this Web thing worked on the NeXT machine, because I think that was all it was working on at that point.

Frana: It was the first time you had seen it?

McCahill: Yes. So we looked at it and he said, 'Guys we should really work together on all this, we should try to find a way to meld these together.' We said, 'Well, okay, we'll think about it.' We went back and tried to make sense out of the documentation and the whole Web thing, but Tim had put most of his documentation up as HTML documents and I was having a hard time making sure I had read the whole thing because it didn't have any real structure. Well, after a couple of weeks of that, Farhad and I said, 'Well, maybe, but I don't think we can figure this out right now...'

[laughter]

McCahill: And we worked on Gopher stuff instead.

Frana: That's a great story!

McCahill: Which was the right decision to make for the next year and a half or so, but in retrospect, may have been the wrong decision to make in the grand scheme of things.

[laughter]

McCahill: Perhaps we should have worked harder to figure out Tim's stuff and meld the two. Because what we would have ended up with, if we had melded the two properly, is the thing I really ended up admiring - the equivalent of Hyper-G which was hierarchy for the documents, full-text search to find the documents and documents that have hypermedia links in them. It wouldn't have had what Hyper-G has with the two-way links, but we would have had a lot of what they had, if we would have been smart enough to realize it at that time. But we weren't smart enough to figure that out, we were kind of distracted.

Frana: Well, but did you thrive with that kind of naiveté, you know- we're stumbling around, there is all this architecture out there, all these authorities out there we don't know about, but we are going to do it anyway?

McCahill: Well, what was cool about that time was that nobody knew what was the right thing at all. For quite a long time, Brewster Kahle was pushing full-text search solves all problems, just do it all in WAIS. Put all the documents in WAIS that will do it. You had the old standby-anonymous FTP- supplanted with an index so you could actually find the stuff- the Archie thing. You have this World Wide Web thing going on that had its place, you had Gopher, and there were other contenders at that point. Nobody knew what was going to be the right thing that was going to predominate. So it was fun because you could be some guys from nowhere in the computing world- Minnesota- where Internet computing wasn't a big player really- have an idea... there weren't big barriers to entry because you could put stuff up, you could do guerilla marketing as an announcement, answer some questions when people asked for help, and make a difference. That was a really cool time to be able to do stuff because you didn't have the behemoths swamping the communication channels. Now it is really hard to do interesting stuff for Internet protocols because you've got all the big players squatting on all the communication channels and all the mind-share. So that was a very fun time. And perhaps, mainly because it is a large research university, the University of Minnesota always had a sort of okay-to-pretty-good Internet connection, and had some infrastructure around so if you were in the right place you could play in this game. But the rest of the world then discovered us. But that was cool. It was fun.

Frana: Now how did you come to be here?

McCahill: Here now?

[laughter]

Frana: Okay. I'm sorry. That was too deep and philosophical.

McCahill: I drove my sports car and...

Frana: How did you come to the University of Minnesota and what was your early education like, and experiences? Are you one of those people that used to break into computer labs at night to play with things or are you one of these people who had a friend a bit older and were ten years old and got started? Or are you one of these people who got formal training?

McCahill: Well, basically, I grew up moving a lot. My father worked for an oil company, Conoco. At that point the theory was, 'If you're an executive, you want to move him around and show him different parts of the organization.' They don't do stuff now the way they used to then. Until we moved to Minnesota, the longest I had lived in one place was five years in Denver. I think. Something like that. We moved about every two years- something like that. So we bounced around a lot. I've seen zillions of school districts, up and down the middle of the US. I moved to Minnesota when I was in junior high, I was an eighth grader here- seventh or eighth

grade. And I was a nerd. I was into science big time. When I was a little kid I used to get up really early to launch rockets. I read a ton of science fiction and that stuff. So you know the type. Minnetonka Junior High got a Time-Sharing Terminal- one of those awful little teletypes with the paper tape. And because I had been reading all this science fiction and I am a science nerd and then doing little science-y projects whenever they'll let me, just playing with science stuff because I think that's cool. 'A computer! Well! I'll have to spend some time with this.' There was a math teacher who was encouraging us little nerdy kids to do nerdy stuff- I mean it is what you should be doing. So he let a couple of us loose on the Time-Sharing Terminal and we were writing dumb little BASIC programs in like seventh or eighth grade- I think eighth grade at that point. The early positive feedback that got me very interested in this is I was taking a math course and the math teacher didn't know how to do stuff with computers. And one of the reasons they'd got computers was to expose kids a little bit to this stuff. So I got to teach math class for one or two sessions because I could program in BASIC- twenty line programs and that kind of stuff, graphs...

Frana: Right.

McCahill: After a while though, the party ended because we were burning through too much time.

[laughter]

They didn't realize that if you are charging connect time, those kids in eighth grade can be very single minded about things- they could burn through a lot of time really quickly. And once we figured out we could store our stuff on this drum storage they had, then the charges went through the roof. So then they clamped down and we didn't get so much access anymore. I did a little bit of stuff; again when I was in high school with Time-Sharing Terminals, but I wasn't so into it because there wasn't much access to be able to play with it at that point, because the school district had learned. We taught them too well early on. In high school though there was a little community of guys who were writing programs just to show off to each other. One of the mean tricks you could do since they were letting us store the programs on something besides paper tape if they were not too big, is store a really lame program with some bugs and have the other guys name on it, to make the other guy look bad.

[laughter]

So if there is a guy you didn't like had this really lame program stored under his name and then he'd say, 'No it's not mine, I'm not that stupid.' So, high school there was a little bit going on with computing, but not too much. I get real interested in chemistry. And then it's time to go to college- and where I could afford to go that had enough technology to be interesting to me, was the University of Minnesota. Because of tuition, and they had big labs and that stuff. I looked at some of the liberal arts schools and looked at their chemistry labs and said- their chem labs suck. My high school's lab was better. Forget it. So I came to the U, then I got completely turned off with computing because there was a FORTRAN course that you had to take where you had to do punch cards – punch cards suck. I think I did okay in the class but I hated it, because sitting and

typing stuff on those keypunches and then submitting a deck to the operator. Boy, you had to figure out how to use a 1004 card reader. You know, I never figured out how to use it- I didn't want to- it just- it wasn't interactive. There's no fun in doing that. So, then I ruled out that crap. I decided chemistry was definitely the way to go, I could get a degree in it. English wasn't the way to go- I didn't like English that well. I got a summer job doing - it was part of the National Science Foundation study of river water and sewage treatment plant effluent, - so, analytical chemistry stuff, which was great- as a chemist I should be, I am interested in analytical- that's cool. Those guys had an Apple II, and were starting to think about using the Apple II to do some of the data analysis they were doing on the mainframe. On the mainframe we were doing stuff like sewage treatment effluent analysis- it is kind of a complex mixture. We are doing matches of, well, there's probably these twenty or thirty sorts of chemicals, do a best fit of what the UV spectra of that we've got is versus the UV spectra of each of these things that would help characterize what is in there. We are burning big time dollars doing these fits because it is a lot of number crunching. We've got an Apple II, they are starting to do some of the lightweight data analysis on there. I start figuring out that doing bench chemistry is not as much fun as helping debug the program that is doing the least squares mathematical fits of the complex chemical mixture of the UV spectra and helping try to do something, move off the mainframe so we can spend less money. So I figured out through the summer job and then working for about a year doing just chemistry for this project, that chemistry was fun for thinking- chemistry was not fun for all the dirty beakers to wash and the dangerous chemical stuff that was going on. Computing had all of the thinking- plus, I liked it from way back- and the tools had gotten enough better. It wasn't decks of cards. It was interactive enough that it was fun.

Frana: You could actually see the results.

McCahill: And you could do something about it while you were sitting there, as opposed to, wait twenty minutes and I'll get back to you with the result from your deck of cards. So then, I started taking some more computer science classes. I took a couple that were really good from Marvin Stein, his computer architecture things, and finally understood how computers really worked. So I said to myself- I need a career change. Looking in the paper, the computer guys were getting paid a lot more than the chemists, there's a lot more jobs- how in the hell am I going to make this jump? Take some classes...and then...

Frana: You are also working at this point?

McCahill: Yes. I am working doing some chemistry stuff in the experimental engineering building- half of that was civil engineering; the other half had the computer center in it, strangely enough. Just down the hall from where the lab I was working was where the early early stages of the microcomputer support guys were. And they eventually started looking around for someone to hire who kind of knew some Pascal and had some experience on an Apple II, because they wanted someone to write some programs. So I went and interviewed and said, 'I'm your man.' Human Resources at the U almost didn't forward my application because I put down that I knew UCSD Pascal on an Apple II and the computer center job requisition said you needed to know Apple Pascal on an Apple II. And the HR person didn't realize they were the same thing. So they called me up and asked me, and I got to explain it was the same. I went to the interview and got

the job. I got to write programs full time, and I said, 'This is better.' Since I started at the lowliest of low, at like junior application programmer writing, actually - debugging other peoples stuff and doing maintenance. But I was in the right place because I was doing microcomputer stuff, as opposed to mainframe stuff. I did a little bit of that, but me and another guy in my office, Dave Larson were real microcomputer bigots, even at that point with the lame little Apple II's. I think we had to pull the disk controller cards out once a week and use an eraser to clean the contacts so they'd be stable enough.

Frana: I remember that.

McCahill: Yes. So basically, then I just kind of worked my way up and as the microcomputer stuff just blossomed underneath, I was in the right place at the right time.

Frana: Coming in at the bottom was really an advantage long term.

McCahill: It's an advantage if the whole field explodes underneath you because after a year or two while you were like at the bottom, you know two years more than anybody else and since everything changes all the time it becomes, 'So how did I get here?' Science nerd got a little exposure to computers, until the school district got smart and figured how much we were burning of their money, got turned off by FORTRAN and decks of punch cards, got back into it for data analysis as part of the chemistry stuff I was doing, and then figured out that the thinking part- you get that all the time with computers and you don't have to wash any dishes. You don't get exposed to- if you get the pH of this solution wrong you create cyanide gas.

Frana: Not much chance of being poisoned working with computers.

McCahill: The thing that really turned me off of chemistry was when I was doing that year of analytical stuff on the wastewater. We had to clean these filters out using sodium azide and if you get the pH of that solution wrong, you will make cyanide so you have to keep the pH controlled so it doesn't do the wrong thing. This would make me nervous and sometimes I would get nervous enough that I couldn't decide if the nervousness was nervousness or early stages of cyanide poisoning.

[laughter]

McCahill: And when I'd feel like that I thought, 'This is the wrong place to be, Mark. Do that stuff that pays better and you're not worrying about if you are poisoning yourself.'

Frana: I don't want to take too much of your time, but are you working on Forms Nirvana still or something new ?

McCahill: After Gopher ended, we were depressed for a while, 'Oh, our baby died nobody cares anymore.' Working at the university turned out to be a blessing because the university always has big problems. It had a big problem with wanting to automate some of its business processes. So, five and a half years ago it was time to do something completely different. I took some of the

Gopher team that was still around and we did a document routing and approval system for on-line documents. We moved them around and all that stuff for purchase orders. And because we were being ironic, we said, 'Oh this will be Nirvana if we can get the purchase orders on-line and get them approved. Oh that'll be great.' Actually, though, it solved some serious business problems the U had and it was a great chance to write a cool application that lots and lots of people use that made a difference- maybe not to the whole world, but for this part of the world. I did that.

Frana: And it is more real-time isn't it? What you mean by the business advantages of it. It made the current ledger reflect reality better.

McCahill: It did two things. Your financial documents get processed and into the ledger right away, as opposed to being key punched after two weeks. So you know the balance in the general ledger is probably accurate to within a day, as opposed to there's two weeks plus its slop in there. It also enforced rules when you created the document. It doesn't let you do bogus wrong stuff when you fill the document out. It tells you what you are doing is wrong and it makes you fix it. The third and subtle thing it does - it doesn't let people who are approvers hide, you know who it's waiting for approval. You can rank the people who are approving them slowly to find the slowest approvers and then either get them help or talk to them about why this is a bad thing. I'm aiding that kind of document flow; you have to do it in your organization. There are a lot of benefits. Making approvers more accountable is probably the biggest thing. So, an on-line computer assisted document system- that's Nirvana. We built that thing and that's old news now.

Frana: Has that spread? Do other universities use Nirvana?

McCahill: The University of Wisconsin is using it for some of their grants management stuff and a couple of other universities are poking at it.

Frana: Oh, that would be good. Do we use it for grant management here?

McCahill: Yes, we do. Then there was a second project that was sort of a son of Forms Nirvana that came out about- we started working on it about four years ago. The Electronic Grants Management System. Forms Nirvana was a document you could fill out in one screen. It'll put that up and let you do it. If you are doing an NIH or NSF grant proposal, those things are more like doing taxes. Complex taxes. There we stole the idea that Turbo Tax and MacInTax have, of sort of a directed interview and audit function that will say, 'Hey if you said A to this thing, that implies that you need to have answered this question in this whole other section. I'll jump you over there and help you answer that.' So basically we built a Turbo Tax equivalent for NSF and NIH grant proposals. It helps to make the budget right. So that's another whole project that came out of that thing. EGMS shares a ton of DNA with Forms Nirvana. We basically reused as much code as we possibly could. I'm the boss of about twenty people right now doing those sorts of applications and anything else somebody wants and has money to pay for- a captive consulting shop inside the university- basically just doing consulting bids for inside the U. The big project that we are working on right now that is going to tie all of this stuff together is doing a personalized dynamic portal that will give you a universal To Do list of: here are the Forms

Nirvana documents waiting on you; here are the Web courses, the courses being taught on the web that you are either teaching or taking; and you have specific items that need your attention like- take the quiz; or the student discussions need you to say something; tell me what is going on with the grant proposals you have got; pull out the balances from your accounts in the general ledger tell you how much money is in the bank right now- the equivalent of that. So we are building a portal to pull that sort of stuff across these different enterprise applications into something dynamic, personalized to each person- that's kind of fun.

Frana: Do you have a name for it? Do you call it something?

McCahill: The Portal. This is an interesting one because we are building it off of an open-sourced framework for building these kinds of portals from a company called MetaDot. What we are doing is a little different if you've built in a little middle layer to insulate the MetaDot framework from our enterprise applications. So there's a little middle cache layer that the enterprise applications write to and then the Portal framework can grab things out of there that's to be displayed for each person. You almost have to do this. Otherwise the page would paint too slowly. As you get to the page and it has to hit what five or six different enterprise-wide systems to see what is on the to-do list, and every time the guy hits the page you are going to go and bang those systems? Bad engineering. So basically doing personalized dynamic information systems is the interesting problem right now. This guy I was talking to right before you came; we were talking about it. So in some ways we are still doing information systems. They are just like specialty boutique information systems; one for document routing, one for grant management, and then we are trying to do an over-arching one that can integrate that stuff and give you a personalized newspaper is how I am thinking of it- both external news from, in my case, what do I care about? The Register and SlashDot and Fresh Meat, plus the documents I need to approve today, the WebCT course that I am a student in that I need to take a quiz on, and da da da da da.... So, still information systems, but much more individualized than the old school thing. In some ways I am really writing this because it does feel like what we really should have been building in the first place. The early days of Gopher it was publish information, it's easy to publish- but you publish the same thing to everybody, which is mass market. And mass market is inherently kind of boring. But what's more interesting is published, personalized information tailored to each person. That's what they care about. That's the kind of information system I want. And things like bookmarks that we were putting in Gopher and Web guys put in, are this lame attempt at letting them, the user, customize access to the information. But it is pretty lame. And it is not dynamic- it does not change all the time.

Frana: Pull rather than push?

McCahill: Right.

Frana: Now, this is really hot. People are talking about this all the time on one of the lists I am on. But one of the criticisms I've heard is that people are going to get only what they think they want and they are going to miss out on things they might have got if it was the mass market kind of approach.

McCahill: You can work around that problem by having some randomness thrown in. One of the things I like to do for fun that is not computers is music. And I recently got a really good synthesizer and some of things you can do with a really good synthesizer is build little patches that throw some semi-random stuff in to spice it up a little bit, because if it is all predictable it is boring. The same thing with an information system; if it's all predictable, there's no human randomness happening. It is boring as hell. But it is easy to add that in. You can randomly pick stuff, or let users create content on-line and subscribe to each other's news channels. That would give you plenty of other input from other people.

Frana: Then we would have personalized communities?

McCahill: Yes. That is how you get around that. In fact, one of the guys I met from the Gopher experience was a research guy at Apple (and now at IBM), Tom Erickson, whose wife is in the Architecture department - that is how I met him. Tom's thing has always been that people depend on other people to find stuff in a big way, and we have to be careful about building information systems and computer systems that take the people out of it too much. The more you can put your smart friends into the system- the guys you kind of trust or have interesting ideas throwing things over the fence at you- the better off you are. So the Portal stuff we are trying to do isn't just a universal To-Do list and personalized newspaper; it's also the people interacting with that Portal can publish their own news channel, fill out a form on a page for the news story in your channel. If you think the guy is interesting, subscribe to his channel. Then you'll get his news along with everybody else's. And that will give you the random element.

Frana: Kind of like these radio stations that, you know, Eminem's radio station. You can plug into him or Macy Gray's or.... Are you a big believer in the paperless office?

McCahill: Look at this place!

[laughter]

Frana: I'd say about half and half...

McCahill: Yes, half computer museum, half paper.

Frana: Not bad.

McCahill: It's hard... I gave a talk a couple of weeks ago about grants management and I said paper is a really formidable foe. We are trying to build things that are better than paper, but paper is so powerful in some ways. It is completely free form. You can scribble on it. Any business process you want. When we are trying to build these business systems like document processing, trying to be as good as paper is a major, major challenge and you find out in trying to displace a paper system just how good paper really is. If it needs to go to somebody else, they just hand it to them. It is hard to accommodate that in a clockwork pre-programmed software system. If they want to write a note on it, they just scribble the note on it and they use whiteout. It's hard to beat paper. But paper is hard because it is so flexible; it is hard to control it. Do I believe in a

paperless office? To a point. There are some things where you want to enforce some rigidity on, some business process on. But things like, put your calendar on-line, I hate that with a passion. My calendar is maintained by me, and it is paper.

Frana: Your address book can't disappear by accident if it's written on paper. Unless you lose it, I guess.

McCahill: Yes. And the control issue is another one. But paper is really good for some things. Some of it is used right here- a pile of student job applications. Just looking at that and saying, 'Man, we are not interviewing all of them.' Or if that was all e-mails, I guess I'd say the same thing but I would have hit D a bunch of times.

Frana: Right, because it's not standardized you can almost recognize immediately their strengths and weaknesses.

McCahill: Yes. And there are a bunch of cues in how you organize things that are hard to come up with in computers. There are a bunch of interesting UI things you can play.

Frana: Well, thanks. Thanks, Mark. I appreciate your taking time for this.

McCahill: I think part of the reason I stayed at the U, actually is the still interesting hard projects. Being able to work on something like let's do a CWIS for a whole bunch of people – it's dynamic. That's why I am still at the U. The U has managed to keep just enough interesting projects that are challenging enough around. So when Gopher ended with the grants management stuff, whether you got in big trouble with the National Institute of Health, and you needed to fix it's information systems, I was at the point of being very bored and starting to think I am out of here. There is nothing interesting. When there was a fire drill and they said, 'We have got to build new systems here. You have a shot at doing them, why don't you try it?' And again, I've got a small development team- it's like a sports car- there are some nice niches, a lot of fun. As that stuff is starting to wind down to, 'Let's do the Portal and tie all the information systems together'... That's interesting enough; I'll stick around to do that. As long as there are interesting issues, getting to do work with a reasonable development team, that's about all you can ask for, if you are basically just a software engineer. So here, some place in industry' it doesn't matter that much. If there's interesting problems and if you get to do something that actually gets out in the world, and people use... So that's why I am still here. Okay, that's a good Coda.

Frana: Yes, very nice. Thank you.

End of Interview.