

Dar **Vida** – good for your fitness

Presenting a PHP5 MVC framework



OSCOM 4, Apache Track, September 30, 2004, ETH Zürich

TOC

1. Organisational Aspects

- What is it all about
- Who are the authors
- Why we developed Vida
- Vision, Goals, No-Goals

2. Technical Aspects

- Why in PHP(5)
- Standards
- MVC Basics

3. Project Demo

- The Model
- The View
- Additional features
- Quick tutorial

4. Conclusion

- Roadmap
- Community Building

What is it all about

- First public presentation of Vida
- Vida: a PHP5 Model View Controller framework
- Show history, reason, functionality of the framework
- Mainly show why it is easy to develop on Vida
- How might the future be of Vida?

Who are the authors

- Emanuel Indermühle
 - University of Bern, Computer Science
 - CTO of foresite Systems
 - Developer of foresite PRS (Project Reporting System)
 - Core Developer of Vida
- Matthias Stürmer
 - University of Bern, Business Admin & Computer Science
 - Co-founder of foresite Systems
 - Developer of foresite CMS
 - Coordination, docu, GUI of Vida

Why we developed Vida

- Started with foresite CMS 2 years ago
- Second web application: Project Reporting System
- Other individual web application projects
- All web apps need similar functionalities
- 3 choices:
 1. Going on writing every application from scratch
 2. Use an existing PHP framework
 3. Code a new PHP framework

2. Use an existing PHP framework

1. **php.MVC** www.phpmvc.net

- Good user and developer documentation, includes test classes
- No database abstraction, no detailed rights management
- No use of PHP5 functionality (OO, XML)

2. **Mojavi** www.mojavi.org

- Small framework, basically authentication
- No database abstraction, no form elements
- No documentation, only tutorial

3. **Horde** www.horde.org

- Well organized Open Source Project, many implementations
- Too heavy, initiated on early stages of PHP, incomplete documentation

3. Code a new PHP framework

- History
 - Basic idea of Emanuel
 - Used database abstraction of foresite PRS
 - Developed basic functionality of Vida as Uni project
 - Complete rewriting of View (Containers as in Java) and adding lots of new features
- Motivation of foresite Systems
 - Knowing all functionality of the framework it's easier to build new web applications on top of it.
 - Fun writing an own PHP framework

Vision during framework dev

Web application developers don't have to bother about URLs, sessions, authentication etc. but can focus on writing applications in a more abstract, „clean“ way with few lines of code.

Goals of Vida

1. Offering services which can be used or overwritten
2. Easy to learn and implement for own web applications
3. For small and medium web applications, so far
4. Not too general so functionality speeds up development
5. Same Vida and web app code, different customers

No-Goals of Vida

1. To be specific so few web applications can be developed on top of the framework
2. Rule the world
3. To have a rigid architecture without extensions
4. Stay a 2-person community without criticism

Technical Aspects

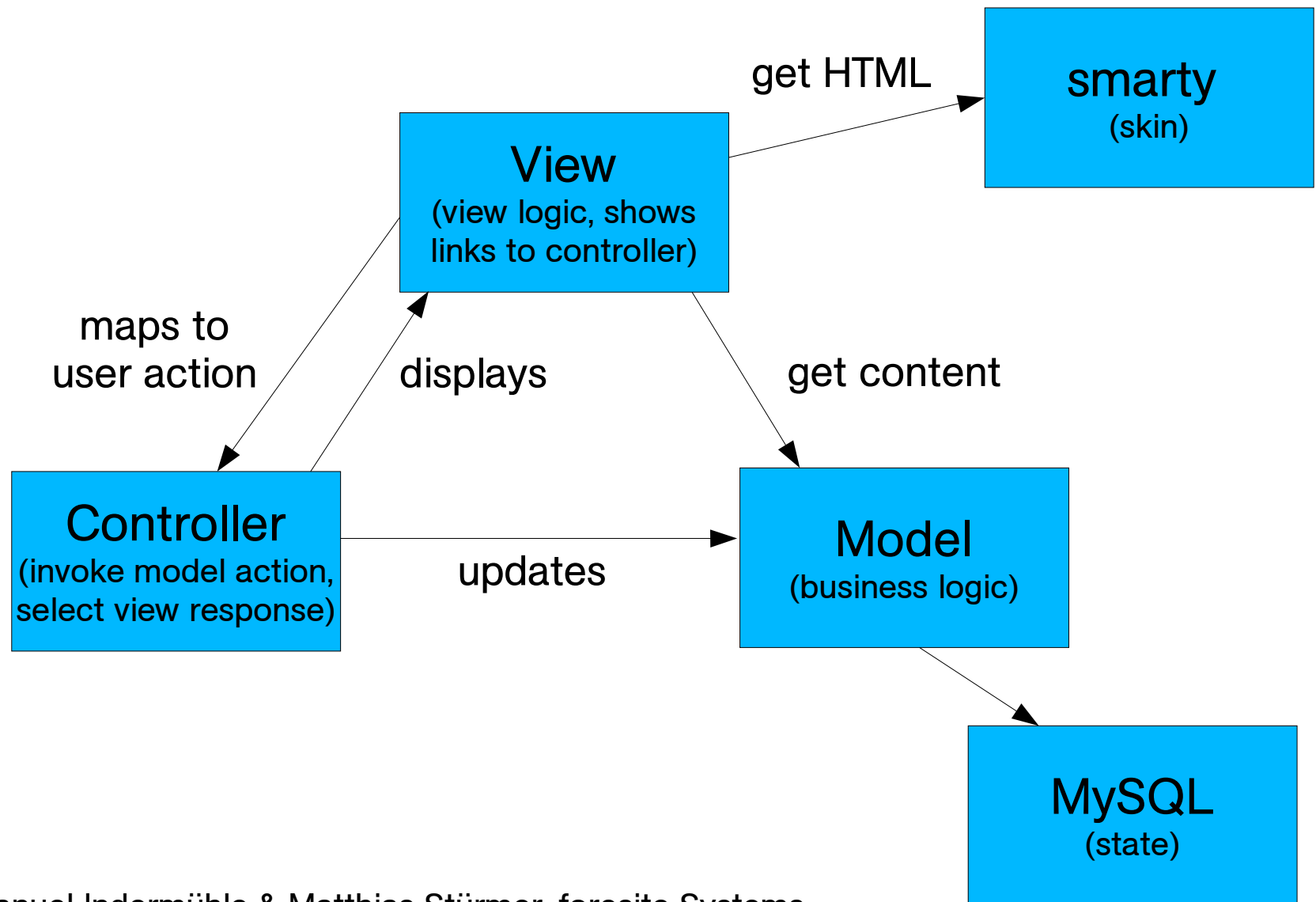
Why in PHP(5)

- PHP as widely used programming language for small and medium web applications (= cheap hosting)
- My skills
- Using new OO features of PHP5
 - Exceptions
 - Proper encapsulation with access levels
 - Static methods

Standards

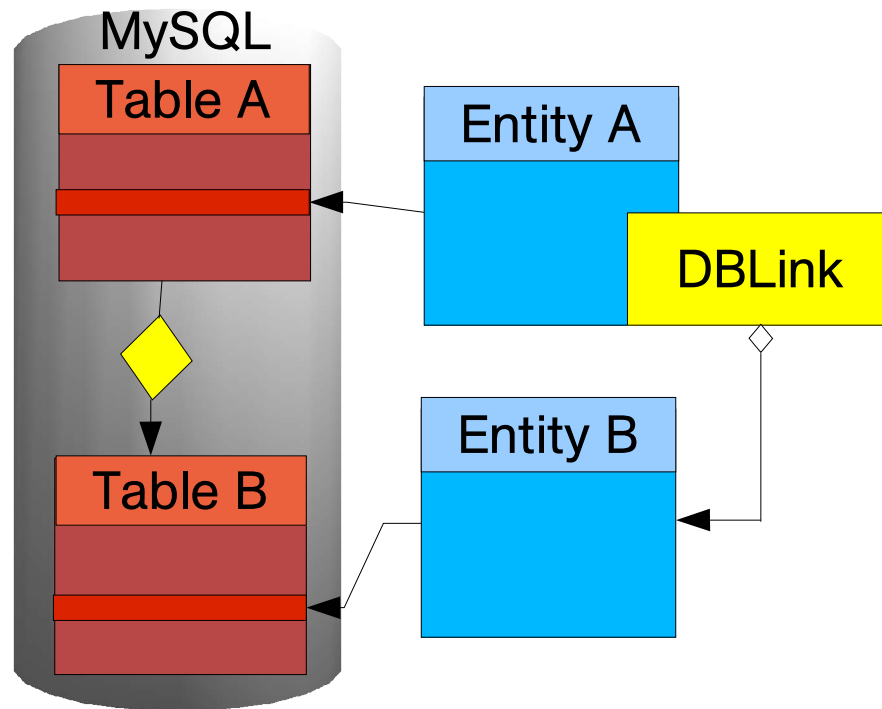
- OO usage
 - Use Vida's functionalities or overwrite them
- Smarty Templates
 - Easy skinable
- XML
 - Language file, rights management

MVC Basics



Project Demo

The Model

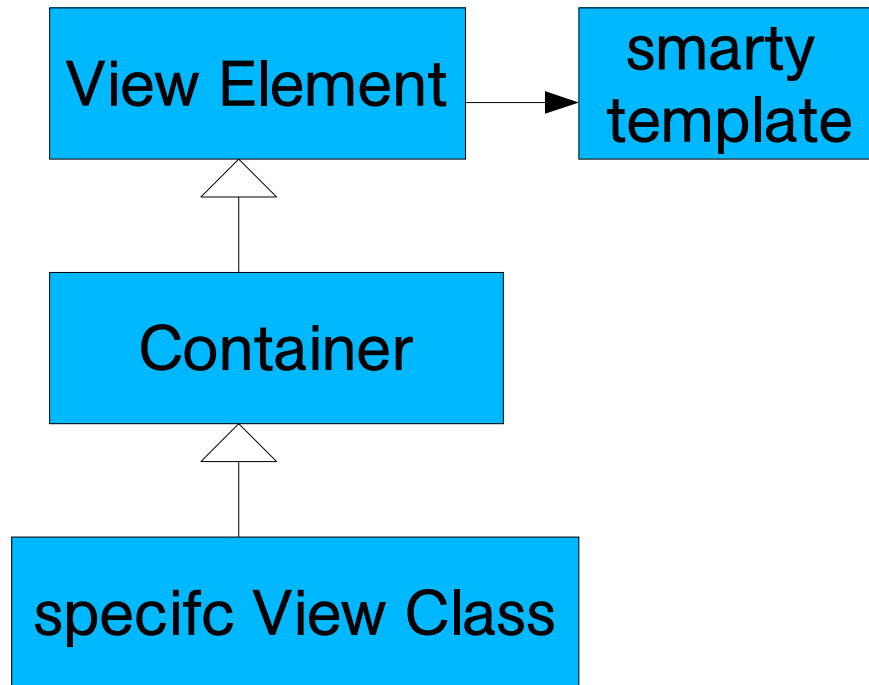


- Is a database abstraction
- Relations between tables are implemented
- Vida provides User/Group Management

Rights Management

- Actions (state changes) of the Model are under controll of a detailed rights management.
- You specify the usergroups that can execute the action only if the defined situations (methods) are true.
- **Example:**
 - Model-type = „contact“
 - action = „delete“
 - usergroup = „secretary“
 - situation1 = „userHasCreatedContact“
 - situation2 = „contactHasNoUserAssigned“

The View



- is a hierarchy of nested objects
- All of them refer to a smarty template
- View logic is OO
- Vida offers a lot of often used, extendable elements
 - filtered lists
 - hierarchical view

Formulars

- The Formular is part of the View that maps to a Controller.
- The Controller works with the Formular input.
- Testing the input to fit the data domain is the Formular's job
- Testing the Input to fit to the Controllers action is the Controller's job
- Vida delivers a lot of complex, though extensible formular elements.

Additional Features

- XML language file – For customizing you can overwrite parts of the language file in the database.
- Database structure versioning – SQL statements to create the database structure can be stored in a directory where they automatically will be executed in the right order.
- Authentication – sends password by email
- Message Management
- Error Management with Exceptions

Quick Tutorial

- How to build a web application with Vida
- Basic design of the Vida API

Conclusion

Roadmap

- 2004-09-30: Vida v0.5 with described features
- 2004-10-31: Vida v0.x with list filters, col selector, db form customizing etc.; foresite CMS implementation started
- 2004-12-20: foresite CMS 2.0beta
- 2005-01-31: Vida v1.0, foresite CMS 2.0

Community Building

- We look for cooperation with PHP developers and companies concerning use and development of Vida.
- Open jobs / where we need you!
 - Co-Developer concerning MVC components
 - „Senior“ Smarty developer
 - New technology developers (WebDAV, LDAP etc.)
 - Support concerning community building process
 - **Users of the framework – developers building web apps with Vida!**
- We are open to any critics, comments, questions...