open development

by thorsten

Table of contents

1. tarGET-X and open developement

This site is created with Apache Lenya and generated with Apache Forrest.

1.1. Recent changes

I started to work on this site again while testing a new forrest feature called the "dispatcher". I very much enjoyed using it for this new version of tarGET-X. The dispatcher allows you to create and change the design within minutes. Further the dispatcher is very flexible and extensible you can turn on contracts, add external data and all in runtime. Through the clear seperation of content, style and logic I finally can reuse all the stuff I already had on my site but with a different layout (whenever and wherever I want).

The dispatcher gives me the flexibility that I was looking for regarding design and url specific layout, but I am tiered of editing the forrest content in the xml editor and want to do it via a web based system. I just added some initial code in lenya and forrest to start the DOCO publication.

1.2. early days

The problem that I had in all this years: I did not found a single good tool to generate this page. After my first versions of pure html I tried the wyona cms to edit and generate this page. The problem was not only that the cms was **really complicated** but as well **total overkill** for a personal homepage back then.

1.3. Apache Forrest

I started to use forrest in 2003 because Apache Lenya is using it to document their project. When I updated my forrest version to build the lenya documentation I was runnning into heaps error. They had been related to a custom skin we used in lenya and the changes in forrest.

Since forrest were looking for a new all css based skin and lenyas custom skin was nearly css based, I developed the initial "pelt" skin in forrest out of the lenya skin and some others. This skin (still default in 2006 in forrest) is quite customizable but hard to extend.

The problem with forrest skins so far has been that if only the design structure changed (html-skeleton) or just some extra content in some pages, the user had to write a completely new skin and implement all functionality.

1.4. Dispatcher

While leading the development of the dispatcher in forrest I started to create the contracts out of the old skins and adopted the design with use cases of forrest devs. The dispatcher is following the concepts I have learned about web based systems. The aim of the dispatcher concept is to provide a flexible framework for creating site and page specific layout for different formats using different data sources through an advanced separation of concerns. The dispatcher was first implemented in Apache Forrest, but it can be used in any cocoon based application.

1.5. DOCO

Still in the end I need to manage the content of the site and I want to do this online. That brings me back to a CMS. I am right now playing around with DOCO. Using a Apache Lenya publication to manage a forrest project. Using the dispatcher to manage the design.

2. about open development

open development means playing an active role in open source development. The most important point when using free software is to get involved into the communities around the software. I am very grateful that I am part of a community, people testing your code and pointing out sharp corners, that helps a lot.

Thanks very much everyone who is active part of an open source community!

2.1. get involved - help your neighbor

As professional developer you can give back feedback and enhancements in many ways. You can send documentation and code enhancements to the issue tracker and mailing list of this community. If you are not giving something back to the community (where you could), the software, you use, will die soon. Open source is more about communities then code! Consider problems that you may solve today for a foreigner would have been tomorrow your problem anyway.

2.2. secure your freedom

On the other hand you can secure the health of the product and the community by being active part of the project. The benefit for yourself is that you will not ending up paying (again) a solution provider on which you are (again) depending on (patches from the provider). When code enhancements and/or bug fixes (patches) are going back into the core software you secure your freedom.

Create patches, be part of the community and secure your freedom.

2.3. word of advice for professional user

Clients should try to always choose a solution provider that is involved in open development and can secure the usability of the software. The client will benefit from lower cost in the future and greater freedom. Hey, after all free software is a "matter of the users' freedom". ;)