LIBREAS PODCAST #3

David Rosenthal im Interview.

Boris Jacob (Interview, Transkription), Manuela Schulz (Foto)
Sprache/Language: English Spieldauer/Running Time: 23 min 23 sec
Datei/File: mp3/5,35 mb
Bitrate: 32 kBit/s
Aufnahmedatum/Recorded: 12. Juni 2007
Veröffentlicht/Published: 30. Juni 2007
Über/About: Start-ups in Silicon Valley, Publishers need for Page views, preserving Second Life, Format Obsolescence in the WWW

Online: http://www.ib.hu-berlin.de/~libreas/libreas_neu/podcasts/podcast_3/
File: http://www.ib.hu-berlin.de/~libreas/libreas_neu/podcasts/podcast_3/LIBREAS_Interview_David_Rosenthal.mp3

LIBREAS talks with David Rosenthal about Start-ups in Silicon Valley, Publishers need for Page views, preserving Second Life, Format Obsolescence in the WWW

- Transkription zum Interview -

00:11

LIBREAS: Today we have David Rosenthal [http://www.lockss.org/lockss/David_rosenthal] with us from LOCKSS and I would like to ask you David to tell us a little bit about your work at LOCKSS and your field of interest there.

David Rosenthal: Lots Of Copies Keeping Stuff Safe came out of listening to my wife, actually she wasn't my wife then, complain, that in her job, at HighWire Press, which is Stanford's publisher, where unhappy, because librarians insisted on receiving paper, even though the information was available online. And LOCKSS was an attempt to get back in the electronic space to the way paper used to work, where libraries purchased a copy of the journal for the content and kept it for their own use, subject to restriction what they can do with that copy under copyright law. So that they couldn't wholesale copy it and compete with the publisher. But the paper system used to work where libraries purchased a copy and their future access to that content was their responsibility. And with the advent of the web, instead of purchasing the copy, they leased access to the publishers copy. And when they stopped paying the publisher, access went away, even though in the old world they would have bought a copy. And so the librarians wanted to keep having paper. And the publishers wanted to stop killing trees and shipping the paper around the country because by doing that they could save a lot of money. So LOCKSS was explicitly an attempt to have a computer system work like the paper library system used to work, with local copies kept at individual libraries. And it turned out that building the system, although the idea was simple, was interesting engineering. And at that stage I was a very burned-out engineer, having done two successful start-ups in

six years in my fifties. And if you know anything about silicon valley, doing one start-up is enough to get burned-out, doing two in a row, is really a guarantee of getting burned-out. And I thought this would be an easy project that would be nice and relaxing to work on, and wouldn't take very long, because the idea was very simple. And it's now eight years later and I'm still working on it.

03:19
LIBREAS: That was a good idea for the libraries, but how do the publishers see it, that you make copies of their works and archive them?

David Rosenthal: That's a really important question. And that ends up controlling how the design has to work, because like the publishers under US law, the publisher has to give you permission to keep a copy of their content and if the publisher doesn't like what you are doing with the content, you can go to jail, it's actually a crime. And we didn't want to end up in jail, so we build up a system where there two ways in which we reassure the publishers.

The first is that in order to collect any content from them they have to give us permission. The LOCKSS system works like a web-crawler, like Google's web-crawler for example, it goes and follows all the links on the side, collects all the pages that it finds there. But before we do any of that we look for a page that has a statement on it that gives us permission. And we keep that page very carefully along with all the pages that it pointed to. So in order for us to do anything, to keep any copies of the publisher's content, they have to give us permission. And any time they decide not to give us permission we will not collect any more content from them. So if at any time they are uncomfortable, they can take the permission down and collection will stop.

And the other part, which is very important in the web, is that a lot of the business models of web publishing depend on the page-views, knowing how many people have looked in your pages. That's how you get rewarded for advertising. If you are for non-profit that's how you justify your existence to sponsors. Or to your members if you are a learning-society or so, you need to know the page views. So every time someone accesses a copy of the content through a LOCKSS Box, the request is first forwarded to the publisher, so they see all the page views and know if someone at a library is abusing their copy of their content in the LOCKSS Box, because they will see the page views.

And the third thing what makes them more reassure is that the system uses replication, which is caused by having copies at many libraries that have subscribed to those journals, to make each one of the copies more reliable without needing backups. And the publisher worry a lot about back-tapes with their content on them, because people make backup-tapes and they put them in racks in the machine-room and other people come in and take them away, or they send them off to Iron Mountain in remote storage, and Iron Mountain looses them and sends them to another … It's just uncomfortable for publishers to have lots of copies around, they can't keep track of. So the LOCKSS-System helps the publishers to keep track of where every one of the copies is and how people are using it.

07:19
LIBREAS: How many publishers and libraries are taking part in the system?

David Rosenthal: I don't have the numbers on top of my head, but the LOCKSS website at www.lockss.org has up to date lists of the publishers. I believe we currently have about 200 libraries taking part in the LOCKSS network worldwide. About two thirds I think are in the US and the UK and the rest of them scattered around the world. There is a machine here at Humboldt I think and one at Göttingen and there are other machines in Europe and South Africa, Singapore, Hong Kong, but you can see the list on the website [http://www.lockss.org/lockss/Libraries].

08:14
LIBREAS: Now we will talk about scholarly communication. What is the deal with the new web 2.0 things like weblogs, photos and videos how are they preserved, do you also take care of these?

David Rosenthal: This is an interesting area. We are starting to preserve blogs. Blogs are relatively easy pieces of web 2.0 to preserve because of the way that content accumulates in a blog. People put up a post and comments on that post accumulate for a while, and after a while nobody is interested in commenting on that post any longer, or commenting on a subsequent post. So it's an append only type of publishing. You append new posts and to each of these posts you append comments, and after a while appending comments to a post stops and it becomes a read only object. Read only objects are easy to preserve. We're in the middle

of a program to release support for various blogs. I think support fort he first one is in a release that goes out today and it's my blog, because that's a very easy small blog to test with, which is blog.dshr.org [http://blog.dshr.org/].

But web 2.0 is actually a much bigger thing than blogs and going beyond blogs starts to get into very difficult areas. Streaming media content is set up precisely that way in order to avoid people actually downloading a copy of it. That means that you can't really collect it through the web, you have to go to the original publisher and get a copy. The biggest kind of problem is illustrated by things like Second Life. We have actually had requests to preserve journals that are published inside Second Life and you could roughly speaking see how it might be possible to do that. Second Life has open sourced its client, so you could build a special purpose client that just looked at this journal inside Second Life and sucked it out and preserved it outside. But I think what people are increasingly going to want to do is to actually preserve Second Life itself. And that's really an incredible challenge to think of how you might do that. Because the whole world of Second Life is a huge object. It's changing very rapidly. It's the entire business asset of a company that isn't going to want to let you suck out their whole business asset out and preserve it somewhere. And it's also the intellectual property of the stuff that people make inside Second Life is complex. I believe that the Second Life license [http://secondlife.com/corporate/tos.php] means, that you own the content you create inside Second Life, and you can sell it and so on. So I can't preserve your content inside Second Life without your permission. So in order to preserve second Life I would have to have permission from everyone who created stuff in Second Life. This is getting completely impossible to do.

And so to the extend the web 2.0 evolves towards virtual environments like Second Life, preserving things gets to be an incredible difficult problem. And just generally preserving things which are a mashup of diverse web services where these services themselves may be mashups of other services you know nothing about. It gets very difficult. If I had to say the number one thing that I was worried about digital preservation at the moment, it's that the evolution of the way people want to use technologies is going in a direction that makes preservation almost impossible, for all sorts of reasons.

13:26

LIBREAS: You are also discussing what different kinds of formats you are preserving. When you mentioned Second Life you can also include computer games. Is that also included?

David Rosenthal: We don't preserve computer games. Stanford has a big project to preserve computer games. It's a very difficult problem. Particularly the problem of preserving console games for things like PlayStation 3 and XBOX 360 and so on. This is difficult because of the way these games use digital rights management. There is a wonderful book by a guy called Bunnie Huang, who is responsible for breaking the digital rights management on the original XBOX and explaining how he did it. It is an incredibly challenging technical task even to break the first generation DRM. And most libraries would not be comfortable having their preservation strategy depend on something that may or may not be illegal. So one approach to this may be to preserve the disc of the game and the box that played it and put them in a refrigerator and hope that in twenty years from now you cold pull them out and they would still work. And there are a number of problems with that, the first is that may of these games now have an online component, so you don't get the full experience of the game, unless the game is currently talking to a server which is owned by the game company and which we are not able to preserve. And the other problem is that actually somebody twenty years from now, somebody who pulls the box out plugs the disc in, and starts playing it, is actually not likely to understand many of the cultural references in the game. And is therefore not likely to get the full experience of playing it, or understand what the full experience was. So what Stanford does is to find highly skilled players of the games and videotape them playing the game, and have them record a commentary of what they where doing and why they where doing it, and what this game meant. To record the game as a cultural event rather than so much as a digital object. Preserving it as a digital object, because of the DRM issues is almost impossible.

16:08
LIBREAS: We already talked about the wide range of digital objects like weblogs and videos, how do you decide what to archive and what not?

David Rosenthal: Our approach is to get everything we can from the website that a reader with a browser could see. We are completely agnostic about what format it's in. We have an automated process that does what a reader would do and records the result. This behaves in many ways like a web-cache. You don't know when you are visiting a website, which of the many caches along the road between your browser and the publisher, the content is actually

coming from, whatever the format it's in. Now for some things this doesn't work, for streaming video it doesn't work. That's a problem we can't do a perfect job. There are other techniques which can, if you persuade the publisher to let you do it that can capture video from the streaming server but they are much more expensive. And one of the main thrusts of LOCKSS is to make reasonably good preservation so cheap, that it is really not a barrier for libraries to do it for themselves. That means that we have to sacrifice some level of perfect preservation. But these systems are engineering products, not abstract science and everybody has to make compromises along the way. And all those compromises are trade-offs between how much it's going to cost and how good the result is going to be. So for simplicity and so we chose the way of collecting everything the browser would see and replaying it to a browser later in the future. And that has some limitations, but it is a very cheep way of doing digital preservation.

18:49
LIBREAS: And how do you assure that you can still read the formats, like proprietary formats like PDF in twenty years or in forty years?

David Rosenthal: There is a paper that we wrote in D-LIB in 2005 [http://www.dlib.org/dlib/january05/rosenthal/01rosenthal.html] which explains our strategy for dealing with formats becoming obsolete. Which is essentially to use the format-negotiation mechanism, that is build into http, to detect that a browser is unable to render a particular format, but we have something stored in, and if that happens invoke a converter, that for this output only, creates a converted copy and send it to the browser. We've demonstrated that this works. And we think this is a reasonable approach to format obsolescence in a web environment, because it's transparent and it leverages mechanisms that are build in the won infrastructure of the web. And also because we think that format obsolescence in a web environment is going to be a very slow process. Because you can't simply remove the ability of browsers to render a particular format. Because there is a lot of reason that support is in the browser is because there is a lot of content out there that people want to read, that is in that format. And you don't have control over the format that it's in, so removing support for a particular format from a browser is a thing that is likely to reduce the market share of your browser. And so people are not going to do it, and it doesn't cost very much to keep the ability to render old formats still alive in the browser. What costs is introducing new formats. That's a market driven process again and people are getting

reluctant to introduce new formats, so for example why is YouTube [http://youtube.com] successful? It's because it's shipping video in JAVA-script effectively and this is leveraging the pre-existing infrastructure of the web. It does not have to get new video players and so on installed in people's browsers, which is a hard thing to do. So the very popularity of the web means that the evolution of formats is slowing down. And it means that there is a huge incentive for publishers to publish material in formats that are already popular. And I have some argument about that on my blog [http://blog.dshr.org/2007/05/format-obsolescence-prostate-cancer-of.html], but basically the argument is, that the concern about format obsolescence is inherited from a world that doesn't exist much any more. Even Microsoft, which is the example most people use is finding it difficult to continue using deliberately, introduced format obsolescence as a way of driving its business model, because its customers are starting to complain about the costs that this is imposing on them. And that is a lot of the reason why there is push behind the open document format […], because the externalities of format obsolescence are beginning to really bite people and they are pushing back on the people who are trying to obsolete the formats.

23:16
LIBREAS: Thank you very much for your time and your answers.

David Rosenthal: Thank you, you've good Questions.