LexEVS 6.4 Software Design Document

Document Information

Author: Craig Stancl, Scott Bauer, Cory Endle
Email: Stancl.craig@mayo.edu, bauer.scott@mayo.edu, endle.cory@mayo.edu
Team: LexEVS
Contract: S13-500 MOD4
Client: NCI CBIIT
National Institutes of Heath
US Department of Health and Human Services

Contents of this Page

The **purpose of this document** is to collect, analyze, and define high-level needs for and designed features of the National Cancer Institute Center for Biomedical Informatics and Information Technology (NCI CBIIT) **LexEVS Release 6.4**.

The focus is on the functionalities proposed by the stakeholders and target users to make a better product.

Design Scope and Requirements

The LexEVS 6.4 Scope Document can be found here: LexEVS 6.4 Scope Document

Detailed Design

The following sections specify how the design will satisfy the requirements for the Lucene search upgrade. This design reflects the wide ranging changes that will be necessary to LexEVS to fully update over three full releases of Lucene. Since Lucene is the heart of the search mechanism that powers efficient searches in LexEVS these changes are necessarily extensive. The focus of these changes can be broken down, to some extent, into three areas.

- Code decoupling from the current Lucene to allow for easier updates to the underlying search implementation.
- Multi-index searches to replace single index searches. This will allow easier maintenance than the large, monolithic index we currently use.
- Using the built in relational index structures of the latest Lucene release to replace the hand built version of this in the current LexEVS implementation.

- Code refactoring to the latest Lucene code base. This requires extensive changes to the code base including replacement of objects with similar behavior for the current code base and adjusting to changes in the Lucene API. This also includes reimplementing a number of customized Lucene analyzers and HitCollectors to insure compatibility with current code unit tests and user expectations.

Some classes are called out to indicate the extent of the changes and to document some of the details of intended adjustments.

Code Decoupling

LexEVS Code Decoupling

Multi-Index Searches

LexEVS Multi-index Lucene Implementation

Changing the Relational Representation in Lucene

LexEVS Lucene Relational Representation

Per-segment Search

LexEVS Per-segment Search Implementation

Hit Collector to Collector Updates

LexEVS HitCollector Update

Analyzer Updates

LexEVS Analyzer Updates

Index Code Refactor: Removing the Indexer Project

LexEVS Indexer Refactor

Audit of Text Matching Use Cases

LexEVS Text Match Algorithm Audit

Consideration of Build Impacts

Consideration of Build Impacts

Index Compatibility with Previous Index Versions

This update will be to Lucene 5.0. This version of Lucene has some backwards compatibility with Lucene 4.0 indexes. However it is completely incompatible with indexes versions 3.x and earlier. While LexEVS API's will maintain backwards compatibilities, any indexes from previous installed will have to be re-indexed in the latest implementation.

Pagination in Lucene

Lazy loading pagination is a broad concept in LexEVS and can encompass both graph and node set capabilities. Because this scope is large we are going to consider this out of scope for this project unless we can define a fairly narrow definition of what we want to do with Lucene's version of this. Currently some lazy loading occurs under the covers in the iterators returned by the coded node set implementation. We also have node graph pagination. In either case we may not need a reimplementation in order to update our Lucene implementation. We are suggesting this become a possible priority for a later implementation and won't fully describe how this might be done here.

Impacts to Existing Users / Applications

Minimal impact overall expected with increased performance and maintenance efficiency expected.

Impact Description	Reference to documented impact
Text Matching Algorithm Changes. Support for a wide ranging text matching capability creates potential for heavy maintenance. We have attempted to characterize the similarities between some term matching implementations with an eye towards exclusion or combination. This exclusion and combination only affects end users if we remove labels as algorithm switches.	Not specifically impact documentation but a background document: Lex EVS Text Match Algorithm Audit

I**ndex File System Changes.** Index files will exist per coding scheme. This creates the opportunity for unmerged terminology indexes that should improve maintenance efficiency through quicker load times and the ability to identify and remove broken indexes without having to reindex the entire service. This will change the appearance of the file system but should not cause any issues for end users. The API will remain the same.	Background: LexEVS Multi-index Lucene Implementation
Faster Query Performance on at Least Some Queries. The goal is to at least make queries no slower than current queries. The use of Block Join Queries has a reputation for being faster. This implementation has some opportunity to provide small indexes if we can properly capitalize during implementation.	Background: LexEVS Lucene Relational Representation
Index Optimization Function Will Go Away. Index optimization no longer serves the purpose originally intended in Lucene. The optimization function should be deprecated and the implementation changed to output a message that indexes no longer need optimizing.	

Decision Points - Approval Needed

Pagination in Lucene - DEFERRED

Reference: LexEVS 6.4 Software Design Document

Sign off	Date	Role	CBIIT or Stakeholder Organization	Approver's Comments (If disapproved indicate specific areas for improvement.)
Larry Wright	4/24 /2015	Govt Project Manager	CBIIT EVS	---
Sherri de Coronado	4/30 /2015	Govt Sponsor	CBIIT EVS	—
Kumar Kuntipuram	4/30 /2015	TPM	Leidos Biomed	—

Reduction of Text Matching Algorithms

Reference: LexEVS Text Match Algorithm Audit

Text Matching Algorithms to be continued: This list to be provided when the development team begins to work on these algorithms.

- Consideration needs to be given to "Contains" search as it doesn't currently behave correctly. (JIRA LEXEVS-XX)

Sign off	Date	Role	CBIIT or Stakeholder Organization	Approver's Comments (If disapproved indicate specific areas for improvement.)
Larry Wright	4/24 /2015	Govt Project Manager	CBIIT EVS	---
Sherri de Coronado	4/30 /2015	Govt Sponsor	CBIIT EVS	—
Kumar Kuntipuram	4/30 /2015	TPM	Leidos Biomed	—

Lucene Code Decoupling - DEFERRED

Reference: LexEVS Code Decoupling

Sign off	Date	Role	CBIIT or Stakeholder Organization	Approver's Comments (If disapproved indicate specific areas for improvement.)
Larry Wright	4/24 /2015	Govt Project Manager	CBIIT EVS	---
Sherri de Coronado	4/30 /2015	Govt Sponsor	CBIIT EVS	—
Kumar Kuntipuran	4/30 /2015	TPM	Leidos Biomed	—

Relevant JIRA Items

Detailed Design - Provide the architecture and design for the new Lucene feature.

LEXEVS-724 - Getting issue details... STATUS

The following JIRA items are all part of LEXEVS-724.

LEXEVS-813 - Getting issue details... STATUS

LEXEVS-814 - Getting issue details... STATUS

LEXEVS-815 - Getting issue details... STATUS

LEXEVS-816 - Getting issue details... STATUS

LEXEVS-817 - Getting issue details... STATUS

LEXEVS-818 - Getting issue details... STATUS

LEXEVS-819 - Getting issue details... STATUS

LEXEVS-820 - Getting issue details... STATUS

LEXEVS-821 - Getting issue details... STATUS

LEXEVS-822 - Getting issue details... STATUS

LEXEVS-823 - Getting issue details... STATUS

LEXEVS-824 - Getting issue details... STATUS

LEXEVS-825 - Getting issue details... STATUS