



Calhoun: The NPS Institutional Archive
DSpace Repository

Reports and Technical Reports

Faculty and Researchers' Publications

2010-05-12

The Challenge of Heterogeneously Licensed Systems in Open Architecture Software Ecosystems

Alspaugh, Thomas A.; Scacchi, Walt

Monterey, California. Naval Postgraduate School

<https://hdl.handle.net/10945/33511>

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun



Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

<http://www.nps.edu/library>



Acquisition Research Program:
Creating Synergy for Informed Change

The Challenge of Heterogeneously Licensed Systems in Open Architecture Software Ecosystems

Thomas A. Alspaugh and Walt Scacchi
Institute for Software Research, University of California, Irvine

Overview

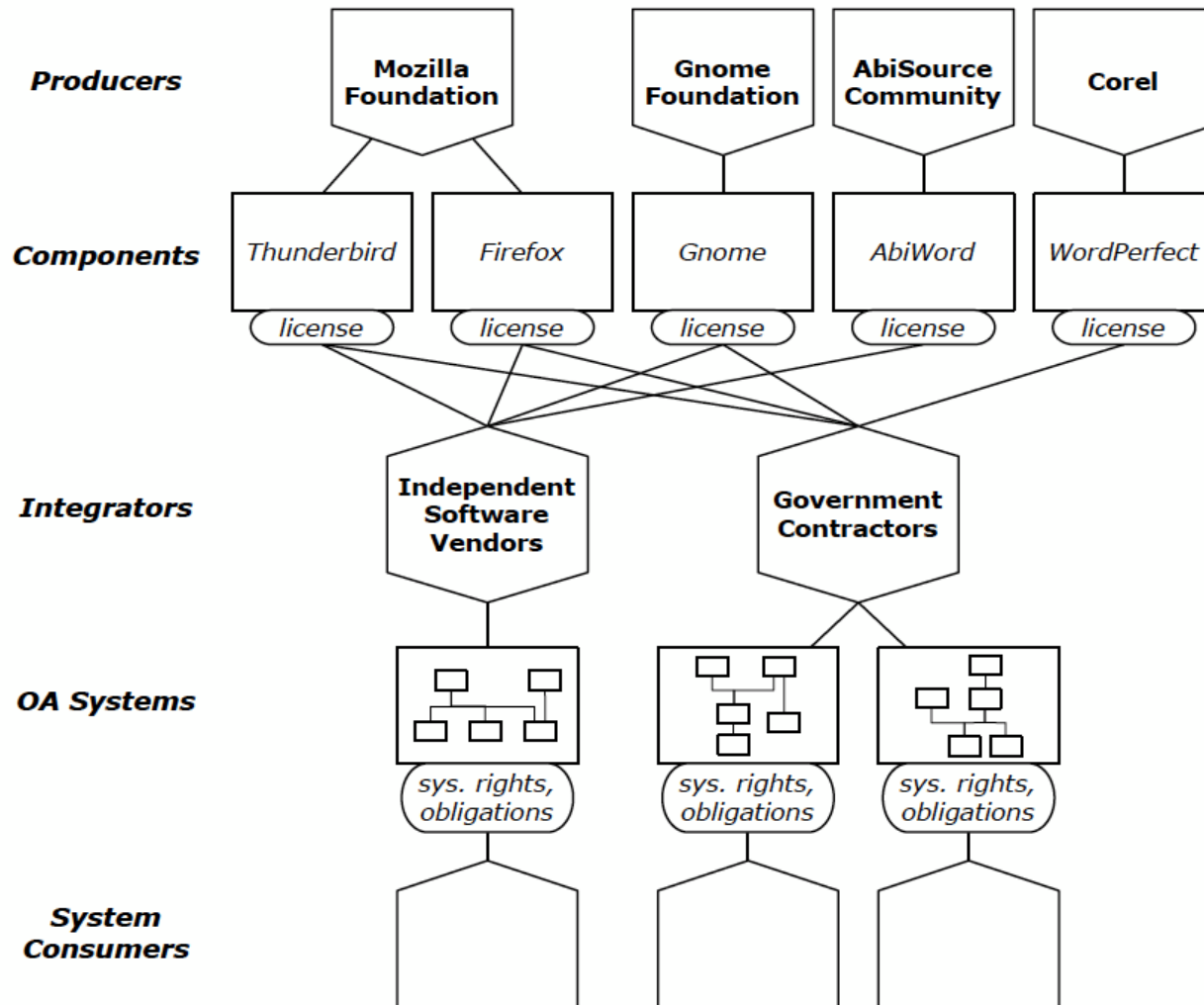
- Background
- Developing Open Architectures (OAs)
- Software licenses, architectures, and analysis
- Discussion
- Conclusions



Background



OA software ecosystem – mapping rights and obligations from producers to consumers



A heterogeneous software license for the *Unity 3D* software system

1. The Mono Class Library, Copyright 2005-2008 Novell, Inc.
2. The Mono Runtime Libraries, Copyright 2005-2008 Novell, Inc.
3. Boo, Copyright 2003-2008 Rodrigo B. Oliveira
4. UnityScript, Copyright 2005-2008 Rodrigo B. Oliveira
5. OpenAL cross platform audio library, Copyright 1999-2006 by authors.
6. PhysX physics library. Copyright 2003-2008 by Ageia Technologies, Inc.
7. libvorbis. Copyright (c) 2002-2007 Xiph.org Foundation
8. libtheora. Copyright (c) 2002-2007 Xiph.org Foundation
9. zlib general purpose compression library. Copyright (c) 1995-2005 Jean-loup Gailly and Mark Adler
10. libpng PNG reference library
11. jpeglib JPEG library. Copyright (C) 1991-1998, Thomas G. Lane.
12. Twilight Prophecy SDK, a multi-platform development system for virtual reality and multimedia. Copyright 1997-2003 Twilight 3D Finland Oy Ltd
13. dynamic bitset, Copyright Chuck Allison and Jeremy Siek 2001-2002.
14. The Mono C# Compiler and Tools, Copyright 2005-2008 Novell, Inc.
15. libcurl. Copyright (c) 1996-2008, Daniel Stenberg <daniel@haxx.se>.
16. PostgreSQL Database Management System
17. FreeType. Copyright (c) 2007 The FreeType Project (www.freetype.org).
18. NVIDIA Cg. Copyright (c) 2002-2008 NVIDIA Corp.



Supporting OA with heterogeneously licensed system components

- Must account for design-time, build-time, and run-time architectures
- Must distinguish architect constructs relevant to software licenses, and license effects
- Must define license architecture
- Must provide automated environment for managing system and license architectures
- Must automate calculations of system license rights, obligations, architectures as they evolve



Developing OA systems with heterogeneously licensed software components



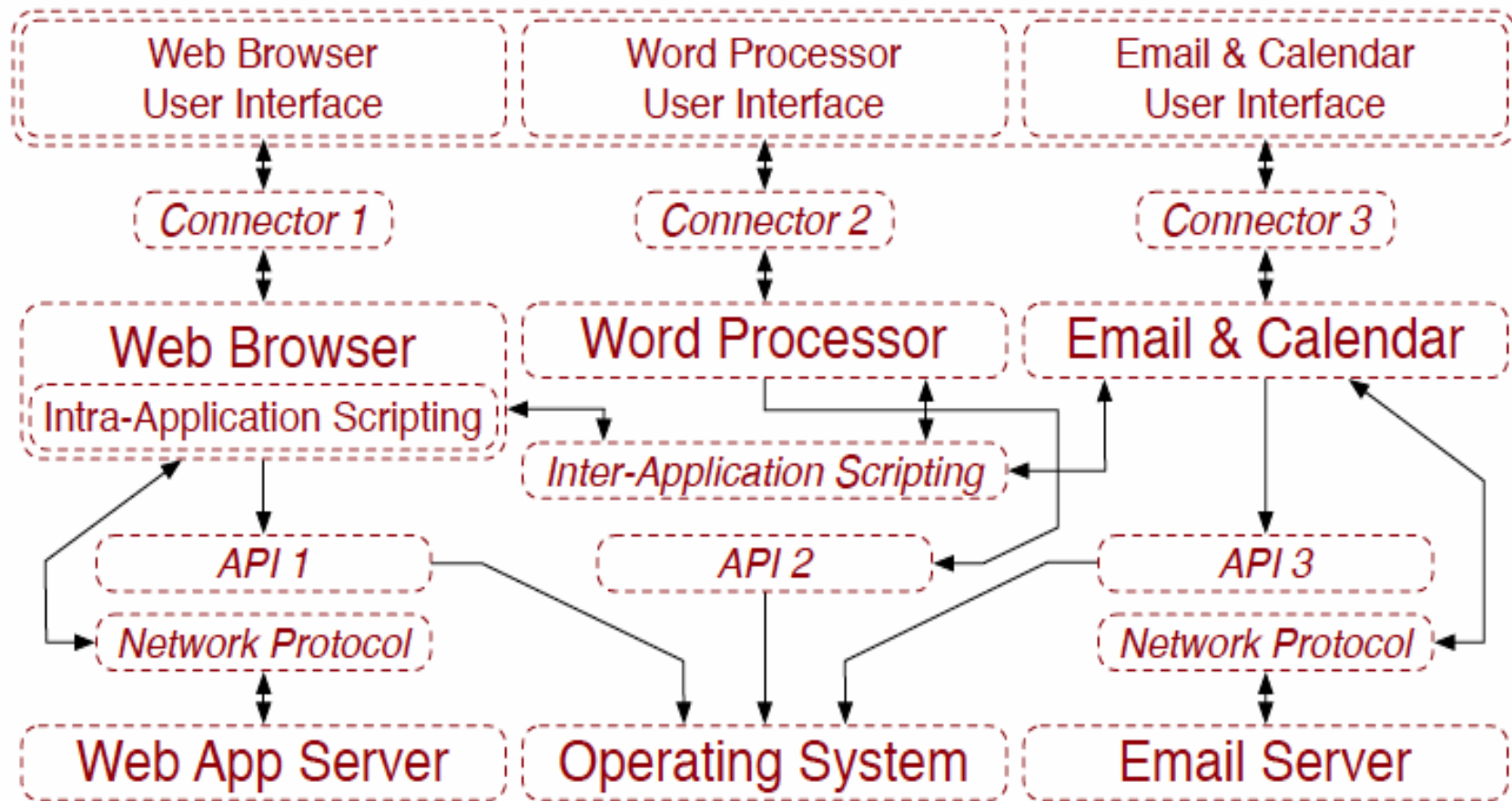
A composed multi-component OA system

The screenshot displays a multi-component Open Architecture (OA) system running on a desktop environment. The system is composed of several interconnected components:

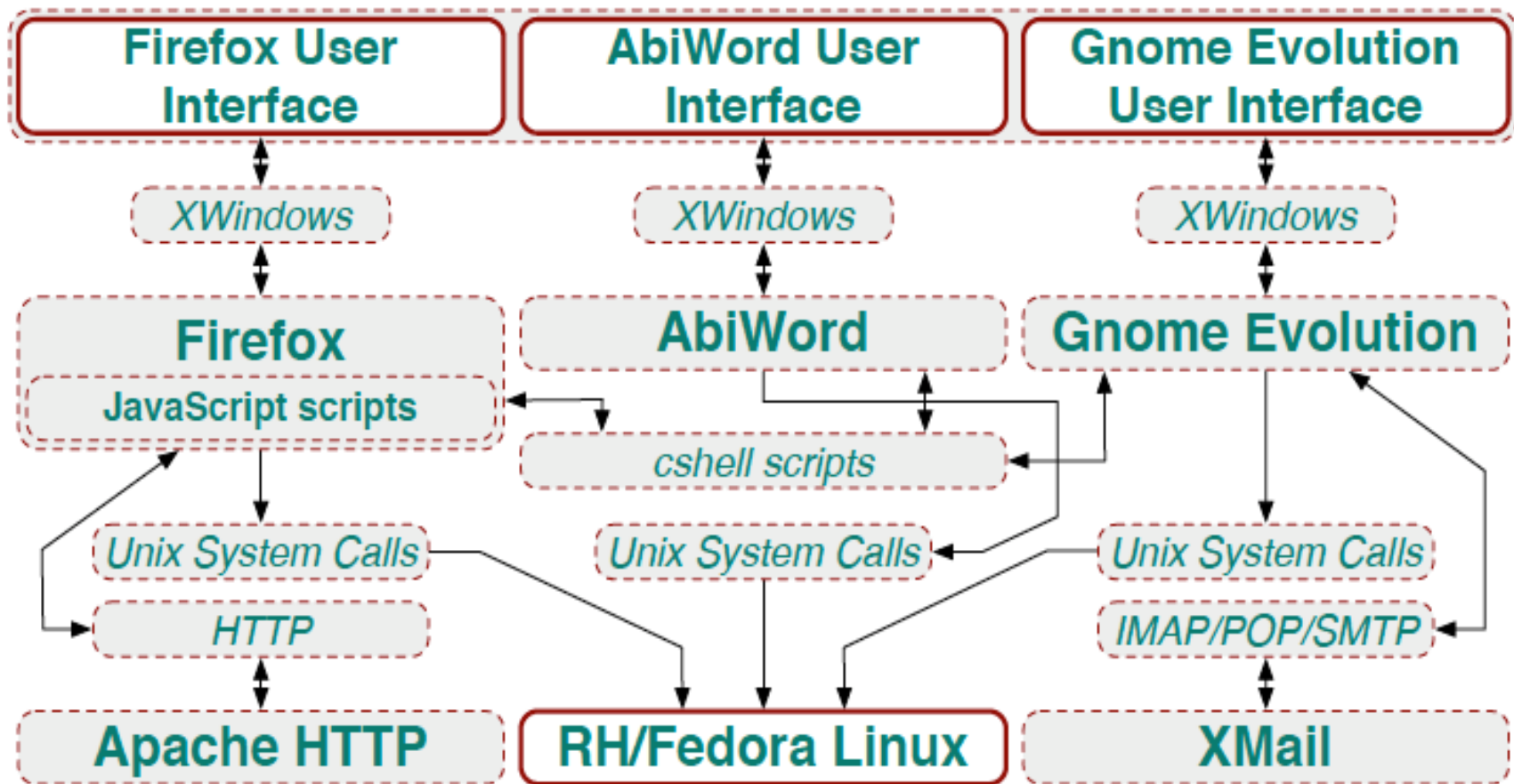
- Website Component:** A Mozilla Firefox browser window showing the 'GAME CULTURE & TECHNOLOGY LAB' website. The page includes a navigation menu, a search bar, and a 'Mission' section.
- Document Component:** A Microsoft Word document titled 'JSS-Figure4-draft.abw' is open, displaying a slide titled 'A Composed Open Architecture Software System at Run-Time'. The slide content is a screenshot of the same multi-component OA system.
- Calendar Component:** An Evolution calendar window is open, showing a daily view for Monday, April 26, 2010. It lists events such as '1:00pm Proposal review meeting' and '3:00pm Work on JSS paper draft'.
- Terminal Component:** A terminal window titled 'liveuser@localhost:~\$' shows the output of the 'ls' command, listing various system files and directories in the /sbin directory.



Design-time architecture



Build-time architecture



Run-time architecture (user view)

The screenshot displays a desktop environment with several applications open:

- Mozilla Firefox:** Displays the Game Culture & Technology Lab website. The page title is "GAME CULTURE & TECHNOLOGY LAB". The content includes a mission statement: "The mission of the Game Culture & Technology Lab is to play with how game metaphors, design principles, and technologies can be utilized for alternative content and context delivery. The focus is on the next generation internet and beyond." It also lists methods like sampling, misuse, hacking, appropriation, reverse engineering, and custom creation.
- Microsoft Word:** Opened document titled "A Composed Open Architecture Software System at Run-Time". The document content is a screenshot of the same website shown in the browser.
- Calendars - Evolution:** Shows a calendar for Monday, April 26, 2010. Key events include:
 - 1:00pm: Proposal review meeting
 - 3:00pm: Work on JSS paper draft
- Terminal:** Shows a shell prompt at `liveuser@localhost:~$`. The user has run `pwd` (output: `/sbin`) and `ls`. The `ls` output lists various system files and directories, including `dmraid.static`, `ifconfig`, `lvchange`, `modinfo`, `pppoe-sniff`, `sfdisk`, and `vgremove`.



Evolutionary changes in OA Systems

- Component evolution
- Component replacement
- Architecture evolution
- Component license evolution
- Change in desired rights or acceptable obligations

- Evolutionary changes reconfigure a system's software ecosystem!



Component replacement and architecture evolution

The screenshot displays a desktop environment with three main windows:

- GCTL - Mission - Mozilla Firefox:** A website titled "GAME CULTURE & TECHNOLOGY LAB". It features a navigation menu with "Mission", "Events", "Projects", "Tasks", "Projects", and "Package". Below the menu is a search bar with "English" selected and a "Search" button. The main content area includes a "Mission" section with text about the lab's goals and a "Methods" section listing techniques like sampling, misuse, and hacking.
- A Composed Open Architecture Software... - Google Docs - Mozilla Firefox:** A Google Docs document titled "A Composed Open Architecture Software System at Run-Time". The document content is partially visible, showing a screenshot of the GCTL website and a terminal window. The terminal window shows a list of SELinux policies and their associated modules.
- Google Calendar - Mozilla Firefox:** A Google Calendar window showing the date "Monday, Apr 26, 2010". The calendar view shows a grid for the month of April 2010, with a task "Proposal review meeting" scheduled for 2:30 PM on Monday, April 26th.

The terminal window in the Google Docs document shows the following SELinux policies and modules:

```
dmraid_static ifconfig lvchange modinfo pppoe-sniff sfdisk vgrremove
dmssetup ifdown lvconvert modprobe pppoe-start shutdown vgrrename
dosfsck ifenlave lvcreate mount.fuse pppoe-status slattach vgs
dosfslabel ifrename lvdisplay mount.nfs pppoe-stop sln vgschan
dump ifup lvxentend start nfs4 ppp-watch start vgsplit
dumpe2fs init lvm mount.ntfs pvchange start uddev weak-modules
e2fsck initctl lvchange mount.ntfs-3g pvck status vgsbind
e2image initlog lvm2diskscan mount.ntfs-fuse pvcreate stop ypbind
```

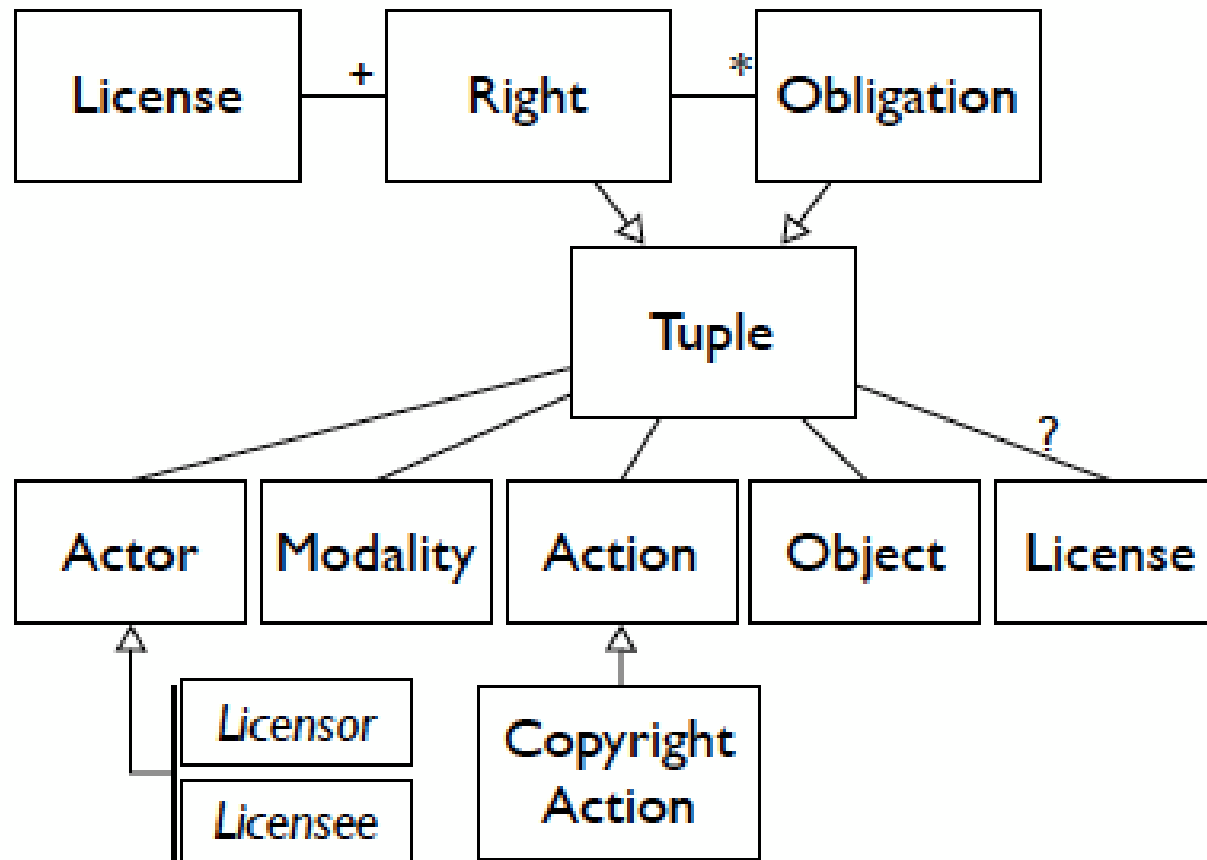
```
[liveuser@localhost sbin]$ pwd
/sbin
[liveuser@localhost sbin]$ cd ../selinux
[liveuser@localhost selinux]$ ls
access checkreqprot compat_net deny_unknown initial_contexts mls policyvers user
avc class disable enforce load member null reject_unknown
booleans commit_pending_boots create enforce member null policy_capabilities relabel
```



Software licenses, architectures, and analysis



Software license meta-model

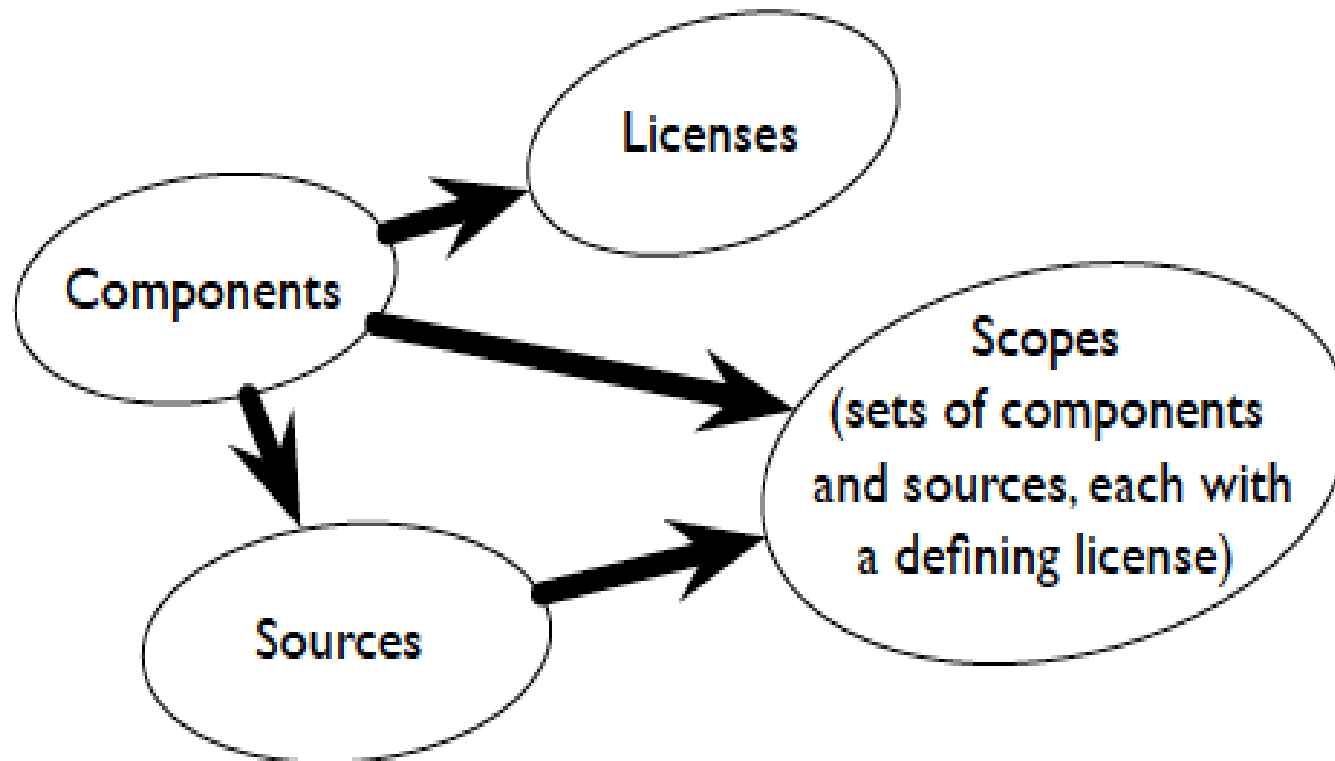


Logical modality and objects of software license rights and obligations

	Modality	Object	License (optional)
Abstract Right	<i>May or Need Not</i>	<i>Any Under This License</i>	<i>This License or Object's License</i>
		<i>Any Source Under This License</i>	
		<i>Any Component Under This License</i>	
Concrete Right		Concrete Object	Concrete License
Concrete Obligation			
Abstract Obligation	<i>Must or Must Not</i>	<i>Right's Object</i>	<i>Concrete License or Right's License</i>
		<i>All Sources Of Right's Object</i>	
		<i>X Scope Sources</i>	
		<i>X Scope Components</i>	



The software license architecture meta-model



Software license analysis

- License types:
 - Strongly reciprocal (GPL), weakly reciprocal (LGPL), academic (BSD), Terms of Service, Proprietary
- Propagation of reciprocal obligations
- Conflicting obligations
- Calculating obligations and rights



Component license annotation prior to analysis

ArchEdit

Licname: Description

Name	Value
value	GPL 2

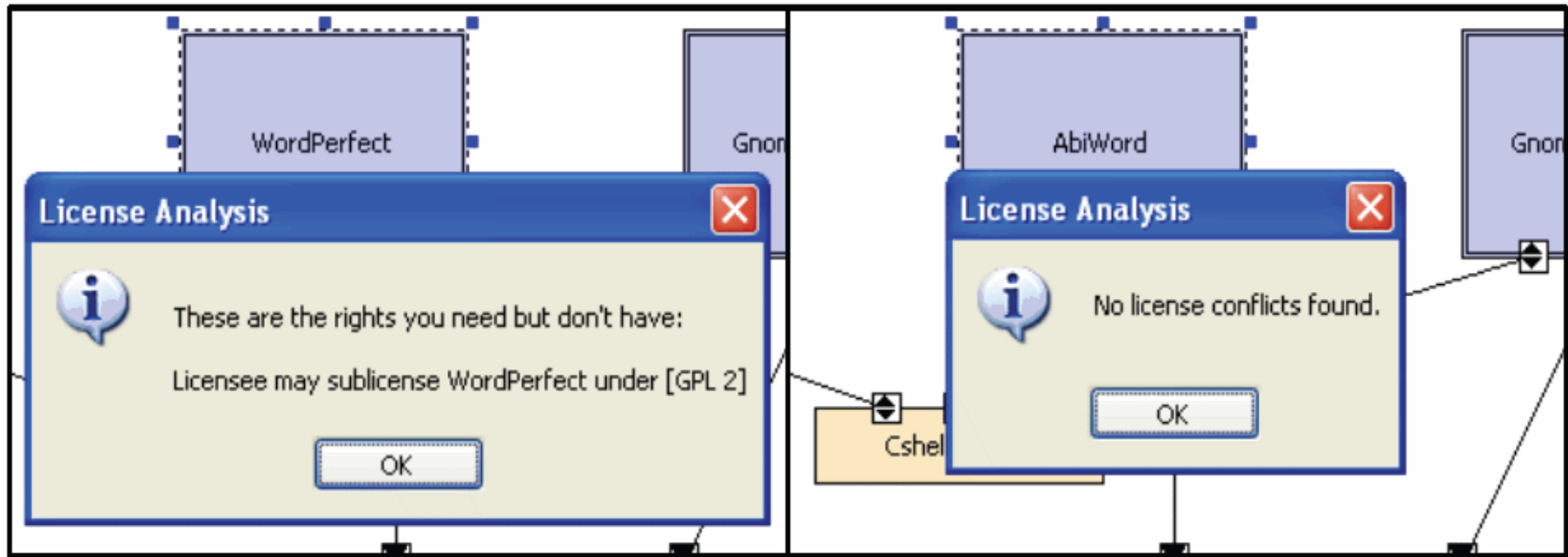


License review during license analysis

The screenshot displays the ArchStudio 4 interface. The main window shows a dependency graph with nodes for Firefox, WordProcessor, and GnomeEvolution, all depending on Unix System. Other nodes include HTTP, Cshell scripts, and IMAP/POP/SMTP. A 'View License Information' window is open, showing details for the GPL 2 license, including the code 'WordProcessor', first revision '9500', and latest revision '9504'. Below the graph, a Mozilla Firefox browser window displays the GNU General Public License Version 2, June 1991. The browser window shows the license text, including the preamble and the first section, which states that the license is intended to guarantee freedom to share and change software.



Results from license analyses with system component replacement



Discussion



Software product lines (SPLs) and OA systems

- An SPL may or may not be an OA system
- If SPL subject to single vendor/proprietary license, then lock-in is possible
- If OA system has design-time reference architecture and instantiated build-time architecture, then OA conforms to an SPL
- If SPL is based on OA with heterogeneously licensed components, then OA conforms to a *virtual SPL*, and works with our approach.



Specifying and analyzing system security requirements as “licenses”

- Security capabilities can correspond to “rights and obligations” in licenses
- Should be possible to specify and analyze system *security architecture* that conform to a *security meta-model*, much like we do for software licenses
- Should be possible to develop computational tools and development environments that can analyze security at design-time, build-time, and run-time, as well as when the system evolves



Conclusions

- Software component licenses and heterogeneously licensed systems becoming more widespread as we move to OA software ecosystems
- Our approach and tools demonstrate the ability to specify, model, and analyze such systems as they evolve, and are subject to diverse licenses
- Our approach is compatible with SPLs and can be extended to support system security licenses



Acknowledgements

This presentation was developed under work supported by the Naval Postgraduate School Acquisition Research Program Assistance Agreement No. #N00244-10-1-0038 awarded by the U.S. Fleet Industrial Supply Center, San Diego (FISC-SD). It has not been formally reviewed by NPS. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. FISC-SD and NPS. The FISC-SD and NPS do not endorse any products or commercial services mentioned in this publication.

Funding also from grant #0808783 from the National Science Foundation. No review, approval, or endorsement implied.

