Bamboo Technology Software Documentation Review

1 Introduction
2 Overview of Software Documentation Review
3 Overall Documentation Organization and Introduction
4 Overall Documentation Organization and Introduction
5 4.1 Bamboo Services Platform (BSP) Hosted Services - UC Berkeley
5 4.2 Scholarly Services - Tufts University and University of Chicago (Proxied Access)
5 4.3 Fedora Repository CMIS Adaptor - UW Madison and Indiana University (Research Environments)
5 4.4 Content Interoperability (CI) Hub - Northwestern University and UI Urbana-Champaign (Interoperability)
5 4.5 Research Environment Modules - UW Madison and Indiana University (Research Env -> Tool Integ, Client Env, Manage groups)
5 4.6 Bamboo DIRT - UC Berkeley
6 WIKI Documentation of APIs
6 5.1 Bamboo Services Platform (BSP) Hosted Services - UC Berkeley
6 5.2 Scholarly Services - Tufts University and University of Chicago
6 5.3 Fedora Repository CMIS Adaptor - UW Madison and Indiana University
6 5.4 Content Interoperability (CI) Hub - Northwestern University and UI Urbana-Champaign
6 5.5 Research Environment Modules - UW Madison and Indiana University
6 6.1 Bamboo Services Platform (BSP) Hosted Services - UC Berkeley
6 6.2 Scholarly Services - Tufts University and University of Chicago
6 6.3 Fedora Repository CMIS Adaptor - UW Madison and Indiana University
6 6.4 Content Interoperability (CI) Hub - Northwestern University and UI Urbana-Champaign
6 6.6 Bamboo DIRT - UC Berkeley
7 Architecture, Design and Code Documentation and Management
7 7.1 Bamboo Services Platform (BSP) Hosted Services - UC Berkeley
7 7.2 Scholarly Services - Tufts University and University of Chicago
7 7.3 Fedora Repository CMIS Adaptor - UW Madison and Indiana University
7 7.4 CI Hub - Northwestern University and UI Urbana-Champaign
7 7.5 Research Environment Modules - UW Madison and Indiana University
7 7.6 Bamboo DIRT - UC Berkeley
8 Issue Management and Tracking
8 8.1 Bamboo Services Platform (BSP) Hosted Services - UC Berkeley
8 8.2 Scholarly Services - Tufts University and University of Chicago
8 8.3 Fedora Repository CMIS Adaptor - UW Madison and Indiana University
8 8.4 Content Interoperability (CI) Hub - Northwestern University and UI Urbana-Champaign
8 8.5 Research Environment Modules - UW Madison and Indiana University
8 8.8 Bamboo DIRT - UC Berkeley

Introduction

This document reviews the existing documentation for the software components of the Bamboo Technology platform. This review focuses on the existing software documentation and the associated documentation required to install, configure, and deploy underlying technology and the software itself. The scope of this review does not include end-user documentation, except insofar as Service API documents aimed at developers who consume services via their API fit a definition of “end user.” A Statement of Work for this review exists at the following wiki page: Phase One Code Review - Statement of Work.

Overview of Software Documentation Review

Starting at the Project Bamboo website’s home page at http://www.projectbamboo.org/, I quickly found the most current Documentation page. These wiki pages gave me a good high-level overview of the Bamboo Technology Platform, its core components, and its overall goals.

Overall Documentation Organization and Introduction

This section is a review of each Bamboo technology component's overall technical documentation. It is meant to assess the organization and clarity of documentation and suggest improvements, if any, that are needed.

Bamboo Services Platform (BSP) Hosted Services - UC Berkeley

The landing page on the wiki for BSP at (Bamboo Services Platform) gives a good set of links to get a high overview of this Bamboo technology component and to learn about its goals and purpose. Specifically the wiki page at Architectural Overview of the Bamboo Services Platform gives an excellent overview of how BSP fits into the larger goals of Scholarly Services and the technologies that BSP uses to help achieve those goals.

Scholarly Services - Tufts University and University of Chicago (Proxied Access)

The page "Proxied Access to Remotely Hosted Tools for Scholarship" gives a good overview of the "Proxied Services" support in Project Bamboo. It describes the high-level purpose of the services and includes specific example services:

- a morphological analysis service, operated by the Perseus project at Tufts University
- a syntactic annotation repository, operated by The Alpheios Project
- philology services run by the ARTFL project at University of Chicago

The page also includes useful links to service APIs and to the currently deployed and live services. As an aside, I would have liked this page to include hyperlinks to things like http://alpheios.net/content/about-alpheios, http://nlp.perseus.tufts.edu/hopper/about, and the other stated example services.
Fedora Repository CMIS Adaptor - UW Madison and Indiana University (Research Environments)

The Bamboo Book Model and the corresponding Fedora CMIS adaptor are introduced and documented at Bamboo Book Model - CMIS Binding and Fedora Repository Implementation. The text gives a detailed overview of Bamboo's book model and its sub-components. As someone new to Bamboo's Book Model, I found the introductory text somewhat abstract and would have appreciated a higher level introduction to the book model's purpose and goals as well as some examples and typical uses.

Content Interoperability (CI) Hub - Northwestern University and UI Urbana-Champaign (Interoperability)

The wiki page Interoperability of Digital Collections provides the purpose of the CI Hub and provides links to pages giving more details of how it is implemented and how it fits into the Project Bamboo ecosystem.

Research Environment Modules - UW Madison and Indiana University (Research Environments to Store Manipulate and Manage Digital Content)

The wiki page Research Environments to Store Manipulate and Manage Digital Content gives a good overview of Project Bamboo's "Research Environments" (aka, "Workspaces"). It also includes a useful history of and details the evolutionary development of the Research Environments. A very useful set of links at the bottom of the page provide more detailed and background information.

Bamboo DiRT - UC Berkeley

The landing page for DiRT (see section) gives a well organized and brief overview of the DiRT tool and its purpose and goals. It would be useful if it included a link to the DiRT website (http://dirt.projectbamboo.org/). Since it didn't, I needed to go searching for it.

Also the title for the section named "How To build a similar directory site using Drupal" was confusing to me. In the main landing page, there was no indication that the DiRT was implemented using Drupal so I assumed the section "How To build a similar directory site using Drupal" was giving instructions on how to create a DiRT instance with an alternate technology tool set. I think a different title (something like "How DiRT was implemented using Drupal" or just "How DiRT was/is implemented") would be less confusing to the reader.

WIKI Documentation of API's

This section is a review of each Bamboo technology component's documentation for its existing public and/or private API's. It is meant to assess the organization and clarity of documentation and suggest improvements, if any, that are needed.

Bamboo Services Platform (BSP) Hosted Services - UC Berkeley

The BSP provides a set of RESTful HTTP service API's that are designed for client/external use. The landing page for the BSP API's (Service APIs - Centrally-Hosted Bamboo Services) is fairly sparse and I would have preferred at least a small introductory high-level statement and/or overview of the platform. The landing page provides a set of links to each of the existing Services' home page for documentation. Each of the Service home pages follow the same useful structure. The home pages include things like a brief description of the service, a set of use cases, outstanding questions, etc. The detailed API descriptions are laid out for each Service on its "Contract Description" page -a link to which is provided on each Services' home page.
The landing page for the BSP API's (Service APIs - Centrally-Hosted Bamboo Services) is very sparse. I would have preferred at least small introductory high-level statement and/or overview of the platform. A division and logical grouping of the almost two dozen services on the landing page (Bamboo Services Platform) would help readers navigate the page more quickly.

### Scholarly Services - Tufts University and University of Chicago

The wiki based API's for the scholarly services were consistently organized. The Morphological Analysis and Syntactic Annotation services were both fairly well documented, but could perhaps have included more descriptive text. Also, both the Morphological Analysis and Syntactic Annotation services include broken links to service response schemas. For example:


The documentation for Philologic service was minimal and included very little (if any) descriptive text and examples. All of the scholarly services wiki API documentation included references to source code documents. An example:

$\text{REPOSITORY\_ROOT}/sandbox/christensen/trunk/philologic-service/resource/src/main/resources/CollectionsResponse.xsd

Updating these reference to actual hyperlinks to the source code repository would be helpful.

### Fedora Repository CMIS Adaptor - UW Madison and Indiana University

I found the wiki documentation for the CMIS adaptor fairly well organized and useful. It gives a good description of how the Bamboo Book model is bound to CMIS. It provides a useful table show the various Bamboo object types, their extending class, and their corresponding properties. The wiki pages also give a useful description of the related CMIS permissions as well as a rationale for additions beyond the rudimentary 0.5.0 OpenCMIS server permission support. As expected, I found links to both the CMIS implementation code and corresponding JavaDoc pages on the wiki.

### Content Interoperability (CI) Hub - Northwestern University and UI Urbana-Champaign

The wiki page Collection Interoperability Hub (CI Hub) architecture and implementation details the implementation of the CI Hub. It provides a good explanation of how CI Hub is implemented using the Apache Chemistry OpenCMIS including a description of the CI-Hub's ROA layer, SOA layer, and Locator Extensions.

### Research Environment Modules - UW Madison and Indiana University

From what I could gather from the documentation, the Research Environments appears to be an integration of various tools and services. Some of these were part of Project Bamboo and some from outside organizations. It appears that the integration points for the key components of the Research Environments had good documentation. For instance, the documentation I found for several of the integration points was helpful - e.g., Tool integrations in Research Environments, and Tool integration atop the OpenSocial API.

### Bamboo DiRT - UC Berkeley

Not applicable.

### Documentation for Installation, Configuration and Deployment

This section is a review of each Bamboo technology component's documentation for installation, configuration, and deployment. It is meant to assess the organization and clarity of documentation and suggest improvements, if any, that are needed.
The BSP has an excellent set of documents for installing, configuring, and deploying the BSP platform at Centrally-Hosted Bamboo Services - Development-Deployment-Invocation. The documentation is thorough and very informative giving detailed instructions for each of the major components.

**Scholarly Services - Tufts University and University of Chicago**

The only necessary installation and configuration documentation for the Scholarly Services relates to the required BSP proxies all of which is thoroughly documented on the developer workbench page (https://wikihub.berkeley.edu/x/IgNRB).

**Fedora Repository CMIS Adaptor - UW Madison and Indiana University**

I was unable to find any documentation for installing, configuring, and deploying the Fedora CMIS Adaptor.

---

**Content Interoperability (CI) Hub - Northwestern University and UI Urbana-Champaign**

The wiki page Collection Interoperability Hub (CI Hub) architecture and implementation contains a "Configuration" section that seems to contain a minimal description of how to configure the CI Hub.

**Research Environment Modules - UW Madison and Indiana University**

I was able to find at least some documentation related to installation, configuration and deployment for the key Research Environment components. However not all of the documentation appeared to be complete or contain sufficient details.

**Bamboo DiRT - UC Berkeley**

The page currently titled "How To build a similar directory site using Drupal" appears to give thorough, concise, and well organized instructions for how DiRT was implemented using Drupal. Some links to Drupal and Drupal setup would be helpful and convenient to me and the other readers. Since they were not provided, I was left to searching the web (as would other readers not familiar with Drupal) for the proper information and links.

---

**Architecture, Design and Code Documentation and Management**

This section is a review of each Bamboo technology component's documentation for its architecture and design as well as a review of each component's existing code documentation. Is is also a review of each component's source code management -e.g., version control, branching, tagging, commit comments, etc.

**Bamboo Services Platform (BSP) Hosted Services - UC Berkeley**

**Developer's Documentation**

The main landing page for developer's wishing to create services that will integrate with and leverage the BSP was well organized and the information was helpful and complete -see [Developer Workbench Environment for BSP Service Developers](https://wikihub.berkeley.edu/x/IgNRB).

**Source Control**

The source tree at http://svn.code.sf.net/p/projectbamboo/code/platform-services/bsp/trunk/ is structured and logically organized. The code is properly archived and maintained in the Subversion software versioning and revision control system. A top-level "README. txt" file explaining the structure and components would be helpful to a beginner. As far as I could tell, the code in the Subversion repository complied with Bamboo's software version control policy -see [Software Version Control Policy](https://wikihub.berkeley.edu/x/IgNRB). Looking at the Subversion commit logs, there are frequent and well commented code commits to the repository. Most commits are properly annotated with a specific JIRA issue which often helps give more details and context about the committed code.
The BSP sources are built using the Apache Maven build automation tool. Though undocumented, the Maven project files (pom.xml files) look well organized. However, it was difficult for me to recognize and understand the meaning and purpose of several of the Maven modules/projects. In these cases, "README.txt" files in the module's top-level directory would be very helpful.

**Javadocs and Inline Comments**

Most of the class files labelled in "service" packages did contain Javadoc comments. Javadoc comments are present in most but not all of the other class files. Inline comments were sparse but usually helpful and to the point.

**Scholarly Services - Tufts University and University of Chicago**

**Source Control**

The Morphological Analysis and Syntactic Annotation services source code is available at `http://svn.code.sf.net/p/projectbamboo/code/platform-services/`. The source code for these two services was very sparsely commented but fortunately it is well formatted, uses meaningful member and method names and generally easy to read and understand.

**Build Management**

The Morphological Analysis and Syntactic Annotation services use standard Maven build files and conventions. I was able to build both of the services without any problems.

**Javadocs and Inline Comments**

The javadocs for the Morphological Analysis and Syntactic Annotation are available at `http://javadoc.projectbamboo.org/bsp/`. Very little descriptive text is present in the Javadocs. In a random selection of packages, classes, and methods, I found no descriptive annotations or comments.

**Fedora Repository CMIS Adaptor - UW Madison and Indiana University**

**Source Control**

The source code for the CMIS adaptor implementation is properly archived on a Subversion instance at `http://svn.code.sf.net/p/projectbamboo/code/work-space-repository/`. I would have liked to have seen a high-level README at the top of the source tree explaining the three top-levels (cmisclient, fedora, fedora_cmis_repository) and how they related. The Java code seemed to be well organized and was quite easy to read though the comments we very terse and minimal.

**Build Management**

I did not try to build "fedora-cmis-repository" trunk, it appeared to have a standard and straightforward Ant build script. However, I did not see any versioning infrastructure in the Ant build script for the "fedora-cmis-repository" project. Nor did I see any versioning information in any of the source files for the "fedora-cmis-repository." project.

**Javadocs and Inline Comments**

Javadocs for the "fedora-cmis-repository" are available at `http://javadoc.projectbamboo.org/fedora-cmis/`. Unfortunately, they included very little high-level package, class, and method descriptions.

**CI Hub - Northwestern University and UI Urbana-Champaign**

**Source Control**

The CI-Hub source code is organized into ROA and SOA source trees and includes Java, Scala, as well as Spring XML files. All source code, with the exception of the `cihub.properties` configuration file, is relative to cihub-service/ (location in Bamboo’s Sourceforge code repository).

**Build Management**

The CI-Hub's ROA and SOA layers seem to have up-to-date working Maven build script committed to their source tree at `http://svn.code.sf.net/p/projectbamboo/code/platform-services/ci-hub-service/trunk/ci-hub-service/`

**Research Environment Modules - UW Madison and Indiana University**

**Source Control**

The following are links to the Project Bamboo source code repository on Sourceforge:

- Fedora-CMIS interface
- Repository Browser interface (for Drupal)
- Alfresco integration with UC3 Merritt and DataCite

Of these three modules, the Alfresco related code contained the most comments and inline documentation. Reviewing a random selection of source files from the other two modules revealed very few comments and little descriptive text.

**Build Management**
Where applicable, Maven and Ant scripts seems to be available to build Research Environment java-based components. For the PHP related components, a README file at http://svn.code.sf.net/p/projectbamboo/code/work-space-drupal/repository_browser/README.txt gives a description of the artifacts.

Javadocs and Inline Comments

Javadocs for the Fedora CMIS adaptor exist at http://javadoc.projectbamboo.org/fedora-cmis/.

Bamboo DiRT - UC Berkeley

Having no experience developing and maintaining a Drupal instance made it impossible for me to evaluate DIRT's architecture and design. Someone with Drupal development experience and access to the backend Drupal instance should be consulted to give a proper evaluation of DIRT's architecture, design and code documentation.

Source Control

As far as I could tell, DIRT uses no formal VCS (version/revision control system) for its implementation artifacts. Having no experience with Drupal development, I can't say whether or not this would be a best practice that DIRT should be following.

Build Management

Having no experience developing, maintaining, and deploying Drupal instances, I do not know if there are related best practices in this category that DIRT should be following.

Javadocs and Inline Comments

Having no experience developing, maintaining, and deploying Drupal instances, I do not know if there are related best practices in this category that DIRT should be following.

Issue Management and Tracking

This section is a review of each Bamboo technology component's documentation for how it uses issue management and issue tracking software.

Bamboo Services Platform (BSP) Hosted Services - UC Berkeley

The BSP appears to have made good use of a JIRA instance at https://projectbamboo.atlassian.net to manage and track both issues and development tasks. Issues are well described and annotated. Most of the issues are properly categorized and contain time tracking estimates.

Scholarly Services - Tufts University and University of Chicago

The Morphological Analysis and Syntactic Annotation services appear to have made use the JIRA instance to keep track of tasks, issues, and specific versioned releases. The JIRA issues were usually well described, commented, and tracked.

Fedora Repository CMIS Adaptor - UW Madison and Indiana University

The Fedora CMIS Adaptor development team appears to have made very little use of the project's JIRA instance.

Content Interoperability (CI) Hub - Northwestern University and UI Urbana-Champaign

The CI Hub development team appears to have made very little use of the project's JIRA instance.

Research Environment Modules - UW Madison and Indiana University

The Research Environment development teams appears to have made very little (if any) use of the project's JIRA instance.

Bamboo DiRT - UC Berkeley

I was able to find very little evidence that DIRT used an issue management and tracking tool. I found only a handful of issues on the Project Bamboo instance at https://projectbamboo.atlassian.net.