



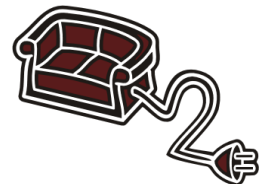
# SOFA 2 COMPONENT SYSTEM

## CURRENT STATUS

ONDŘEJ ČERNÝ PETR HOŠEK MICHAL PAPEŽ VÁCLAV REMEŠ

# Outline

- SOFA 2
- Previous work
- Goals
- Current Status
  - ▣ SOFA 2 Runtime
  - ▣ SOFA 2 Tools API
  - ▣ Cushion
  - ▣ SOFA 2 IDE
  - ▣ MConsole
  - ▣ SOFA Shop
- Future work



# SOFA 2

- Distributed component system employing hierarchically composed components
- Successor of the SOFA component model
- Complete framework supporting all the stages of an application lifecycle
- The component model defined by means of its meta-model
- <http://sofa.ow2.org>



# Previous Work

- Runtime development basically done
  - ▣ Some of the features not implemented
- Cushion as the only existing tool
- Prototype of SOFA IDE
  - ▣ Mostly unusable due to fragile implementation
  - ▣ Eclipse standards not followed
- Partial implementation of MConsole
  - ▣ Missing lots of needed logic

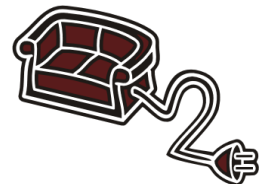
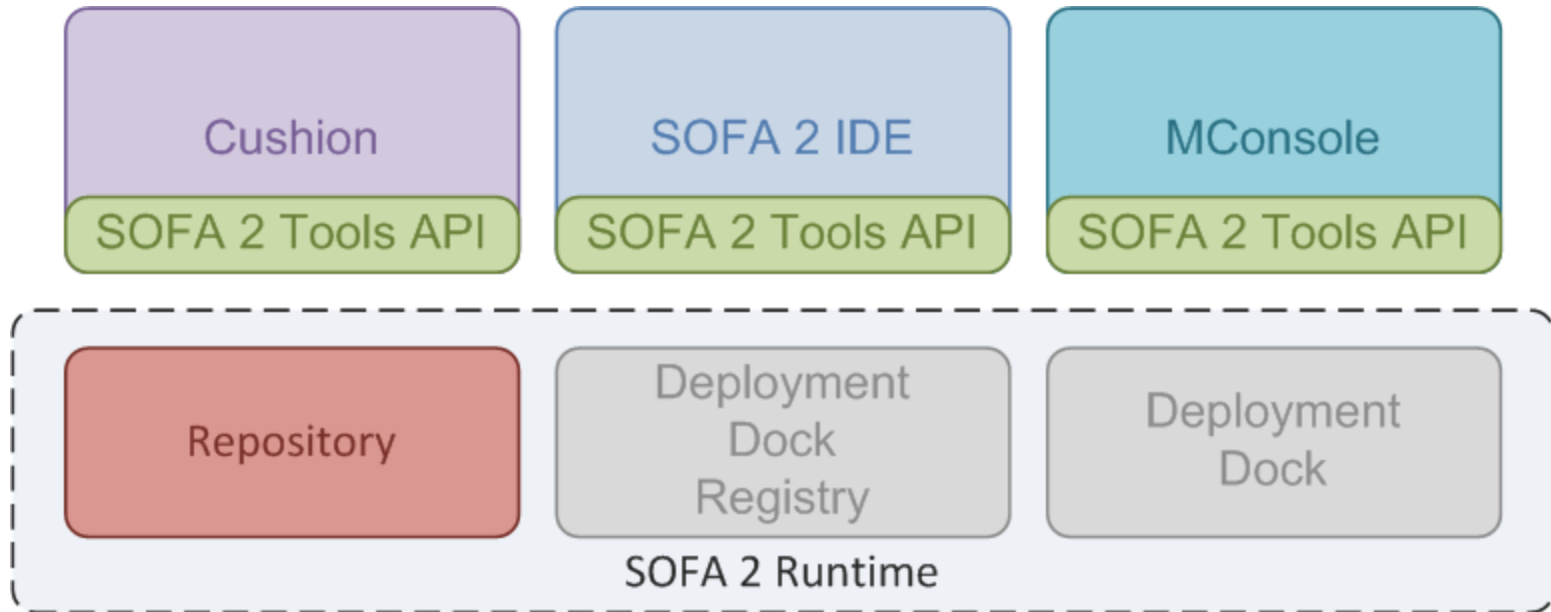


# Goals

- Complete development environment
  - ▣ Command line and visual interface
  - ▣ Support of all stages of development
- Tools for managing evolution of components
  - ▣ Manage multiple versions of single component
- Distribution packages of SOFA 2 components
- Tools for managing and monitoring component runtime
- "Autoconfiguration" execution mode

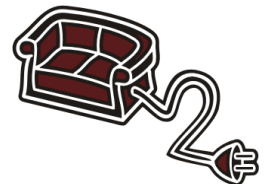


# Overall View



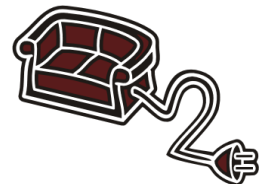
# SOFA 2 Runtime

- Entity *deletion* and *cloning/merging* support
  - ▣ Many repository server changes made
- Repository *locking* support
- *Microarchitecture* and *aspect* support
- *Autoconfiguration* support
  - ▣ Configuration server introduced
  - ▣ Uses existing scripts and code
- Improved runtime component support (properties, self-shutting, etc.)
- General refactoring and bug fixing



# SOFA 2 Tools API

- ❑ Originally part of the Cushion
  - ▣ Separated as a general API for SOFA 2 tool support
- ❑ Based upon SOFA 2 ADL which is a simplification of SOFA 2 model aimed on user friendliness
- ❑ Complete set of actions for all development and management tasks
- ❑ Introduces workspace abstraction
  - ▣ Maps SOFA entities to files and directories
- ❑ Programming language flexibility
  - ▣ Language for implementing components





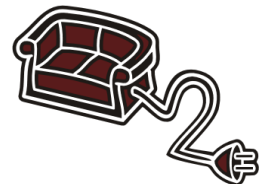
# SOFA 2 Tools API

- *Package distribution* supported by export and import actions
  - ▣ Implemented as a part SOFA 2 Tools API
  - ▣ Can be easily integrated in development tools
- Package is an archive containing entity's ADL description, code and dependency list
- Export and import with dependencies (recursive) supported



# Cushion

- *Development and management* tool
  - ▣ Command line oriented
- Completely refactored and made more user-friendly
- Based entirely on top of SOFA 2 Tools API
- Full support of versions' *tags*
- Support for *microarchitecture* and *aspects* added



# SOFA 2 ADL

- Eclipse extension reflecting the need of ADL processing inside the SOFA 2 IDE and MConsole
  - ▣ Modeled using the *Eclipse Modeling Framework*
  - ▣ Model (de)serialized from/into the ADL XML file used by the SOFA 2 Tools API and other tools
  - ▣ Extensions to support logic needed by the tools
- *Eclipse QVTO* transformation for runtime conversion between models



# SOFA 2 ADL

- Form editors created using the *Eclipse JFace* UI toolkit
  - ▣ Replacement for the simple generated editors
  - ▣ *Eclipse JFace Databinding* used to speed up development
- Graphical architecture editor created using *Eclipse Graphical Modeling*



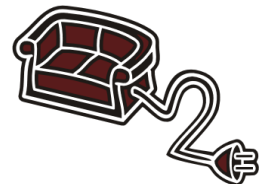
# SOFA 2 IDE

- ❑ Complete SOFA 2 *development* process support inside *Eclipse IDE*
  - ❑ Based on top of SOFA 2 ADL
  - ❑ Seamless Eclipse integration following the standards
- ❑ Contributes project nature and perspective as well as number of views, editors and wizards
- ❑ Extension points and API for building new Eclipse plugins with access to SOFA 2 infrastructure



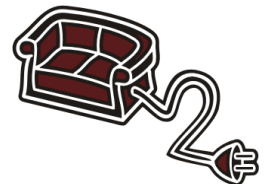
# SOFA 2 IDE

- Entity versioning and repository access through SOFA 2 Team Provider implementation
  - ▣ Using *Eclipse Team Support* as a standard way for versioning and repository management in Eclipse
- Built on top of the SOFA 2 Tools API
  - ▣ Encapsulating its logic
  - ▣ Providing all the actions directly from *Eclipse IDE*



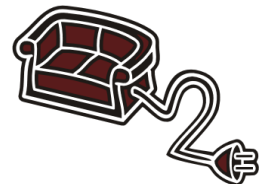
# MConsole

- Runtime *management* tool
  - ▣ Both Eclipse plugin and standalone application
  - ▣ Sharing common libraries and SOFA 2 ADL model with SOFA 2 IDE
- Support for runtime environment management and application launching
  - ▣ Support for the autoconfigured environment
  - ▣ SOFA 2 launch configuration implementation
  - ▣ Runtime environment visualisation using *Eclipse Zest* framework



# SOFA Shop

- Based on CoCoME
  - ▣ Just use cases, not the architecture
- Modeled by using means of SOFA 2
- "Real-life" project implemented using SOFA 2 component system
- Developed using our development tools
  - ▣ Cushion used during the initial development phase
  - ▣ Switched to SOFA 2 IDE during work
  - ▣ Gained feedback used to improve the tools





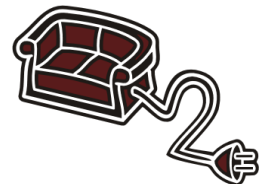
# SOFA Shop

- Pure object database used for data storage
  - ▣ Smooth integration and easy first-time configuration, but poor stability and tool support
- *Eclipse Modeling Framework* based data model
  - ▣ *Model To Text* transformation utilized for generating code
- User interface using *Swing* toolkit



# Development and Build Process

- Many subprojects separated during the project
  - ▣ Dependencies explicitly specified using the *Apache Ivy* dependency manager
  - ▣ External dependencies resolution
- Build process simplified and made fully automatic
- india.ms.mff.cuni.cz
  - ▣ SOFA 2 build server running the *Hudson* continuous integration server and *Apache Ivy* repository



# Conclusion

- General usability of SOFA 2 component system overall improved
  - ▣ Many missing features implemented
  - ▣ Quality usable development tools
  - ▣ Simplified usage and build process
- Working real-life application implementation
- Projects already building on top of our work

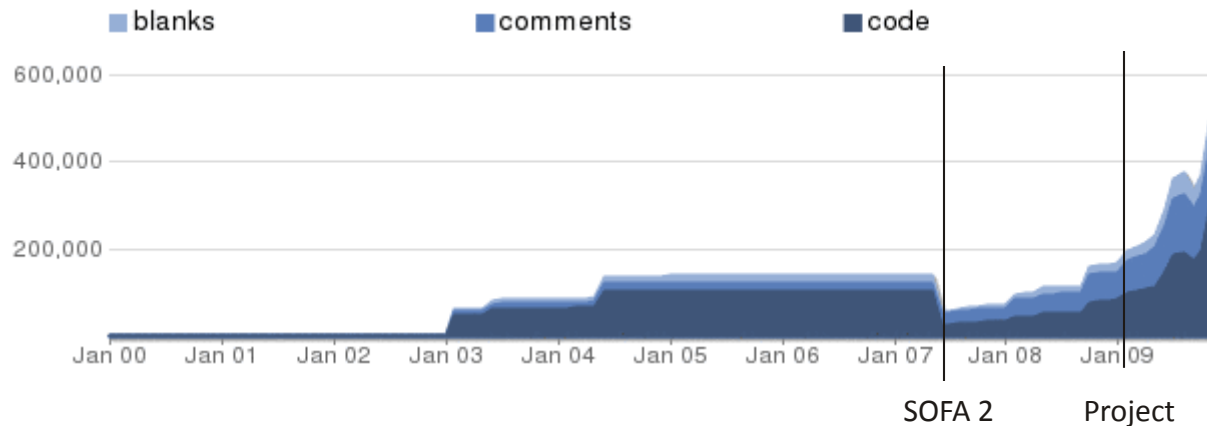


# Statistics

## □ Project cost (estimated by Ohloh)

<b>Codebase</b>	234,689 lines of code
<b>Effort (est.)</b>	61 Person Years
<b>Cost</b>	\$3,328,797 (avg. salary \$55,000 per year)

## □ Lines of code



# Future Work

- Support for development of enterprise applications
- Dynamic languages runtime support
- Support for cloud computing and SOFAnet 2
- Migration and load balancing support
- Support for deployment into embedded environment
- SOFA HI profile of SOFA 2 intended for real-time embedded applications



# Questions and Discussion

---

