Kuali OLE – A Next Generation Library Management System
DEFF Master Class, Technical University of Denmark, Copenhagen, August 28, 2015

Technical Overview of OLE
Julian Ladisch
Verbundzentrale des GBV (VZG)
Modular Design

Kuali + Rice + modules + Java

= open source system for academic libraries

Kuali
Malaysian for a kind of wok

+ 

Rice

+ 

modules

+ 

Java

Author of the equation of images: Ingolf Kuss
System Architecture

- Java: Rice (Middleware), Spring Framework, Java Server Pages
- Search server: Solr
- Data sources:
  - Identification: LDAP / Active Directory, Shibboleth
  - OLE database
  - GOKb
- GUI (roles & permissions, workflow, workbench search, …)
- External systems:
  - External discovery system
  - Interfaces for RFID and ILL
System Architecture

- OLE is written in Java 7
  - About 5,350 Java files
  - Consisting of around 1 million lines (code, comments, blank lines)
  - Configuration and setup files

- Java Middleware
  - Kuali OLE (application) makes use of Kuali Rice (middleware), Java Server Faces and the Java Spring Framework.
  - In addition several hundreds freely available Java libraries.
  - Rice Middleware
    - In use in other Kuali software at about 20 universities for years
    - Open Source
System requirements – Hardware

• Server as a sandbox installation
  • Desktop PC, or virtual server
  • Memory: OLE recommends 8 GB RAM
    • 4 GB is possible (runs in two of three VZG installations)
    • 32 GB recommended for a small library
  • Multi core processor (e.g. 4 CPUs with 3.3 GHz each)

• Kuali OLE early adopters
  • University of Chicago
    • 2 Dell PowerEdge M620 with 20 cores each, 128 and 64 GB RAM, 800 GB SSD
  • Lehigh University
    • IBM x3550, 8 core Xeon ES-2667, 32 GB RAM (as at September 22, 2014)
  • SOAS
    • VM with 8 cores, 64 GB RAM
System requirements – Software

• Server
  • Operating system: Windows 7, Linux, e.g. Ubuntu (hbz, VZG, Chicago), Debian (Lehigh), RedHat (Chicago), CentOS (SOAS), SLES (hbz), OpenSuSE
  • Required software packages:
    • Java 7 RE
    • Tomcat 6/7
    • Database: MySQL 5.5/5.6, MariaDB or Oracle

• Client
  • Any popular web browser (e.g. IE, Firefox, Google Chrome)
  • JavaScript enabled
  • PDF viewer for printing (e.g. invoice, hold slip, routing slip)
Data Management

• Relational Database Management System (RDBMS)
• MySQL/MariaDB uses
  • InnoDB/XtraDB, supporting ACID-compliant transaction features and foreign keys
  • MyISAM for sequence tables and unique identifier tables
• About 1500 database tables

<table>
<thead>
<tr>
<th>Table usage</th>
<th>OLE tables</th>
<th>Framework tables</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular data</td>
<td>About 300</td>
<td>About 600</td>
<td>About 900</td>
</tr>
<tr>
<td>Sequence/unique identifier</td>
<td>About 300</td>
<td>About 300</td>
<td>About 600</td>
</tr>
<tr>
<td>Sum</td>
<td>About 600</td>
<td>About 900</td>
<td></td>
</tr>
</tbody>
</table>

• REST interfaces, e.g. for title and item queries
Search Engine

(Public) Discovery
- Interface for any external discovery system
- DigiBib (hbz)
- VuFind (Chicago, Lehigh, SOAS, VZG)
- EBSCO Discovery Service

Internal Search for staff
- Based on Apache Solr, an open source search platform
- Provides
  - Full text search
  - Highlighting
  - Facet search
  - Dynamic clustering
  - Proprietary formats (e.g. Word, PDF)
- Written in Java
Scalability

Setup of the University of Chicago Library

Write operations at replication servers.
Read operations at the replication slaves.

Source: Dale Arntson (University of Chicago Library) “OLE Deployment Architectures” at Kuali Days November 12, 2014
Hosting, Cloud Computing and Multi-Tenancy

• Hosting possible
• Client Server System in daily use
• What is Cloud Computing?
  • “Ubiquitous network access to a shared pool of configurable computing resources”
    (en.wikipedia.org)
  • Cost reductions and efficiency benefits compared to traditional hosting solutions
• Prerequisites
  • Hosting possible ✓
  • Scalability ✓
  • Management tools
Multi-Tenancy

• Multi-Tenancy
  • Several libraries use the same software and hardware installation
  • Dynamic resource allocation
  • Access restricted to own data

• Agenda for OLE 4.0
  • Full cloud capability and multi-tenancy after completing
    • Tenant administration software and tenant separation
    • Cloud management software
Operating Schemes

Private Cloud
• Administration by a library service centre (e.g. hbz/VZG)
• Administration by a commercial affiliate

Local instance
• Administration by the library or the university’s IT department
• Administration by a commercial affiliate
  • Hardware on campus or remote
Public development

• Sonar – software metrics
  http://sonar.kuali.org/dashboard/index/136042

• Latest code changes
  https://fisheye.kuali.org/changelog/ole

• Ticket system for enhancements requests and bugs
  https://jira.kuali.org/browse/OLE
Public development – OLE’s software metrics by Sonar

<table>
<thead>
<tr>
<th>Lines of code</th>
<th>Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>569,068</td>
<td>6,755</td>
</tr>
<tr>
<td>999,375 lines</td>
<td>521</td>
</tr>
<tr>
<td>257,078 statements</td>
<td>43,072</td>
</tr>
<tr>
<td>5,366 files</td>
<td>23,193 accessors</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Issues</th>
<th>Rules compliance</th>
</tr>
</thead>
<tbody>
<tr>
<td>42,773</td>
<td>83.7%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Package tangle index</th>
<th>Dependencies to cut</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.6/method</td>
<td>9.3% (1.457 cycles)</td>
<td>211 between packages, 675 between files</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unit Tests Coverage</th>
<th>Unit test success</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.9%</td>
<td>95.2%</td>
</tr>
<tr>
<td>3.4% line coverage</td>
<td>0 failures</td>
</tr>
<tr>
<td>1.5% branch coverage</td>
<td>11 errors</td>
</tr>
</tbody>
</table>

Technical Debt

- 18.7%
- $3,276,770
- 6,554 man days
Code Quality

• Agenda for improvements
  • Hire software architect
  • Code reviews
  • Use GitHub (currently Subversion/Fisheye/Crucible)

• Code has room for improvements
  • Source code documentation
  • Error handling, debugging
  • Unit test coverage
  • Some code is unnecessary complicated
Public Installations

- Demo installations, try it out!
- kuali.org
  - OLE 1.6: http://tst.ole.kuali.org/ http://dev.ole.kuali.org/
  - OLE 2.0: http://qa.ole.kuali.org/ http://staging.ole.kuali.org/
- hbz and VZG
  - http://ole.gbv.de/ (Ubuntu 14.04 LTS, 4 GB RAM)
    - Configured for University Library of Hildesheim
    - With title, instance and item data
  - http://kuali.hbz-nrw.de/ (Ubuntu 12.04 LTS, 48 GB RAM, 8 vCPUs, 1 TB HD)
    - Configured for University Library of Wuppertal
    - With title, instance and item data
Licence

Allowed and free of charge:
Run, copy, modify and distribute the software

Patent clause

Viral: All other modules of the software must also be under the same licence

The changed source code must be provided to

• the recipient of the binary program
• the user of the binary program (SaaS user)
Licence

**Educational Community Licence ECL 2.0** (for OLE prior to OLE 3.0)

- Based on Apache license 2.0
- Only difference: Patent clause, suitable for academic institutions
- Open Source: Run, copy, modify and distribute the software without charge
- Source code must show who changed the code and must be accompanied by the licence text
- No restrictions for software binaries – ECL code may be incorporated into proprietary code where only binaries are published.
Licence

AGPL (Affero General Public Licence) – from OLE 3.0 onwards
• Based on GNU General Public License (GPL) 3.0
• “Copyleft”, “Viral licensing”: Any program that contains some AGPL code must be under AGPL.
• Any user of a software has the right to get the source code even if the software runs on some remote host and is used only over computer networks.
  • This is different from GPL where network users doesn’t have this right (application service provider “loophole”).
• Making the source code available to all users
  • Prevents software add-ons that only a single software hoster offers (no monopoly, no vendor lock-in)
  • Enables further community driven development.
Internationalization

Internationalization is designing a software in a way that make adoptions of the software to different countries easy.

Localization is adapting an internationalized software to a specific country.

Internationalization

• Translation of texts in software and documentation (planned)
• Validation of phone numbers and postal codes – done ✓
• Date format – partly done (✓)
• Currency format and symbol – done ✓
• Number format (decimal mark, digit grouping separator) (planned)

Localization

• United Kingdom – done ✓
Thank you.

Julian Ladisch (VZG)
julian.ladisch@gbv.de
Sources:

Slide 2
Overall Image „Equation of Images“: Ingolf Kuss
Image „Wok“: http://upload.wikimedia.org/wikipedia/commons/a/a8/Wok-our-hand-hammered.jpg, Attribution/License: „By Tane Chan (http://www.wokshop.com/) [CC BY-SA 3.0 nl (http://creativecommons.org/licenses/by-sa/3.0/nl/deed.en)], via Wikimedia Commons“; we resized the image
Image „Rice“: http://pixabay.com/de/reis-lebensmittel-essen-wei%C3%9F-316532/, License: CC0 Public Domain
Image „Vegetables“: http://pixabay.com/de/karotten-zwiebel-gurke-gem%C3%BCse-155715/, License: CC0 Public Domain
Image „Caffe cup“: http://pixabay.com/de/kaffee-tasse-kaffeetasse-getr%C3%A4nk-1106/, License: CC0 Public Domain

Slide 4

Slide 10/20
Image „green tick“: Quelle: http://pixabay.com/de/zecken-marke-gr%C3%BCn-recht-39830/, License: CC0 Public Domain