

Karsten Dambekalns

TYPO3 5.0

The Road to Phoenix

Since work has begun on TYPO3 5.0 at the TYPO3 Developer Days 2006, steady progress has been made on the design and code. Thus the recently concluded Developer Days 2007 presented an occasion for looking back as well as marking the beginning of further interesting developments.

In the meantime people outside of the TYPO3 and PHP world have been taking notice of the project. This is demonstrated not only by the discussions on the mailing lists of numerous other projects, such as phpGACL, eZ Components and eGroupware, but also by David Nuescheler's visit to this year's Developer Days.

The Status Quo

Until now, the work has been focused on the architectonic foundations of the new system. Alongside numerous smaller gatherings, a meeting was held in Berlin, at which architectonics and development methods were discussed in a relaxed atmosphere for two days. As the last '5.0 article' (T3N No.6) showed, the creation of a new, future-proof foundation is one of the most important reasons for the new development. In the last six months significant progress has been made in this area, proving the efficacy of the thorough approach taken.

Presentation	View
	Controller
Domain	Application Logic (Service Layer)
	Domain Model (Domain Layer)
Data source	Data Mapper
	Data Source Abstraction

The TYPO3 5.0 architecture consists of layers, which allow a clear separation of the individual code parts.

The recourse to well established development patterns proved itself useful by above all allowing the developers to speak the same language. Techniques such as domain driven design [1] soon made clear that many peripheral arenas, commonly designated as infrastructure, must be reconfigured alongside the development of the actual CMS. Moreover, other techniques (which are common in other programming languages) were converted into the PHP programming, such as aspect-orientated programming and inversion of control. Although these techniques are just making their entry into the PHP world, some of them are already 20 years old. Another pillar of the development is the principle of continuous integration [2]. For many of the problems, individual solutions (e.g. Mojavi, Seasar, or GAP) already exist, as do frameworks such as symfony. But these solutions are not effective enough, or are no longer (sufficiently) maintained, or have not yet been released officially. Furthermore, we did not want to impose Japanese code documentation upon the TYPO3 developers.

One Project; Two Products: CMS and Framework

Following hard upon these realisations was the development of an individual framework. In this way, a high-quality, consistently programmed and documented solution may be found.



Breakfast in Berlin, before the meeting: Sebastian Kurfürst, Karsten Dambekalns, Ronny Unger, Christian Jul Jensen and Robert Lemke (from left to right).

In the end the users benefit too because they need not acquire six, eight or even more APIs, programming styles and libraries. Because the framework itself is quite comprehensive without containing TYPO3-specific code, the division of labour, as announced at the Developer Days 2007, was the next logical step. In future, there will be two products: 'TYPO3 CMS' and 'TYPO3 Framework'. A first release of the TYPO3 framework is planned for the TYPO3 Conference 2007 at Karlsruhe (the roadmap under [3] is updated constantly).

Many parts of the framework are already complete. In the coming months work will focus above all on the completion and improvement of the code foundation. Hitherto the framework has made available functions in the following domains:

- Aspect-oriented programming
- Dependency injection (inversion of control) and component-management dependency
- Expanded reflection functions
- Package management
- Utility functions
- Model view controller

Since the support for the model view controller is still being developed, at the moment it is showing the most dramatic changes. By contrast, the component management and the AOP support already have a stable set of functions, so that the preparation of framework snapshots has been planned for the near future. In the long term we hope that other projects use the framework too and that feedback as well as maintenance of the framework occurs outside the TYPO3 community. But this is all still up in the air: at the moment the division serves as an aid for smooth development. It ensures that the developers make a clear distinction between work on the CMS and the infrastructure.

The Content Repository

Completely freeing the saving of files in TYPO3 from the design of the relational databases was an early idea; soon the use of a content repository with a JSR 170-compatible interface came about.

We are continuing to pursue this solution and in doing so enjoy prominent support. David Nuescheler of Day Software, the leading light of the groups experimenting with the JCP standards JSR-170 and JSR-283 (the successor of JSR-170), offered his support and gave a well-received talk at the TYPO3 Developer Days 2007. At the same time, he clearly reiterated the benefits offered by a content repository with a standardised interface. The presentation slides of all talks relating to TYPO3 5.0 can be found in the project domain under [4].

The planned use of a JSR-170 compatible content repository is not only technically interesting: the API was standardised in a Java community process, and yet, as David Nuescheler emphasised, it is no 'java API'. And so we are presented with the wonderful opportunity of bringing the hitherto strictly separated worlds of PHP and Java closer together.

The development of our native PHP-implemented content repository has not yet commenced, but some research and rough planning has taken place, and a team will be assembled in the coming weeks.

The People Behind TYPO3 5.0

The team behind TYPO3 5.0 is forever growing and changing – and should continue to do so in the future. If you would like to help pave the way for TYPO3, just drop us a line. Right now the team has the following members:

Robert Lemke is the head of the team and is responsible for keeping all the balls in the air. His love of beautiful and clean code is a great asset in this role, especially when combined with his insatiable appetite for technical literature and documentation.

Karsten Dambekals also plays a part in the development of the architectonic foundations and will direct the implementation and integration of a PHP-based content repository. In addition he is responsible for the technical foundations of the doc-book-based project documentation.

Daniel Brun is the newest entrant and with his experience has already contributed a great deal to the discussions about the architecture. We are especially thankful for his cooperation because he has a lot of knowledge outside of the TYPO3 and PHP field.

Sebastian Kurfürst has already made a name for himself as a member of the core team. As a part of the 5.0 team, his help has been apparent in the project's every nook and cranny: from tests, to the development of prototypes for individual ideas, to the maintenance and coordination of the work surrounding the development platform.

Ronny Unger took the (thankless?) task upon himself to port the JSR-170 API unit tests from Java to PHP: he has already ported hundreds of test cases in dozens of tests. In doing so, he has isolated and eliminated memory problems associated with the use of the Java bridge.

Christian Jul Jensen is part of the TYPO3 community bedrock and in future will tend to the soul of TYPO3.

Links and Literature

 *Softlink 1772*

- [1] Domain Driven Design: <http://domaindrivendesign.org>
- [2] Continuous Integration: <http://www.martinfowler.com/articles/continuousIntegration.html>
- [3] Roadmap: <http://typo3.org/teams/5.0-development/roadmap/>
- [4] TYPO3 5.0 Project: <http://typo3.org/gimmefive>