Dokeos users day - Valence

Dokeos LCMS Design

Jean-Marie Maes & Bart Mollet

Overview

- Why does Dokeos need improvement?
- What can an LCMS do for us?
  - What is it?
  - Why is it better?
  - How do we implement it?
- How does the new stuff look?

Dokeos's Shortcomings

- From a user perspective
  - No exchange of information between courses
  - No search feature
  - Multiple upload of objects if used in more than one course or group

A Proposed Solution: LCMS

- Concept
  - Content Management System
  - *Learning* CMS
- Advantages
- Implementation

Content Management System

- Centralized
- Collaborative
- CRUD
  - Create
  - Read
  - Update
  - Delete

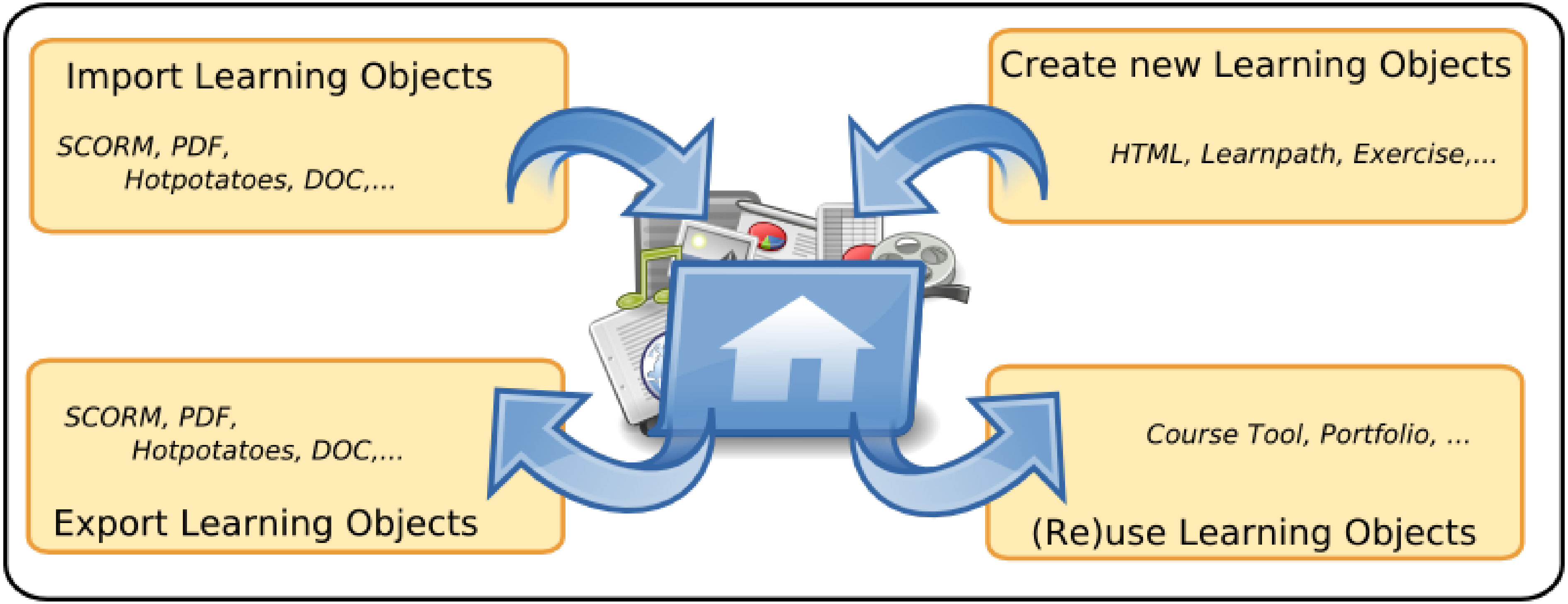
Learning CMS

- CMS + learning objects = LCMS
- Learning object
  - Owned by user, not course
  - Published in one or more courses, portfolios, …
- Competitors taking similar approach

LCMS Advantages

- User-oriented architecture
  - Learning objects may be published in several courses, portfolios, …
  - Personal learning object repository
  - Sharing learning objects between users
- Fully modular learning object architecture
- Generalization ⇒ fast & easy searching

Central Object Repository



Import Learning Objects

SCORM, PDF, Hotpotatoes, DOC,...

Create new Learning Objects

HTML, Learnpath, Exercise,...

SCORM, PDF, Hotpotatoes, DOC,...

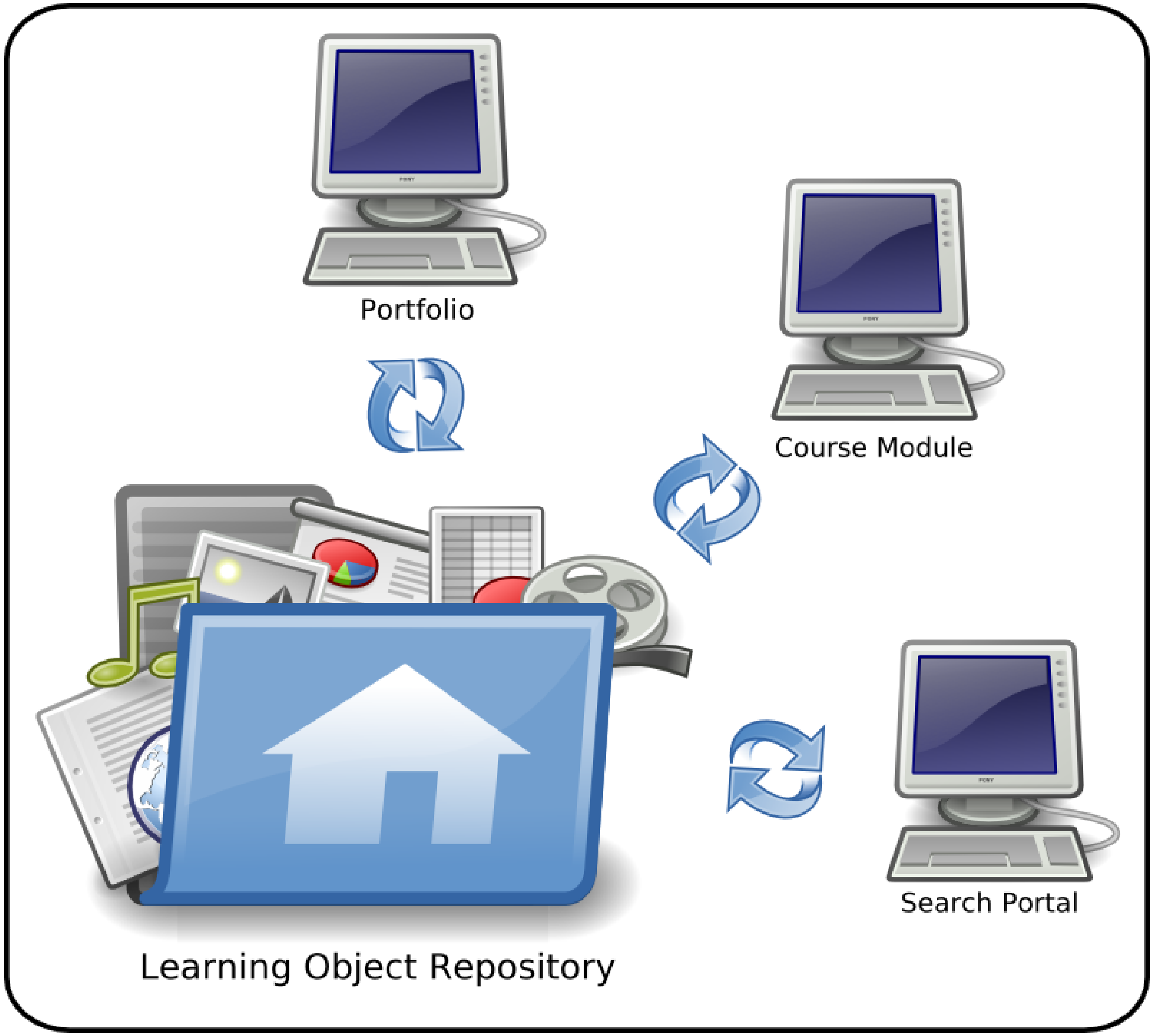
Export Learning Objects

Course Tool, Portfolio, ...

(Re)use Learning Objects

Applications using the Object Repository

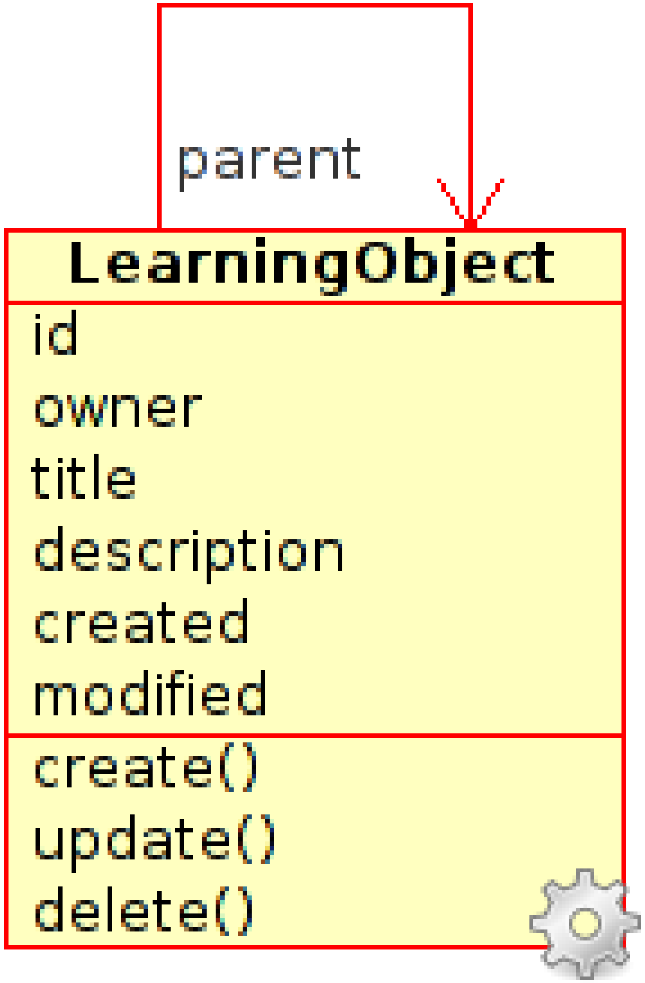
- Applications
  - Portfolio
  - Course Management
  - Search Portal
- Using central Object Repository
  - Local or remote
  - One or multiple



Dokeos's Shortcomings

- From a developer perspective
  - Code duplication
  - Layering issues
  - Consistency not on par

Basic Learning Object



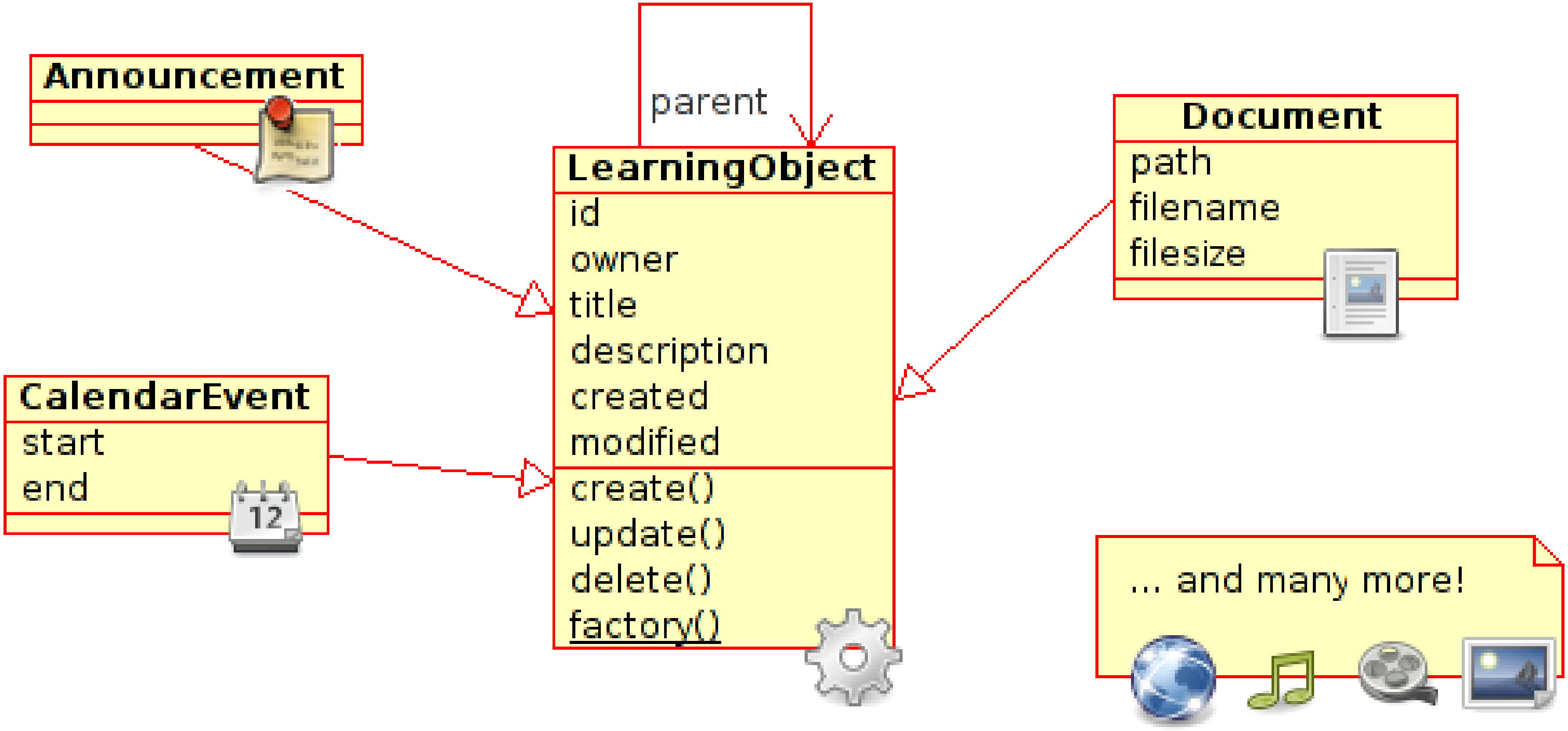
parent

LearningObject

id
owner
title
description
created
modified

create()
update()
delete()

Actual Learning Objects

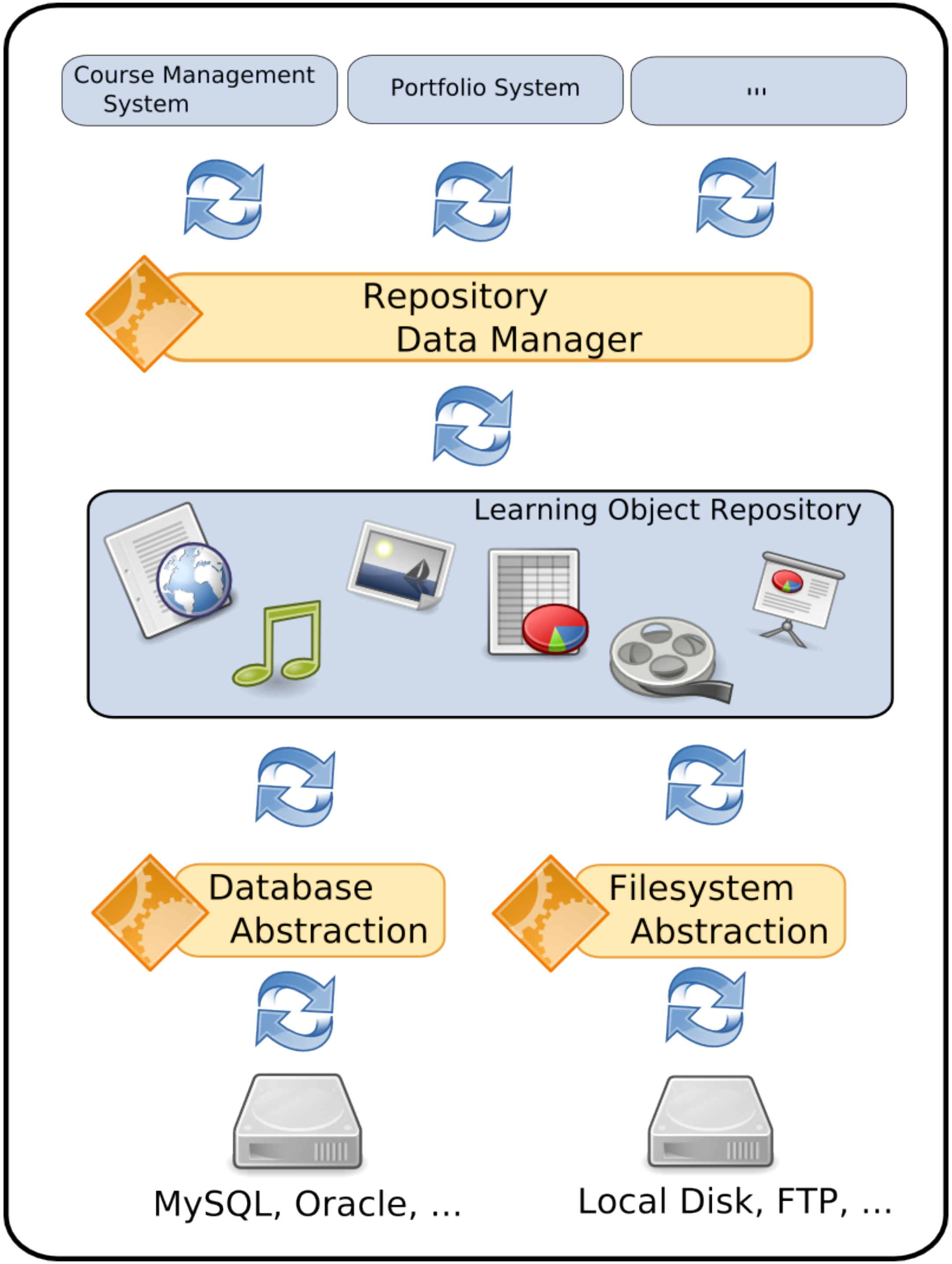


Objectives

- Object-oriented design
  - PHP 5
  - Design patterns
  - Refactoring
- No feature loss

- Code aesthetics
  - Readability
  - Consistency
- Modularity

Layered Design



Demonstration

Recap

- LCMS means
  - All old features + many new ones
  - Modularity ⇒ extensibility
  - A semi-fresh start
- New architecture involves
  - Persistent learning object repository
  - Framework for repository-based applications
- Long road ahead

End of Presentation

Thanks to Tom Brutin & Tim De Pauw