Purpose and Focus of Document

The purpose of this document is to collect, analyze, and define high-level needs and features of the NCICB caDSR Data Warehouse 1.0 release. This document focuses on the functionalities proposed by the product stakeholders and target users in order to make it a better product. The use-case and supplementary specifications document will detail how the framework will fulfill these needs.

Vision and Dependencies

Vision or Problem Statement

caDSR data is stored in a transactional database schema that adheres to the ISO 11179 standard. However, the types of searches that users want to perform against the caDSR database are complex and driven by a wide variety of needs. The complex data structures of a transactional system are usually not well suited for optimizing searches and result is long wait times while underlying queries perform searches and joins to resolve associations and collections in order to present the caDSR information in various views.

The recommended solution is to use proven data warehousing technologies consisting of storing all necessary data for optimized searches. The data is organized in one or more “star” schemas. Each star schema essentially corresponds to a domain specific data mart and should meet a large set of needs for a community.

The mature technologies of data warehousing can be used without concern for supporting transactional integrity in the same schema. Derived data, additional data, duplicated data, all can be planned to optimize the search capabilities.

Stakeholder, Technical and User Descriptions
Stakeholder Summary

<table>
<thead>
<tr>
<th>Customer Name</th>
<th>Role</th>
<th>Interest / Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>Denise Warzel</td>
<td>NCI CBIIIT Core Product Line Manager/ MDR Product Manager</td>
<td>---</td>
</tr>
<tr>
<td>Dave Hau</td>
<td>Associate Director of Core Infrastructure Engineering</td>
<td>---</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NCICB Staff/Contractor Name</th>
<th>Role</th>
<th>Responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charles Griffin</td>
<td>Project Manager</td>
<td>---</td>
</tr>
<tr>
<td>Denis Avdic</td>
<td>Technical Architect</td>
<td>---</td>
</tr>
<tr>
<td>Nadine Azie</td>
<td>Data Architect</td>
<td>---</td>
</tr>
</tbody>
</table>

Technical Environment

This product uses the following technical components which have been derived from the current NCICB Technology Stack.

- **Client Interface**
  - Mozilla v. 2.0.0.1 and above
  - IE Browser 7.0
- **Application Server**
  - Jboss 4.0.5
- **Database Server**
  - Oracle 10g
- **Operating System (Warehouse APIs)**
  - Linux
  - Windows 32 bit

Current Solution to Meeting Needs

At the time of writing, caDSR offers a variety of web applications that users can use to search and view caDSR content. The CDE Browser and the UML Browser are the primary user facing tools that are used for searching caDSR content. These tools search against the the transactional caDSR database.

caDSR also provides programmatic client APIs based on the caDSR domain model that application programmers can utilize to query the caDSR database. The client APIs make calls to the remote caDSR server API hosted at NCI CBIIIT. The caDSR Server APIs are generated by the caCORE SDK system and thereby have the same architecture of a caCORE SDK generated system.

Proposed Solutions to Meeting Needs

The search requirements for the system are complex and driven by a wide variety of needs. The complex data structures of a transactional system are usually not well suited for optimizing searches. he desire to enable semantic driven searches, combining vocabulary, metadata and real data searches can be accommodated by designing the data structures explicitly for that purpose. The LexBIG terminology server in use by EVS is the most likely candidate for publishing caDSR content for this purpose since the data structures are already designed to support queries across OWL structures. The recommended solution is to use proven data warehousing technologies to improve searches for the primary use cases, CDE Browser, UML Model Browser and other caDSR specific searches, and in later iterations explore the OWL representation for specific analytical search engines if LexBIG does not prove fruitful.

The data warehouse approach consists of storing all necessary data for optimized searches. The data is organized in one or more "star" schemas. Each star schema essentially corresponds to a domain specific data mart and should meet a large set of needs for a community.

The mature technologies of data warehousing can be used without compromise for the need to support transactional integrity in the same schema. Derived data, additional data, duplicated data, all can be planned to optimize the search capabilities.

The data warehouse should be part of the overall run time repository to facilitate access by all applications. Many of the tools that currently access the caDSR transactional database should switch their search capabilities to the optimized query services and supporting datamarts. Additionally new tools that will be created to meet the expanding search and query needs of users should utilize the data warehouse.

In this iteration, we will be testing to load scenarios for the data warehouse: (1) ETL for the a standalone data warehouse and (2) oracle technology for providing materialized views.

The stand alone data warehouse instance will be populated by performing ETL (Extract Transform and Load) procedures written as stored procedures using PL/SQL. Another instance of the same data warehouse schema will be provided as Oracle Materialized Views that utilized Oracle specific technology to update the warehouse schema in lieu of the ETL process.

Performance tests will be run against both data warehouse schemas (standalone and Materialized Views) to determine the the plus and minuses of the two approaches.

Product Dependencies
This release is dependent on the caCORE components or products documented on the dependencies wiki page (example for caCORE 3.2).

[Provide additional explanation as applicable. For example, “The EVS vocabulary systems are used by the Java client to retrieve and validate concept information for naming and defining meanings.”]

Summary of Key Stakeholder or User Needs - Stakeholder and User Requirements

The following subsections provide a description of key requirements to address the solution as perceived by the stakeholders and users.

**Functional**

- Make Enhancements to the iteration 1 prototype warehouse schema based on new use cases/search patterns
- Implement the warehouse as materialized views
- Perform performance tests on standalone warehouse and materialized views and compare the results
- Generate APIs for applications and users to utilize the warehouse

**Non-Functional**

- Identify new use cases and search patterns that the data warehouse needs to support
- Identify a tool that can be refactored to utilize the warehouse as a reference implementation

In-Scope Requirements and Enhancements

**In-Scope Functional Requirements (Enhancements or New Features) - (Proposed)**

Each new enhancement, modification or new feature is described in detail below.

**Standalone Warehouse Schema**

GF#14268 Modify the Data Warehouse design based on the defined and approved Use Cases (new search patterns)

GForge link

GF# 14270 Perform schema enhancements based on the design

GForge link

**Materialized Views**

GF#14271 Create Materialized views for iteration 1 data warehouse schema

GForge link

GF#14272 Modify the materialized views based on iteration 1 schema for the iteration 2 use cases (search patterns)

GForge link

**Performance Test**

GF#14280 Identify and document a performance testing strategy for the warehouse

GForge link

GF#14281 Execute the performance tests on the stand alone data ware house

GForge link

GF#14282 Execute the performance tests on the materialized views

GForge link

GF#14283 Document the comparison of performance tests between materialized views and the standalone schema based on a real use case that will be implemented and included in the deployment of the data warehouse

GForge link

**Warehouse APIs**

GF#14279 Create a domain model for the warehouse schema(s)

GForge link

GF#14278 Create an SDK generated system based on the warehouse domain model and data model mappings
In-Scope Functional Bug Fixes

None

In-Scope Non-Functional Requirements - (Proposed) - Requirements/Use Cases (Search Patterns)

This section describes in detail all the non-software related requirements which must be met for this release but do not add functionality. These requirements are included in the scope and project plan due to level of effort or relative importance to the overall success of delivery of the release.

GF#14254 Identify Uses Cases (Search Patterns) that will be the basis for enhancements to the iteration 1 data warehouse schema or creation of new data marts

GF#14267 Identify a caDSR tool that can be refactored to utilize the data warehouse as a proof of concept

In-Scope General Support Activities

None

Out of Scope Requirements and Enhancements

Out of Scope Functional Requirements (Enhancements or New Features)

Items that are out of scope were evaluated as part of the initial scoping activities for this release, and subsequently not included in the final approved scope. These items are also documented in the cumulative backlog of requirements found on the product GForge site.

None

Out of Scope Functional Bug Fixes if Applicable

None

Out of Scope Non-Functional Requirements

None

Out of Scope General Support Activities

None

Document History and Project Information

<table>
<thead>
<tr>
<th>Document Version:</th>
<th>Click the Info tab. View the Recent Changes or click the link to view the page history.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Modified:</td>
<td>Refer to the first line displayed in the document window.</td>
</tr>
<tr>
<td>Project GForge site:</td>
<td>[caCORE:Project GForge site link]</td>
</tr>
<tr>
<td>Most current version:</td>
<td>Unless the display includes a notice that you are viewing a previous version, you are viewing the most current version of this Scope Document for the release indicated in the title.</td>
</tr>
<tr>
<td>Revision history:</td>
<td>Click the Info tab. In the Recent Changes area, click the link to view the page history.</td>
</tr>
<tr>
<td>Review history:</td>
<td>Click the Info tab. In the Recent Changes area, note the developer who made each change and the date and time. Refer to the Key People Directory for their roles. Click the link to view any page or to view the page history, and then click the link for a page. When the page opens, view the comments and changes made in that version.</td>
</tr>
<tr>
<td>NCICB Management</td>
<td>Role</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>Denise Warzel</td>
<td>caDSR Product Manager and caCORE Product Line Manager</td>
</tr>
<tr>
<td>Dave Hau</td>
<td>caDSR Engineering Manager</td>
</tr>
</tbody>
</table>