

Andreas Vox

Femke Snelting

A USER SHOULD NOT BE ABLE TO SHOOT HIMSELF IN THE FOOT

While in the background participants of the Libre Graphics Meeting 2007 start saying goodbye to each other, Andreas Vox makes time to sit down with us to talk about Scribus, the Open Source application for professional page layout. The software is significant not only to it's users that do design with it, but also because Scribus helps us think about links between software, Free Culture and design. Andreas is a mathematician with an interest in system dynamics, who lives and works in Lübeck, Germany. Together with Franz Schmid, Petr Vanek (subik), Riku Leino (Tsoots), Oleksandr Moskalenko (malex), Craig Bradney (MrB), Jean Ghali and Peter Linnel (mrdocs) he forms the core Scribus developer team. He has been working on Scribus since 2003 and is currently responsible for redesigning the internal workings of its text layout system.

This weekend Peter Linnel presented amongst many other new Scribus features ¹, 'The Color Wheel', which at the click of a button visualises documents the way they would be perceived by a colour blind person. Can you explain how such a feature entered into Scribus? Did you for example speak to accessibility experts?

I don't think we did. The code was implemented by subik ², a developer from the Czech Republic. As far as I know, he saw a feature somewhere else or he found an article about how to do this kind of stuff, and I don't know where he did it, but I would have to ask him. It was a logic extension of the colour wheel functionality,

because if you pick different colours, they look different to all people. What looks like red and green to one person, might look like grey and yellow to other persons. Later on we just extended the code to apply to the whole canvas.

It is quite special to offer such a precise preview of different perspectives in your software. Do you think it is particular to Scribus to pay attention to these kind of things?

Yeah, sure. Well, the interesting thing is... in Scribus we are not depending on money and time like other proprietary packages. We can ask ourselves: Is this useful? Would I have fun implementing it? Am I interested in seeing how it works? So if there is something we would like to see, we implement it and look at it. And because we have a good contact with our user base, we can also pick up good ideas from them.

There clearly is a strong connection between Scribus and the world of prepress and print. So, for us as users, it is an almost hallucinating experience that while on one side the software is very well developed when it comes to .pdf export for example, I would say even more developed than in other applications, but then still it is not possible to undo a text edit. Could you maybe explain how such a discrepancy can happen, to make us understand better?

One reason is, that there are more developers working on the project, and even if there was only one developer, he or she would have her own interests. Remember what George Williams said about FontForge... ³ he is not that interested in nice Graphical User Interfaces, he just makes his own functionality... that is what interests him. So unless someone else comes up who compensates for this, he will stick to what he likes. I think that is the case with all Open Source applications. Only if you have someone interested and able to do just this certain thing, it will happen. And if it is something boring or something else... it will probably not happen. One way to balance this, is to keep in touch with real users, and to listen to the problems they have. At least for the Scribus team, if we see people complaining a lot about a certain feature missing... we will at some point say: *come on, let's do something about it*. We would implement a solution and when we get thanks from them and make them happy, that is always nice.

Can you tell us a bit more about the reasons for putting all this work into developing Scribus, because a layout application is quite a complex monster with all the elements that need to work together... Why is it important you find, to develop Scribus?

I use to joke about the special mental state you need to become a Scribus developer... and one part of it is probably megalomania! It is kind of mountain climbing. We just want to do it, to prove it can be done. That must have been also true for Franz Schmid, our founder, because at that time, when he started, it was very unlikely that he would succeed. And of course once you have some feedback, you start to think: *hey, I can do it... it works. People can use it, people can print with it, do things ... so why not make it even better?* Now we are following InDesign and QuarkXpress, and we are playing the top league of page layout applications ... we're kind of in a competition with them. It is like climbing a mountain and then seeing the next, higher mountain from the top.

In what way is it important to you that Scribus is Free Software?

Well... it would not work with closed software. Open software allows you to get other people that also are interested in working on the project involved, so you can work together. With closed software you usually have to pay people; I would only work because someone else wants me to do it and we would not be as motivated. It is totally different. If it was closed, it would not be fun. In Germany they studied what motivates Open Source developers, and they usually list: 'fun'; they want to do something more challenging than at work, and some social stuff is mentioned as well. Of course it is not money.

One of the reasons the Scribus project seems so important to us, is that it might draw in other kinds of users, and open up the world of professional publishing to people who can otherwise not afford proprietary packages. Do you think Scribus will change the way publishing works? Does that motivate you, when you work on it?

I think the success of Open Source projects will also change the way people use software. But I do not think it is possible to foresee or plan, in what way this will change. We see right now that Scribus is adopted by all kinds of

idealists, who think that is interesting, lets try how far we can go, and do it like that. There are other users that really just do not have the money to pay for a professional page layout application such as very small newspapers associations, sports groups, church groups. They use Scribus because otherwise they would have used a pirated copy of some other software, or another application which is not up to that task, such as a normal word processor. Or otherwise they would have used a deficient application like MS Publisher to do it. I think what Scribus will change, is that more people will be exposed to page layout, and that is a good thing, I think.

In another interview with the Scribus team ⁴, Craig Bradney speaks about the fact that the software is often compared with its proprietary competition. He brings up the ‘Scribus way of doing things’. What do you think is ‘The Scribus Way’?

I don’t think Craig meant it that way. Our goal is to produce good output, and make that easy for users. If we are in doubt, we think for example: InDesign does this in quite an OK way, so we try to do it in a similar way; we do not have any problems with that. On the other hand... I told you a bit about climbing mountains... We cannot go from the one top to the next one just in one step. We have to move slowly, and have to find our ways and move through valleys and that sometimes also limits us. I can say: *I want it this way* but then it is not possible now, it might be on the roadmap, but we might have to do other things first.

When we use Scribus, we actually thought we were experiencing ‘The Scribus Way’ through how it differences from other layout packages. First of all, in Scribus there is a lot more attention for everything that happens after the layout is done, i.e. export, error checking etc. and second, working with the text editor is clearly the preferred way of doing layout. For us it links the software to a more classic ways of doing design: a strictly phased process where a designer starts with writing typographic instructions which are carried out by a typesetter, after which the designer pastes everything into the mock-up. In short: it seems easier to do a magazine in Scribus, than a poster. Do you recognize that image?

That is an interesting thought, I have never seen it that way before. My

background is that I did do a newspaper, magazine for a student group, and we were using PageMaker, and of course that influenced me. In a small group that just wants to bring out a magazine, you distribute the task of writing some articles, and usually you have only one or two persons who are capable of using a page layout application. They pull in the stories and make some corrections, and then do the layout. Of course that is a work flow I am familiar with, and I don't think we really have poster designers or graphic artists in the team. On the other hand ... we do ask our users what they think should be possible with Scribus and if a functionality is not there, we ask them to put in a bug report so we do not forget it and some time later we will pick it up and implement it. Especially the possibility to edit from the canvas, this will approve in the upcoming versions.

Some things we just copied from other applications. I think Franz ⁵ had no previous experience with PageMaker, so when I came to Scribus, and saw how it handled text chains, I was totally dismayed and made some changes right away because I really wanted it to work the way it works in PageMaker, that is really nice. So, previous experience and copying from another applications was one part of the development. Another thing is just technical problems. Scribus is at the moment internally not that well designed, so we first have to rewrite a lot of code to be able to reach some elements. The coding structure for drawing and layout was really cumbersome inside and it was difficult to improve. We worked with 2.500 lines of code, and there were no comments in between. So we broke it down in several elements, put some comments in and also asked Franz: *why did you did this or that*, so we could put some structure back into the code to understand how it works. There is still a lot of work to be done, and we hope we can reach a state where we can implement new stuff more easily.

It is interesting how the 2.500 lines of code are really tangible when you use Scribus old-style, even without actually seeing them. When Peter Linnel was explaining how to make the application comply to the conservative standards of the printing business, he used this term 'self-defensive code' ...

At Scribus we have a value that a file should never break in a print shop. Any bug report we receive in this area, is treated with first priority.

We can speak from experience, that this is really true! But this robustness shifts out of sight when you use the inbuilt script function; then it is as if you come in to the software through the backdoor. From self-defence to the heart of the application?

It is not really self-defence ... programmers and software developers sometimes use the expression: 'a user should not shoot himself in the foot'. Scribus will not protect you from ugly layout, if that would be possible at all! Although I do sometimes take deliberate decisions to try and do it ... for example that for as long as I am around, I will not make an option to do 'automatic letter spacing', because I think it is just ugly. If you do it manually, that is your responsibility; I just do not feel like making anything like that work automatically. What we have no problems with, is to prevent you from making invalid output. If Scribus thinks a certain font is not OK, and it might break on one or two types of printers ... this is reason enough for us to make sure this font is not used. The font is not even used partially, it is gone. That is the kind of self-defence Peter Linnel was talking about. It is also how we build .pdf files and PostScript. Some ways of building PostScript take less storage, some of it would be easier to read for humans, but we always take an approach that would be the least problematic in a print shop. This meant for example, that you could not search in a .pdf. ⁶ I think you can do that now, but there are still limitations; it is on the roadmap to improve over time, to even add an option to output a web oriented .pdf and a print oriented .pdf ... but it is an important value in Scribus is to get the output right. To prevent people to really shoot themselves in the foot.

Our last question is about the relation between the content that is layed out in Scribus, and the fact that it is an Open Source project. Just as an example, Microsoft Word will come out with an option to make it easy to save a document with a Creative Commons License ⁷. Would this, or not, be an interesting option to add to Scribus? Would you be interested in making that connection, between software and content?

It could well be we would copy that, if it is not already been patented by Microsoft! To me it sounds a bit like a marketing trick ... because it is such an easy function to do. But, if someone from Creative Commons would ask

for this function, I think someone would implement it for Scribus in a short time, and I think we would actually like it. Maybe we would generalize it a little, so that for example you could also add other licenses too. We already have support for some meta data, and in the future we might put some more function in to support license managing, for example also for fonts.

About the relation between content and Open Source software in general ... there are some groups who are using Scribus I politically do not really identify with. Or more or less not at all. If I meet those people on the IRC chat, I try to be very neutral, but I of course have my own thoughts in the back of my head.

Do you think using a tool like Scribus produces a certain kind of use?

No. Preferences for work tools and political preference are really orthogonal, and we have both. For example when you have some right wing people they could also enjoy using Scribus and socialist groups as well. It is probably the best for Scribus to keep that stuff out of it. I am not even sure about the political conviction of the other developers. Usually we get along very well, but we don't talk about those kinds of things very much. In that sense I don't think that using Scribus will influence what is happening with it.

As a tool, because it makes creating good page layouts much easier, it will probably change the landscape because a lot of people get exposed to page layout and they learn and teach other people; and I think that is growing, and I hope it will be growing faster than if it is all left to big players like InDesign and Quark... I think this will improve and it will maybe also change the demands that users will make for our application. If you do page layout, you get into a new frame of mind ... you look in a different way at publications. It is less content oriented, but more layout oriented. You will pick something up and it will spread. People by now have understood that it is not such a good idea to use twelve different fonts in one text ... and I think that knowledge about better page layout will also spread.

1. http://wiki.scribus.net/index.php/version_1.5.4%2B-New_Features
2. Petr Vanek
3. *I think the ideas behind it are beautiful in my mind*
4. <http://www.kde.me.uk/index.php?page=fosdem-interview-scribus>
5. Schmid
6. because the fonts get outlined and/or reencoded
7. <http://creativecommons.org/press-releases/entry/5947>

COPYRIGHT © CONSTANT 2014 — COPYLEFT: This is a free work, you can copy, distribute, and modify it under the terms of the Free Art License. *Trademarks, quoted material and linked content are copyrighted according to their respective owners.* ENJOY!