Understanding the Requirements for Open Source Software Development

Walt Scacchi
Institute for Software Research
University of California, Irvine
Irvine, CA 92697-3425
Wscacchi@ics.uci.edu

http://www.ics.uci.edu/~wscacchi/Presentations/OSS-Requirements

Overview

- Research methodology
- Community characteristics
- Software Requirements process
- Open source processes for Requirements
- Software Informalisms
- Implications
- Conclusions

Research methodology

- Prior empirical (case) studies of Open Source Software Development (OSSD) Projects
  - Mockus, Fielding, Herbsleb, 2000, 2002, Apache httpd server
  - Reis and Fortes, 2002, Mozilla Web browser
  - Schach *et al.*, 2002; Holt *et al.*, 2000, Linux Kernel
  - Koch and Schneider 2001; German 2002, GNOME User Interface
  - Jorgensen, 2001, FreeBSD operating system
  - Garg *et al.*, 2002, OSSD ("progressive open source") within HP

Research methodology

- Individual case studies: significant details, but limited (and *premature*) generalization, little/no comparative analysis
  - Halloran and Scherlis, 2002, comparative study of software tools and code volume in *eleven* OSSD projects, all in one domain (Internet infrastructure)
- No studies that examine *multiple OSSD projects* in *multiple domains*
  - Such studies would offer higher degree of comparative analyses and generalization of results

Research methodology

- Comparative case studies
  - Multiple open software development projects
    - Across four communities
      - Two *research* oriented
      - Two *development* oriented
- Qualitative ("grounded theory") techniques
- Analyzing and modeling
  - development processes
  - work practices
  - community structures

Community characteristics

- According to *Steve Ballmer* (CEO, Microsoft)
  - "We have to compete with free software, on value, but in a smart way. We cannot price at zero, so we need to justify our posture and pricing. Linux isn't going to go away--our job is to provide a better product in the marketplace."
  - "Linux is not about free software, it is about *community*"(emphasis added).
  - London, 24 September 2002, speaking on MS, its "Most Valued Professionals" (MVPs), and "shared source" vs. "open source"

Community characteristics

- *Development* oriented domains
  - Networked computer games
  - Internet infrastructure
- *Research* oriented domains
  - Astrophysics/deep space imaging
  - Academic software design

Software Requirements process

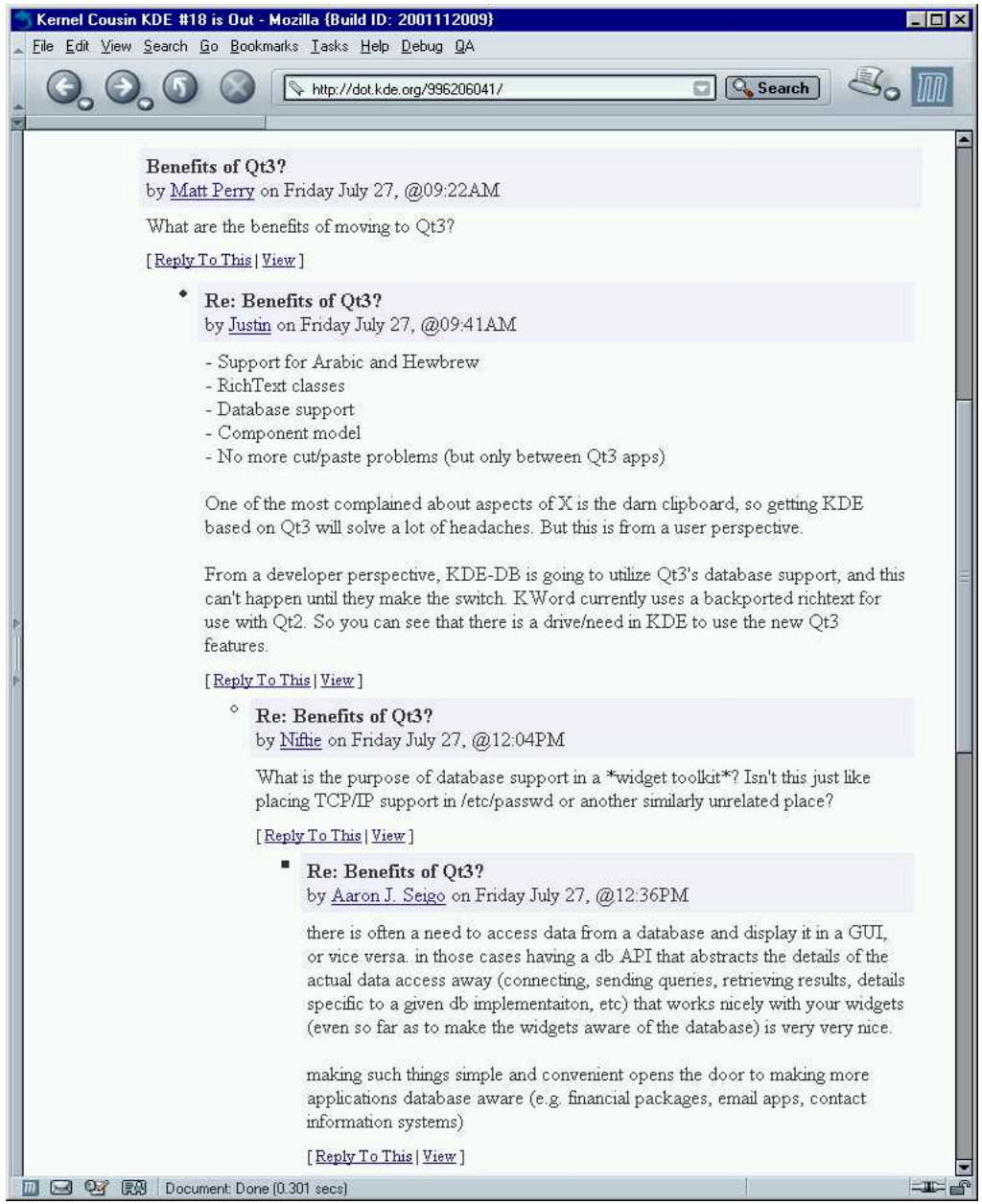
- Classic Requirements Engineering Process
  - Elicitation
  - Analysis
  - Specification and modeling
  - Validation
  - Communicating and managing

Open source processes for Requirements

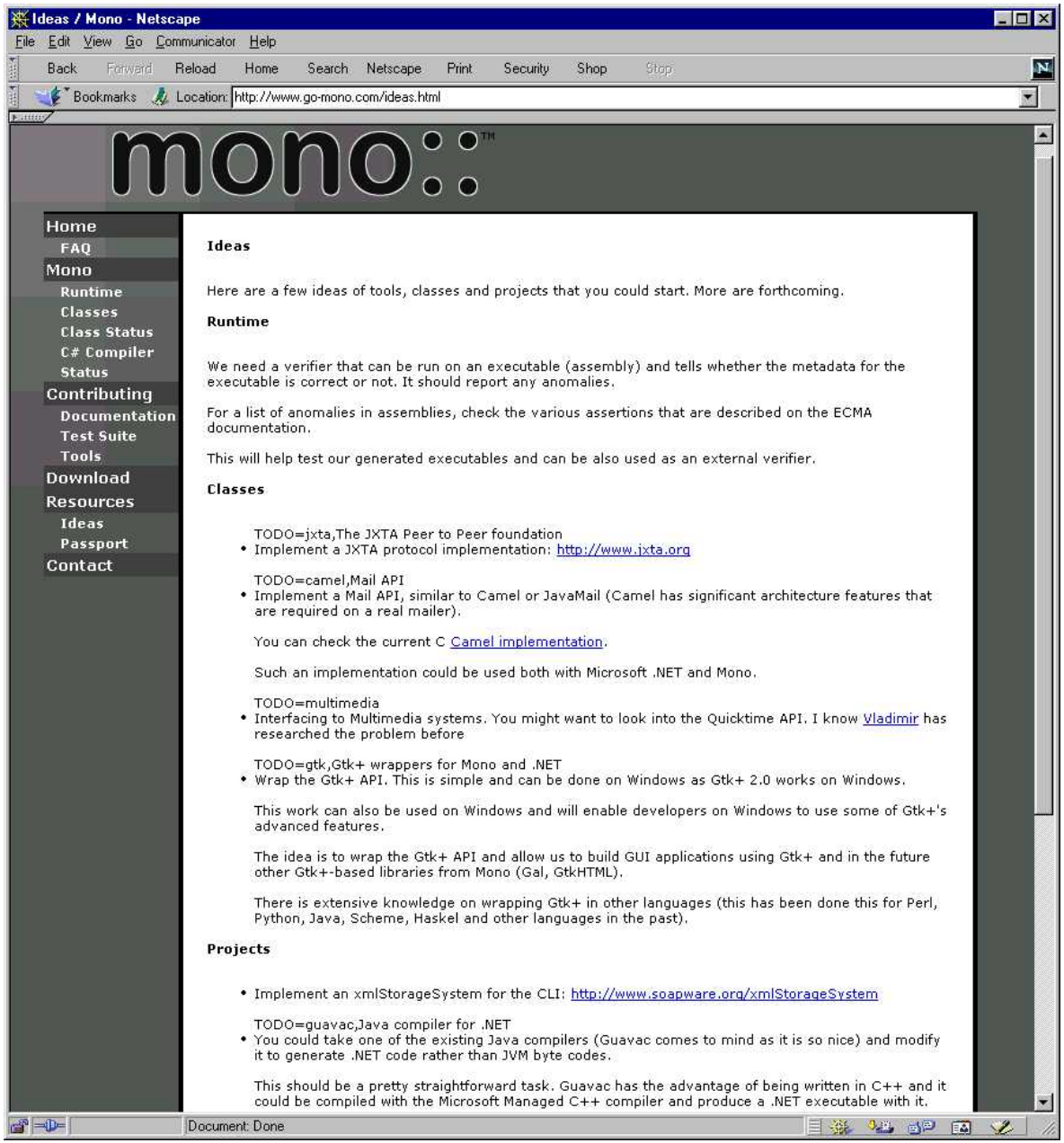
- *Post-hoc* assertion of requirements+design
- Reading, sense-making, accountability
- Continually emerging webs of discourse
- Condensing and hardening discourse
- Global access to discourse

Open source processes for Requirements

- OSS Requirements are
  - not explicit
  - not formal
- *QED*, OSS Requirements embedded within "informalisms"
- Example OSS informalisms follow


Kernel Cousin KDE #18 is Out - Mozilla {Build ID: 2001112009}
File Edit View Search Go Bookmarks Tasks Help Debug QA
http://dot.kde.org/996206041/
Search
Benefits of Qt3?
by Matt Perry on Friday July 27, @09:22AM
What are the benefits of moving to Qt3?
[Reply To This | View]
Re: Benefits of Qt3?
by Justin on Friday July 27, @09:41AM
- Support for Arabic and Hewbrew
- RichText classes
- Database support
- Component model
- No more cut/paste problems (but only between Qt3 apps)
One of the most complained about aspects of X is the darn clipboard, so getting KDE based on Qt3 will solve a lot of headaches. But this is from a user perspective.
From a developer perspective, KDE-DB is going to utilize Qt3's database support, and this can't happen until they make the switch. KWord currently uses a backported richtext for use with Qt2. So you can see that there is a drive/need in KDE to use the new Qt3 features.
[Reply To This | View]
Re: Benefits of Qt3?
by Niftie on Friday July 27, @12:04PM
What is the purpose of database support in a *widget toolkit*? Isn't this just like placing TCP/IP support in /etc/passwd or another similarly unrelated place?
[Reply To This | View]
Re: Benefits of Qt3?
by Aaron J. Seigo on Friday July 27, @12:36PM
there is often a need to access data from a database and display it in a GUI, or vice versa. in those cases having a db API that abstracts the details of the actual data access away (connecting, sending queries, retrieving results, details specific to a given db implementaiton, etc) that works nicely with your widgets (even so far as to make the widgets aware of the database) is very very nice.
making such things simple and convenient opens the door to making more applications database aware (e.g. financial packages, email apps, contact information systems)
[Reply To This | View]
Document: Done (0.301 secs)


Vision - Mozilla {Build ID: 2001112009}
File Edit View Search Go Bookmarks Tasks Help Debug QA
http://argouml.tigris.org/vision.htm
Search
Project Home News Downloads Mailing Lists Docs Source Issues Members
Highlights: Tours | Documentation | Developer Zone
Tigris
http://www.tigris.org
Login
Registered User
New User
Tigris.org
Home
Community
News
Help
Search
Featured Projects
EyeBrowse
Anzu
Scarab
Subversion
ArgoUML
GEF
new Jabber
new binarycloud
All Projects...
Categories
construction
design
issuetrack
libraries
profession
scm
students
techcomm
ArgoUML Vision
Cognitive Support for Design
Software design is a cognitively challenging task. Designers must manually enter designs, but the primary difficulty is decision-making rather than data-entry. If designers improved their decision-making capabilities, it would result in better designs.
Current CASE tools provide automation and graphical user interfaces that reduce the manual work of entering a design and transforming a design into code. They aid designers in decision-making mainly by providing visualization of design diagrams and simple syntactic checks. Also many CASE tools provide substantial benefits in the area of version control and concurrent design mechanisms. One area of design support that has been not been well supported is analysis of design decisions.
Current CASE tools are usable in that they provide a GUI that allows designers to access all the features provided by the tool. And they support the design process in that they allow the designer to enter diagrams in the style of popular design methodologies. But they typically do not provide process support to guide the designer through the design task. Instead, designers typically start with a blank page and must remember to cover every aspect of the design.
ArgoUML is a domain-oriented design environment that provides cognitive support of object-oriented design. ArgoUML provides some of the same automation features of a commercial CASE tool, but it focuses on features that support the cognitive needs of designers. These cognitive needs are described by three cognitive theories:
reflection-in-action,
opportunistic design, and
comprehension and problem solving.
Following the UML Specification
ArgoUML is based directly on the UML 1.3 specification. In fact, a large part of ArgoUML was generated automatically from the UML specification. ArgoUML is (to the best of our knowledge) the only tool that implements the UML meta-model exactly as specified. In contrast, current commercial tools use tools use basically the same internal representation of the design that they used in previous versions.
Furthermore, it is our goal to provide comprehensive support for OCL (the Object Constraint Language) and XMI (the XML Model Interchange format), which other tools currently do not support.
Community Software Development
As with all software produced by our research group, ArgoUML is available for free and can be used in commercial settings. For terms of use, see the license agreement presented when you download ArgoUML.
ArgoUML was originally developed by a small group of people as a research project. ArgoUML has many features that make it special, but it does not implement all the features that commercial CASE tools provide. We are providing the source code for ArgoUML for you to review, customize to your needs, and improve. Over time, we hope that ArgoUML will evolve into a powerful and useful tool for everyone to use.
Community software development has worked successfully to produce the FSF's GNU utilities and the Linux operating system. It has also worked in producing GEF, the Graph Editing Framework, which is a major component of ArgoUML.
For more information on contributing to the development of ArgoUML, see the Project page.
Document: Done (0.341 secs)


Ideas / Mono - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://www.go-mono.com/ideas.html
mono::™
Home
FAQ
Mono
Runtime
Classes
Class Status
C# Compiler
Status
Contributing
Documentation
Test Suite
Tools
Download
Resources
Ideas
Passport
Contact
Ideas
Here are a few ideas of tools, classes and projects that you could start. More are forthcoming.
Runtime
We need a verifier that can be run on an executable (assembly) and tells whether the metadata for the executable is correct or not. It should report any anomalies.
For a list of anomalies in assemblies, check the various assertions that are described on the ECMA documentation.
This will help test our generated executables and can be also used as an external verifier.
Classes
TODO=jxta,The JXTA Peer to Peer foundation
Implement a JXTA protocol implementation: http://www.jxta.org
TODO=camel,Mail API
Implement a Mail API, similar to Camel or JavaMail (Camel has significant architecture features that are required on a real mailer).
You can check the current C Camel implementation.
Such an implementation could be used both with Microsoft .NET and Mono.
TODO=multimedia
Interfacing to Multimedia systems. You might want to look into the Quicktime API. I know Vladimir has researched the problem before
TODO=gtk,Gtk+ wrappers for Mono and .NET
Wrap the Gtk+ API. This is simple and can be done on Windows as Gtk+ 2.0 works on Windows.
This work can also be used on Windows and will enable developers on Windows to use some of Gtk+'s advanced features.
The idea is to wrap the Gtk+ API and allow us to build GUI applications using Gtk+ and in the future other Gtk+-based libraries from Mono (Gal, GtkHTML).
There is extensive knowledge on wrapping Gtk+ in other languages (this has been done this for Perl, Python, Java, Scheme, Haskel and other languages in the past).
Projects
Implement an xmlStorageSystem for the CLI: http://www.soapware.org/xmlStorageSystem
TODO=guavac,Java compiler for .NET
You could take one of the existing Java compilers (Guavac comes to mind as it is so nice) and modify it to generate .NET code rather than JVM byte codes.
This should be a pretty straightforward task. Guavac has the advantage of being written in C++ and it could be compiled with the Microsoft Managed C++ compiler and produce a .NET executable with it.
Document: Done


Unreal Tournament Editing - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://www.unrealtournament.com/editing/
MAIN PC PS2 DC LINUX MAC
Unreal TOURNAMENT
latest news announcements
New Maps by BadKarma
Pearman Alpha Released
Unreal Engine News for July 2001
Community Spotlights
latest file downloads
TouchSense Patch [260KB]
UT Patch v436 [7MB]
UT PS2 Footage [26MB]
UT Technology Movie [42MB]
Discussion
General
UT Matches
Rocket Arena
Chaos
Editing
Signup!
What's Hot
LAN Parties
Events
Buy It
Latest News
In the Press
Gallery
Gameplay
Tips & Tricks
Community
Behind the Scenes
Fan Alley
Downloads
Game Help
Editing & Scripting
European Web Site
EDITING & SCRIPTING
The greatest thing about the Unreal Engine Technology which Unreal Tournament uses, is that you can actually create your own levels, mods, skins, models, and more; making for an endless amount of new things to play with, and new things to see. Heck, people even get hired for doing these things, so give it a shot, you just might like it.
Below are links to various sections which give some detail on where to learn about editing, as well as a few tutorials and downloads to help get you started. Good luck!
Editing Resources
Editing Utilities
Console Commands
UnrealED Quick Reference
INFOGRAMES
EPIC GAMES
Copyright Info | Privacy Policy | E-Mail Webmaster | Questions and Comments
Document: Done

The Chandra Automatic Data Processing Infrastructure - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail Size Print Edit

Address http://adass.org/adass/proceedings/adass00/P1-32/ Go Links

The Chandra Automatic Data Processing Infrastructure

David Plummer and Sreelatha Subramanian
Harvard-Smithsonian Center for Astrophysics, 60 Garden St. MS-81, Cambridge, MA 02138

Abstract:

The requirements for processing Chandra telemetry are very involved and complex. To maximize efficiency, the infrastructure for processing telemetry has been automated such that all stages of processing will be initiated without operator intervention once a telemetry file is sent to the processing input directory. To maximize flexibility, the processing infrastructure is configured via an ASCII registry. This paper discusses the major components of the Automatic Processing infrastructure including our use of the STScI OPUS system. It describes how the registry is used to control and coordinate the automatic processing.

1. Introduction

Chandra data are processed, archived, and distributed by the Chandra X-ray Center (CXC). Standard Data Processing is accomplished by dozens of ``pipelines'' designed to process specific instrument data and/or generate a particular data product. Pipelines are organized into levels and generally require as input the output products from earlier levels. Some pipelines process data by observation while others process according to a set time interval or other criteria. Thus, the processing requirements and pipeline data dependencies are very complex. This complexity is captured in an ASCII processing registry which contains information about every data product and pipeline. The Automatic Processing system (AP) polls its input directories for raw telemetry and ephemeris data, pre-processes the telemetry, kicks off the processing pipelines at the appropriate times, provides the required input, and archives the output data products.

2. CXC Pipelines

A CXC pipeline is defined by an ASCII profile template that contains a list of tools to run and the associated run-time parameters (e.g., input/output directory, root-names, etc.). When a pipeline is ready to run, a pipeline run-time profile is generated by the profile builder tool, pbuilder. The run-time profile is executed by the Pipeline Controller, pctr. The pipeline profiles and pctr support conditional execution of tools, branching and converging of threads, and logfile output containing the profile, list of run-time tools, arguments, exit status, parameter files, and run-time output. This process is summarized in Figure 1.

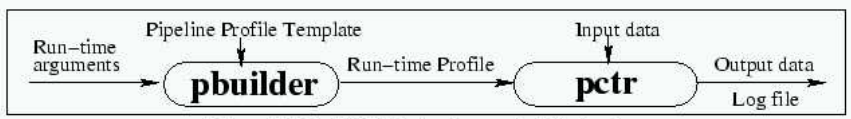


Figure 1: The CXC Pipeline Processing Mechanism.

Internet

The Chandra Automatic Data Processing Infrastructure - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail Size Print

Address http://adass.org/adass/proceedings/adass00/P1-32/ Go Links

4. Standard Pipeline Processing Threads

Figure 2 represents the series of pipelines that are run to process the Chandra data. Each circle represents a different pipeline (or related set of pipelines). Level 0 processing (De-commutation) will produce several data products that correspond to the different spacecraft components. Data from the various components of the spacecraft will follow different threads through the system. The arrows represent the flow of data as the output products of one pipeline are used as inputs to a pipe (or pipes) in the next level. Some pipelines are run on arbitrary time boundaries (as data are available) and others must be run on time boundaries based on observation interval start and stop times (which are determined in the level 0.5 pipe, OBI_DET).

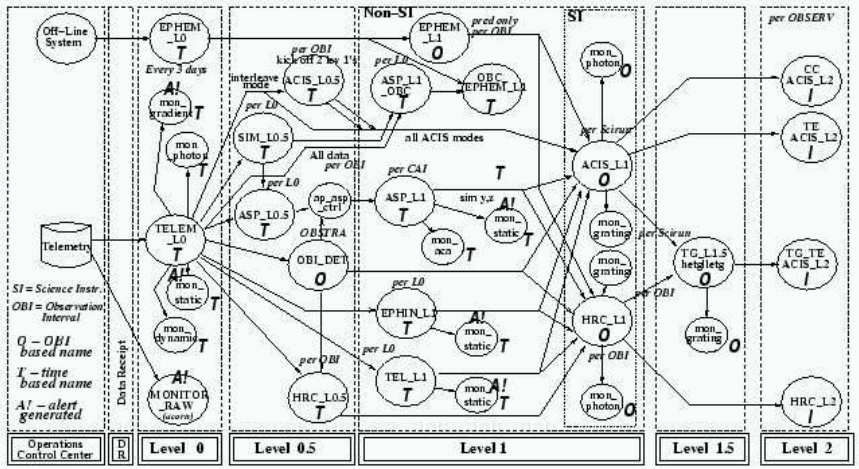


Figure 2: Standard Processing Threads.

5. Pipeline Processing Registry

The complete pipeline processing requirements for Chandra are very complex with many inter-dependencies (as can be seen in Figure 2). In order to run the pipelines efficiently in a flexible and automated fashion we configure the Automatic Processing system with a pipeline processing registry. We first register all the Chandra input and output data products. We can then capture the processing requirements and inter-dependencies by registering all the pipelines. Data products are registered with a File_ID, file name convention (using regular expressions), method for extracting start/stop times, and archive ingest keywords (detector, level, etc.). Pipelines are registered with a Pipe_ID, pipeline profile name, pbuilder arguments, kickoff criteria (detector in focal plane, gratings in/out, etc.), input and output data products (by File_ID), and method for generating the ``root'' part of output file names.

Internet

Open source processes for Requirements

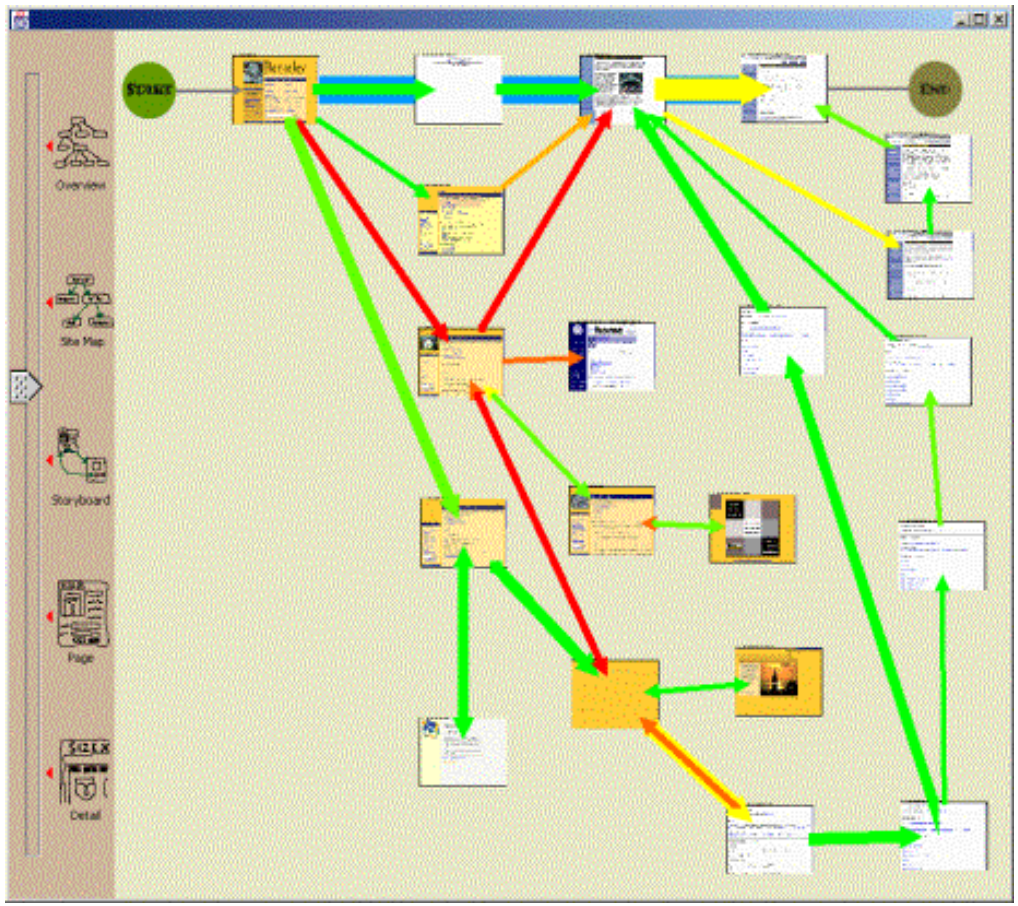
- Elicitation
- Analysis
- Specification and modeling
- Validation
- Communicating and managing
- *Post-hoc* assertion
- Reading, sense-making, accountability
- Continually emerging webs of discourse
- Condensing and hardening discourse
- Global access to discourse

Software Informalisms

- Community communications
  - Threaded discussion forums
  - Email (list servers)
  - Newsgroups
  - IRChat/Instant messages
  - Community digests (“Kernel Cousins”)

Software Informalisms

- *Scenarios of Usage* as linked Web pages



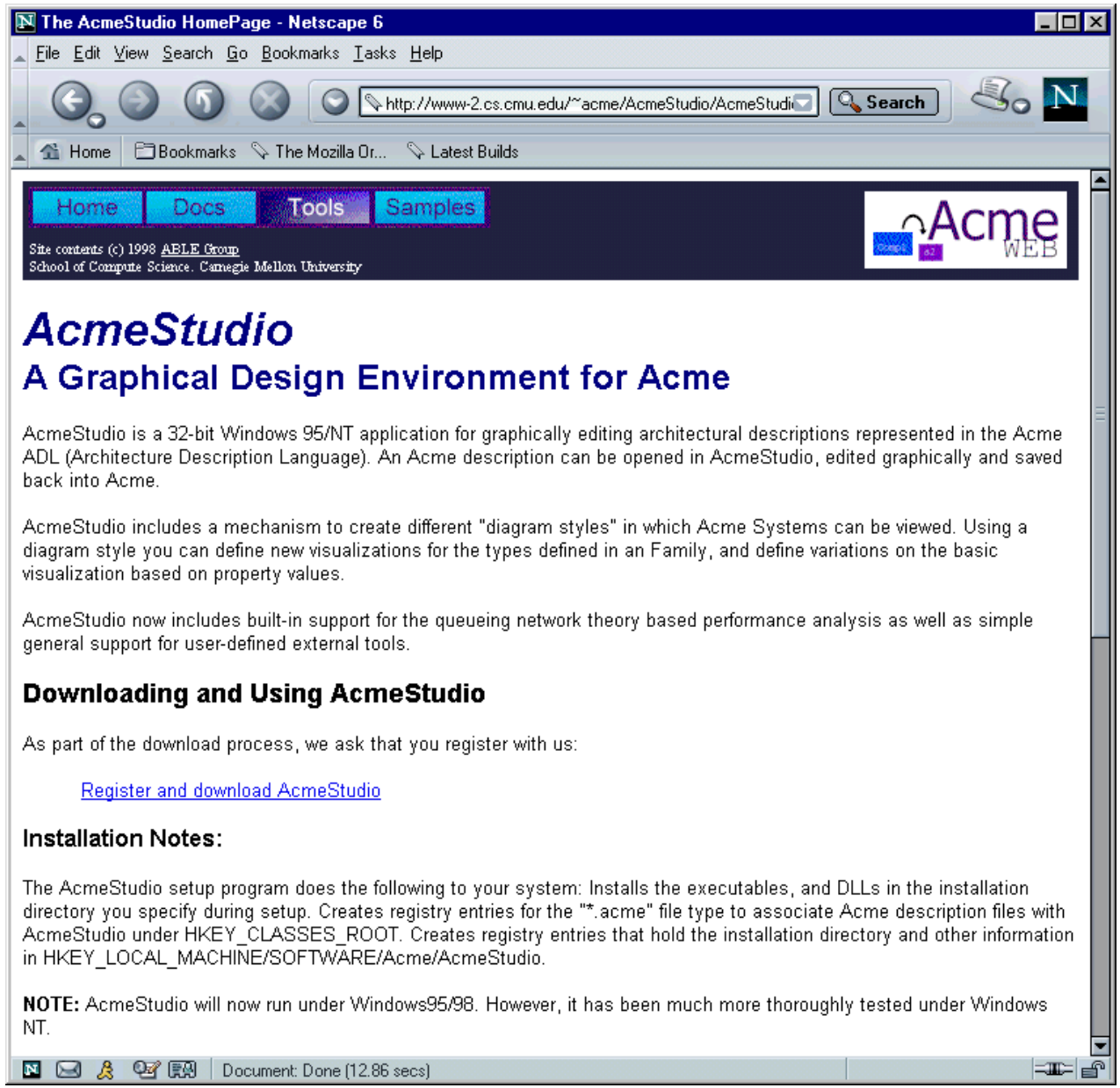
Software Informalisms

- How-To guides, To-Do lists, FAQs
- Traditional software user documentation
  - Unix/Linux man pages
- External publications
  - trade articles
  - scholarly research papers
  - books (cf. O'Reilly Books)

Software Informalisms

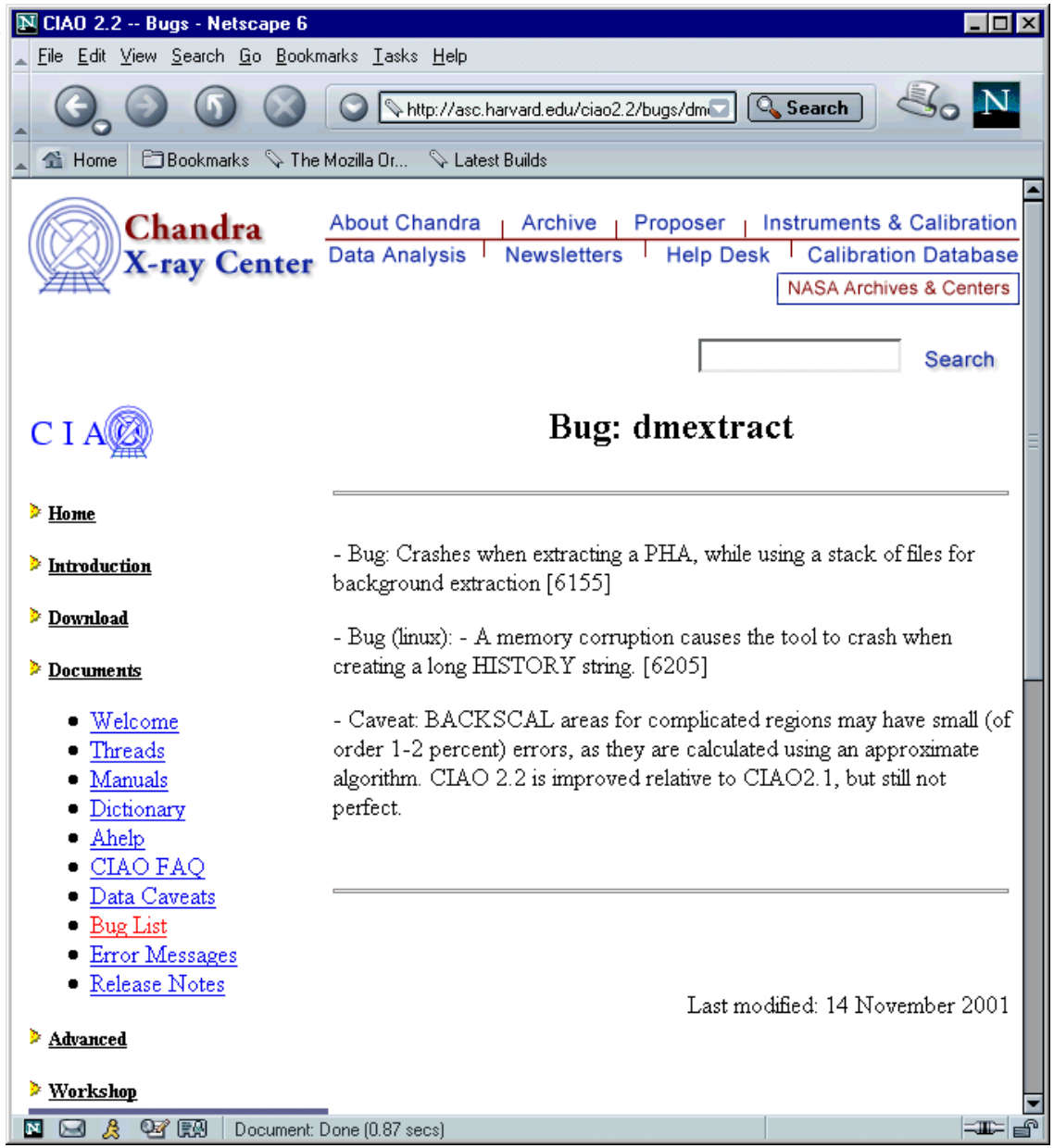
- Open Software Web Sites
  - Community Web sites
  - Community Software Web sites
  - Project Web sites
  - Source code Webs/Directories


SourceForge: Welcome - Netscape 6
File Edit View Search Go Bookmarks Tasks Help
http://sourceforge.net/
Search
Home Bookmarks The Mozilla Or... Latest Builds
SOURCEFORGE .net
my sf.net | software map | foundries | about sf.net | My Favorites
Login via SSL
New User via SSL
Search
Software/Group
Search
SF.net Resources
· Site Docs
· Site Status
· Site Map
· Compile Farm
· Project Help Wanted
· New Releases
· Contact Support
Most Active
1 Arianne RPG
2 Compiere ERP + CRM Business Solution
3 phpMyAdmin
4 phpAdsNew
5 JBoss.org
6 Gaim
7 PCGen -- A d20 Character Generator
8 FreeCraft real-time strategy game engine
9 DC++
10 MegaMek
Breaking Down the Barriers to Open Source Development
SourceForge.net is the world's largest Open Source development website, with the largest repository of Open Source code and applications available on the Internet. SourceForge.net provides free services to Open Source developers, including project hosting, version control, bug and issue tracking, project management, backups and archives, and communication and collaboration resources.
Create a new account to use all the features SourceForge.net has to offer. Learn more about our project development resources and how to register your project.
SourceForge.net is powered by SourceForge, the collaborative software development platform from VA Software.
SourceForge.net Development Foundries
SourceForge.net Foundries are topic-focused areas where developers connect and collaborate around software development.
SourceForge.net Statistics
Hosted Projects:
41,258
Registered Users:
436,845
SourceForge at Work
Bring SourceForge Enterprise Edition to your company.
SourceForge.net Newsletter
Email Address:
HTML Text
Subscribe me!
Most Active This Week
(100%) Arianne RPG
(99.9891%) Compiere ERP + CRM Business Solution
(99.9783%)
PC Components Clocks
A lovely wall mountable clock composed entirely of PC Components! Talk about geeky, these clocks will delight your friends and make you the envy of all nearby PC technicians...
www.thinkgeek.com
http://sfads.osdn.com/cgi-bin/adlog.pl?,tkgk5926en


The AcmeStudio HomePage - Netscape 6
File Edit View Search Go Bookmarks Tasks Help
http://www-2.cs.cmu.edu/~acme/AcmeStudio/AcmeStudi
Search
Home
Bookmarks
The Mozilla Or...
Latest Builds
Home
Docs
Tools
Samples
Site contents (c) 1998 ABLE Group
School of Compute Science. Carnegie Mellon University
Acme
WEB
AcmeStudio
A Graphical Design Environment for Acme
AcmeStudio is a 32-bit Windows 95/NT application for graphically editing architectural descriptions represented in the Acme ADL (Architecture Description Language). An Acme description can be opened in AcmeStudio, edited graphically and saved back into Acme.
AcmeStudio includes a mechanism to create different "diagram styles" in which Acme Systems can be viewed. Using a diagram style you can define new visualizations for the types defined in an Family, and define variations on the basic visualization based on property values.
AcmeStudio now includes built-in support for the queueing network theory based performance analysis as well as simple general support for user-defined external tools.
Downloading and Using AcmeStudio
As part of the download process, we ask that you register with us:
Register and download AcmeStudio
Installation Notes:
The AcmeStudio setup program does the following to your system: Installs the executables, and DLLs in the installation directory you specify during setup. Creates registry entries for the "*.acme" file type to associate Acme description files with AcmeStudio under HKEY_CLASSES_ROOT. Creates registry entries that hold the installation directory and other information in HKEY_LOCAL_MACHINE/SOFTWARE/Acme/AcmeStudio.
NOTE: AcmeStudio will now run under Windows95/98. However, it has been much more thoroughly tested under Windows NT.
Document: Done (12.86 secs)

Software Informalisms

- Software bug reports
  - Ad hoc report Web
  - Bugzilla (database tracking)
- Issue tracking
  - Issuezilla


CIAO 2.2 -- Bugs - Netscape 6
File Edit View Search Go Bookmarks Tasks Help
http://asc.harvard.edu/ciao2.2/bugs/dm
Search
Home
Bookmarks
The Mozilla Or...
Latest Builds
Chandra X-ray Center
About Chandra
Archive
Proposer
Instruments & Calibration
Data Analysis
Newsletters
Help Desk
Calibration Database
NASA Archives & Centers
Search
CIAO
Home
Introduction
Download
Documents
Welcome
Threads
Manuals
Dictionary
Ahelp
CIAO FAQ
Data Caveats
Bug List
Error Messages
Release Notes
Advanced
Workshop
Bug: dmextract
- Bug: Crashes when extracting a PHA, while using a stack of files for background extraction [6155]
- Bug (linux): - A memory corruption causes the tool to crash when creating a long HISTORY string. [6205]
- Caveat: BACKSCAL areas for complicated regions may have small (of order 1-2 percent) errors, as they are calculated using an approximate algorithm. CIAO 2.2 is improved relative to CIAO2.1, but still not perfect.
Last modified: 14 November 2001
Document: Done (0.87 secs)

Software Informalisms

- Software extension mechanisms
  - Inter-application scripting
    - Csh, Perl, Python, Tcl, scripting
    - Pipelines (cf. CXCDS)
  - Intra-application scripting (e.g., *UnrealScript*)
  - Plug-in architectures
    - Apache server architecture

Software Informalisms

- Open source software licenses
  - GNU Public License (GPL)
  - Lesser/Library GPL (LGPL)
  - Artistic License
  - Mozilla Public License (MPL)
  - SUN Public License (SPL)
  - and 25 more (http://opensource.org)
  - "Creative Commons" Project at Stanford Law School developing public license framework

Implications

- Software informalisms are the *media* of software requirements
- Software informalisms are the *subject* of software requirements
- OSS Requirements are *implied activities or capabilities*
- *(Re)reading* and *reviewing* informalisms is a prerequisite to writing open software

Implications

- Developing open software requirements is a *community building process*
  - not just a technical development process
  - open source peer reviewing creates a *community of peers*
- OSSD processes often iterate *daily* versus infrequent singular (milestone) SLC events
  - frequent, rapid cycle time (easier to improve) *vs*. infrequent, slow cycle time (hard to improve)

Implications

- Determining the quality of open software requirements:
  - not targeted to consistency, completeness, correctness
  - instead focusing attention to community building, freedom of expression, ease of informalism navigation (traceability), implicit vs. explicit informalism structuring

Conclusions

- Developing open software requirements is *different* than requirements engineering
  - not better, not worse, but different and new
  - more social, more accessible, more convivial
- Open source software systems don't need and probably won't benefit from classic software requirements engineering.

Conclusions

- Need to *integrate* OSSD with SE
  - development infrastructure (tools and environments)
  - development processes
  - developer community
  - across multiple domains
    - Scientific research
    - Commercial development

Conclusions

- People use OSS development tools to create, update, distribute, or browse OSS informalisms
- OSSD tool taxonomy:
  - Seven level hierarchy; more than 40 tool types
  - http://www.ics.uci.edu/~wscacchi/Software-Process/Open-Software-Process-Models/Open-Source-Software-Tools.html

Acknowledgements

- *Project collaborators*:
  - Mark Ackerman, UMichigan,
  - Margaret Ellliot, Ph.D., Mark Bergman, Xiaobin Li, UCI-ISR
  - Julia Watson, The Ohio State University
- *Funding support*:
  - National Science Foundation, IIS#-0083075, ITR#-#0205679
  - Defense Acquisition University, N487650-27803

References

- W. Scacchi, Understanding the Requirements for Developing Open Source Software, *IEE Proceedings--Software*, 149(1), 24-39, 2002. http://www.ics.uci.edu/~wscacchi/Papers/New/Understanding-OS-Requirements.pdf
- W. Scacchi, Exploring Open Source System Acquisition Processes and Architectures, Final Report, June 2002, http://www.ics.uci.edu/~wscacchi/ProjectReports/DAU-Final-Report-2002.pdf
- W. Scacchi, *Open EC/B*: A Case Study in Electronic Commerce and Open Source Software Development, Final Report, July 2002, http://www.ics.uci.edu/~wscacchi/ProjectReports/CRITO-Final-Report-2002.pdf
- W. Scacchi, Is Open Source Software Development Faster, Better, and Cheaper than Software Engineering?, *2nd Workshop on Open Source Software Engineering,* Orlando, FL, May 2002, http://www.ics.uci.edu/~wscacchi/Papers/New/ICSE02-Workshop-Scacchi-01.pdf