Software Architecture Document

Natural Language Processing Cell

Version 1.0

1. Table of Contents

2. Revision History

Date	Version	Description	Author(s)
11/29/2007	1.0	Version 1.0	Sergey Goryachev

3. Abstract

This is a software architecture document for Natural Language Processing (NLP) cell. It identifies and explains important architectural elements. This document will serve the needs of stake holders to understand system concepts and give a brief summary of the use of the NLP message format.

4. Overview

The natural language processing (NLP) cell parses unstructured text documents using natural language processing and extracts clinical concepts such as principal diagnoses, discharge asthma medications and smoking status.

The cell returns concepts that can be divided in three different categories.

The first category is UMLS concepts. These concepts result from mapping parts of the document to concepts in the Unified Medical Language System (UMLS) database. Each concept may belong to multiple semantic types such as Finding, Medication, Procedure, etc. An example of UMLS concept is principal diagnosis, which is a UMLS concept of "Finding" semantic type that is discovered in the principal diagnosis-related section of the document. The NLP cell supports principal diagnoses extraction as one of its standard operations.

The second category of concepts is regular expression concept - a concept that results from matching a document text to a set of regular expression based rules. One example of such concept is a discharge medication. NLP cell supports discharge medication extraction as a standard operation. Basically, discharge medication concept is a result of matching a discharge medication related section of text document to a list of regular expressions that define medications.

The last category is smoking status concepts. These concepts result from classification of parts of the document using a classification model trained on the human-annotated set of smoking-related sentences.

4.1 NLP Definitions, Acronyms and Abbreviations

- Principal Diagnosis – a concept representing principal diagnosis extracted from a text document.
- Discharge Medication – a concept representing discharge medication extracted from a text document.
- Smoking Concept – a concept representing a smoking status (e.g., current smoker, past smoker) extracted from a text document.
- Pipeline – a list of NLP components connected together, which is used to extract specific type(s) of NLP concepts from a text document.

4.2 Assumptions / Constraints

The UMLS database shall not contain protected health information.

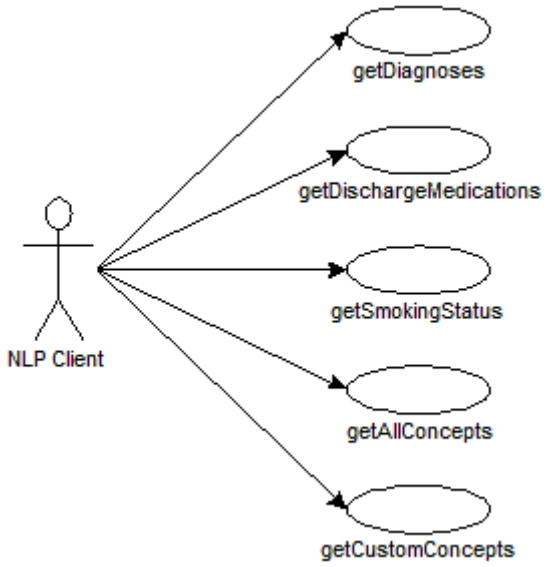
4.3 Technical Platform

The technology used to build the product is as follows:

- Java 2 SE, version 5 or higher.
- MySQL Community Edition Database Server, version 5.0.
- GATE Natural Language Processing Framework.
- UMLS (Unified Medical Language System).
- Apache Tomcat Application Server.
- Apache Axis2 version 1.1.

5. Use Cases

The diagram below depicts common use cases a user may perform with the NLP cell:



5.1 Operations

The NLP service is designed as a collection of operations, or use cases:

1. **getDiagnoses**: returns a list of principal diagnoses codes associated with an unstructured text document sent to the web service. The list is displayed in a tabular format. This operation also returns additional information about each principal diagnosis: the UMLS CUI and UMLS concept preferred name, the list of semantic types to which the concept belongs, the term in the document that mapped to the concept, the list of modifiers (if any) that were assigned to concept (negation, temporal and family history), and the name and categories of document section where the concept was found.
2. **getDischargeMedications**: returns a list of discharge medications extracted from an unstructured text document sent to the web service. The list is displayed in a tabular format. This operation also returns additional information about each discharge medication: the name and categories of document section where the concept was found.
3. **getSmokingStatus**: returns the smoking status (e.g., current smoker) associated with an unstructured text document sent to the web service. There may be more that one sentence in the document that has a smoking status associated with it. The web service returns a list of all smoking status discovered in the report. This operation also returns additional information about each smoking status concept: the name and categories of document section where the concept was found.

4. **getAllConcepts**: returns a list of all available concepts from a document (i.e., principal diagnoses, discharge medications and smoking status) in one service call.
5. **getCustomConcepts**: returns a list of custom concepts. The NLP cell communicates with the NLP core running on the server side. NLP core is configured to run a few standard NLP processing pipelines to support each of the available client operations (1, 2 and 3). Each pipeline consists of NLP components. Each component has its own setup parameters. There are 3 pre-assembled processing pipelines for the 3 standard operations (principal diagnoses extraction, discharge medications extractions and smoking status extraction). However, power users are given an option to (a) change the order of components in a pipeline and (b) change the setup parameters for each component in a pipeline. A custom pipeline allows user to solve new NLP problems such as finding co-morbidities, searching for arbitrary regular expression matches in the document, etc. For example, principal diagnoses extraction is actually a specific case of more general UMLS concepts extraction. Strictly speaking, a principal diagnosis is a UMLS concept found in the document section categorized as “Principal diagnoses related”, that belongs to at least one of semantic types designating medical finding. If we change the section category to “secondary diagnoses related”, we’ll be effectively searching for co-morbidities. Similarly, the discharge medication extraction is a specific case of regular expression matching and can be tweaked to extract other kinds of concepts.

6. Architecture Description

This section provides a description of the architecture as multiple views. Each view conveys the different attributes of the architecture:

1. Components and Connector View
    a. Client Side View
2. Module View
    a. Decomposition View
    b. Uses Style
3. Data View
4. Deployment View

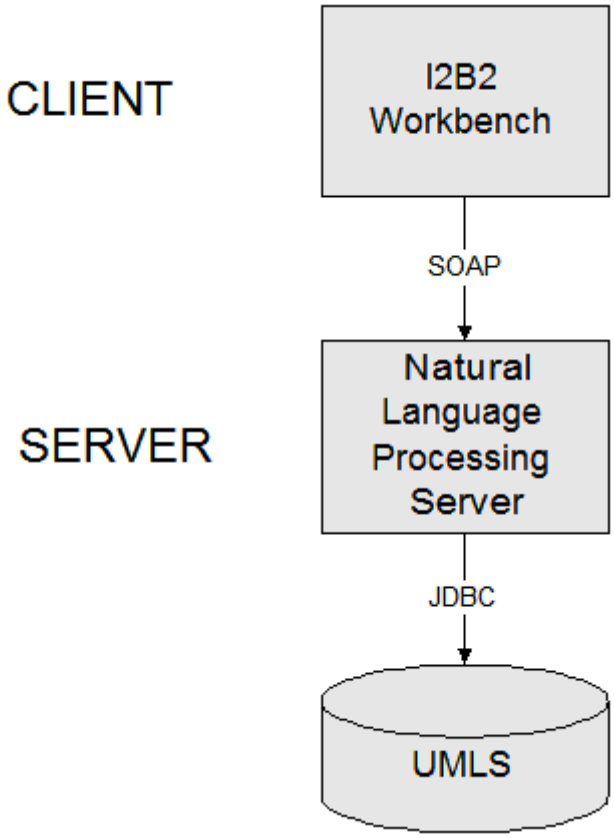
6.1 Components and Connector View

A Components and Connector view represents the runtime instances and the protocols of connection between the instances. The connectors represent the properties such as concurrency, protocols and information flows. Following diagram represents the Components and Connector view for the multi-user installation. As seen below, component instances are shown in more detail with specific connectors drawn in different notations.

6.1.1 Client-Server Style

The NLP system is represented using the Components and Connector Client-Server view.

6.1.1.1 Primary Presentation



6.1.1.2 Element Catalog

Element Name	Type	Description
I2B2 Workbench	Client Component	Web service client submits the requests to NLP server components and renders response XML.
Natural Language Processing Server	Server Component	Provides Web Service Interface for the NLP system. It supports the SOAP protocol.
UMLS	Data Repository Component	This repository is a UMLS database to support UMLS concepts finding (e.g., principal diagnoses)
JDBC	Query Component	SQL query used as a connector between the NLP System and the UMLS database.
Web Service	Request Component	SOAP protocol used to communicate with the external system.

6.1.1.3 Design Rationale, Constraints

N-tier Architecture

The client-server style depicts an n-tier architecture that separates the presentation layer from business logic and data access layer.

6.2 Module View Type

The module view shows how the system is decomposed into implementation units and how the functionality is allocated to these units. The layers show how modules are encapsulated and structured. The layers represent the “allowed-to-use” relation.

The following sections describe the module view using Decomposition and Uses Styles.

6.2.1 Decomposition Style

The “Decomposition” style presents system functionality in terms of manageable work pieces. It identifies modules and breaks them down into sub-modules and so on, until a desired level of granularity is achieved.

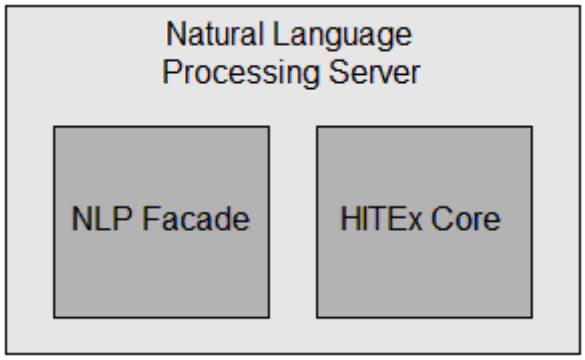
6.2.1.1 Primary Presentation

System	Segment
Natural Language Processing Server	NLP Façade
Natural Language Processing Core	HITEx Core

6.2.1.2 Element Catalog

Element Name	Type	Description
NLP Façade	Subsystem	This subsystem extracts unstructured text document from the request, determines the document type and NLP operation type and selects the appropriate processing pipelines for a given type of NLP operation.
HITEx Core	Subsystem	This subsystem is responsible for assembling and running processing pipelines on the concrete document type, extraction and formatting of processing results.

6.2.1.3 Context Diagram



6.2.2 Uses Style

The "Uses" style show the relationships between modules and sub-modules. This view is very helpful for implementing, integrating and testing the system.

6.2.2.1 Primary Presentation

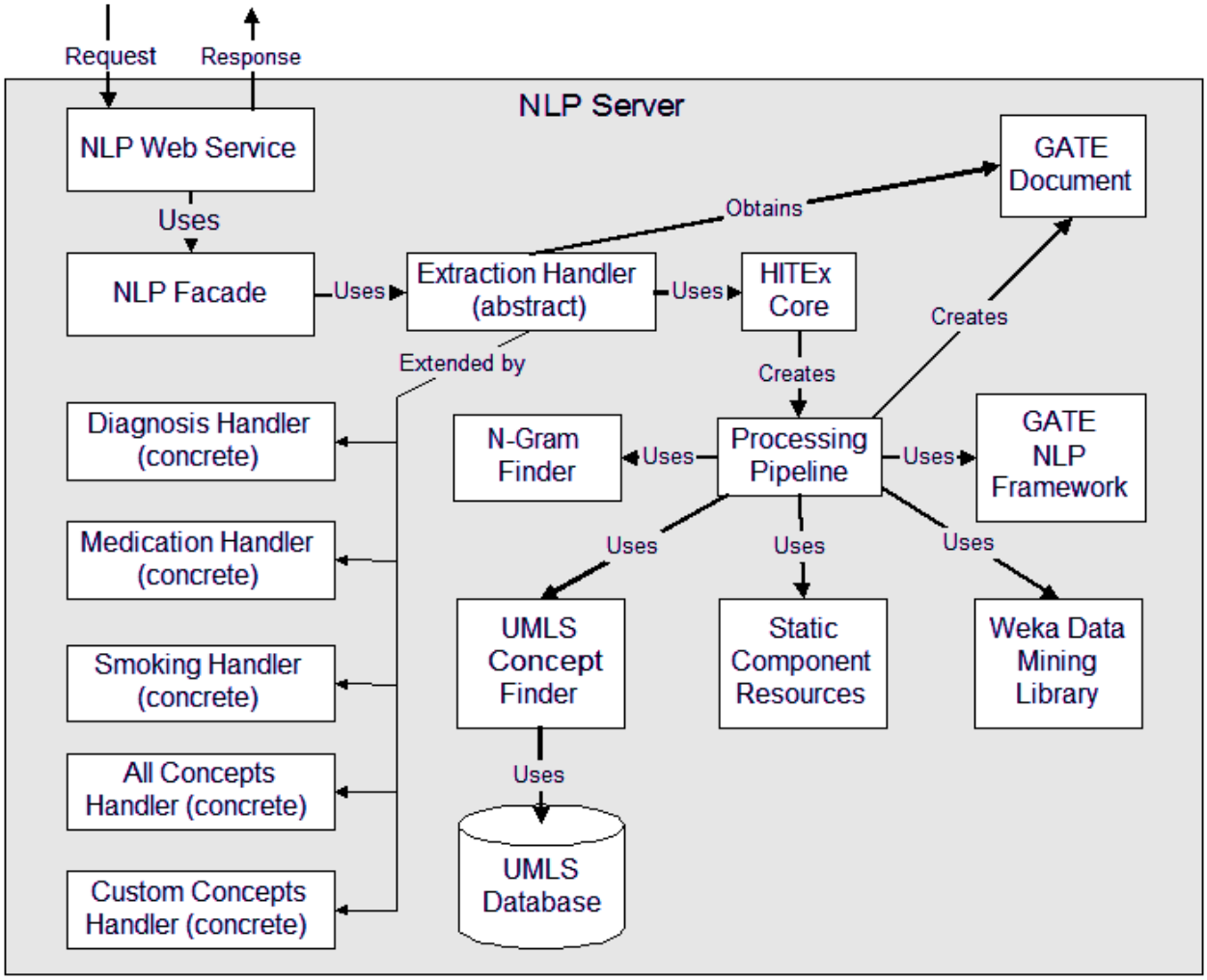
System	Segment
Natural Language Processing Server	NLP Module
NLP Façade Subsystem	NLP Web Service NLP Façade Extraction Handlers
HITEx Core	HITEx Core Module
HITEx Core Subsystem	NLP Pipelines GATE NLP Framework GATE Document UMLS Concept Finder N-Gram Finder UMLS Database Static Component Resources WEKA Data Mining Library

6.2.2.2 Element Catalog

Element Name	Type	Description
NLP Module	Module	Extracts clinical information from unstructured text documents.
NLP Façade	Module	Extracts document text from XML request, determines type of the document and requested operation type, selects the appropriate operation handler and passes the request to it.
NLP Web service	Communication Module	Provides web service interface to NLP operations

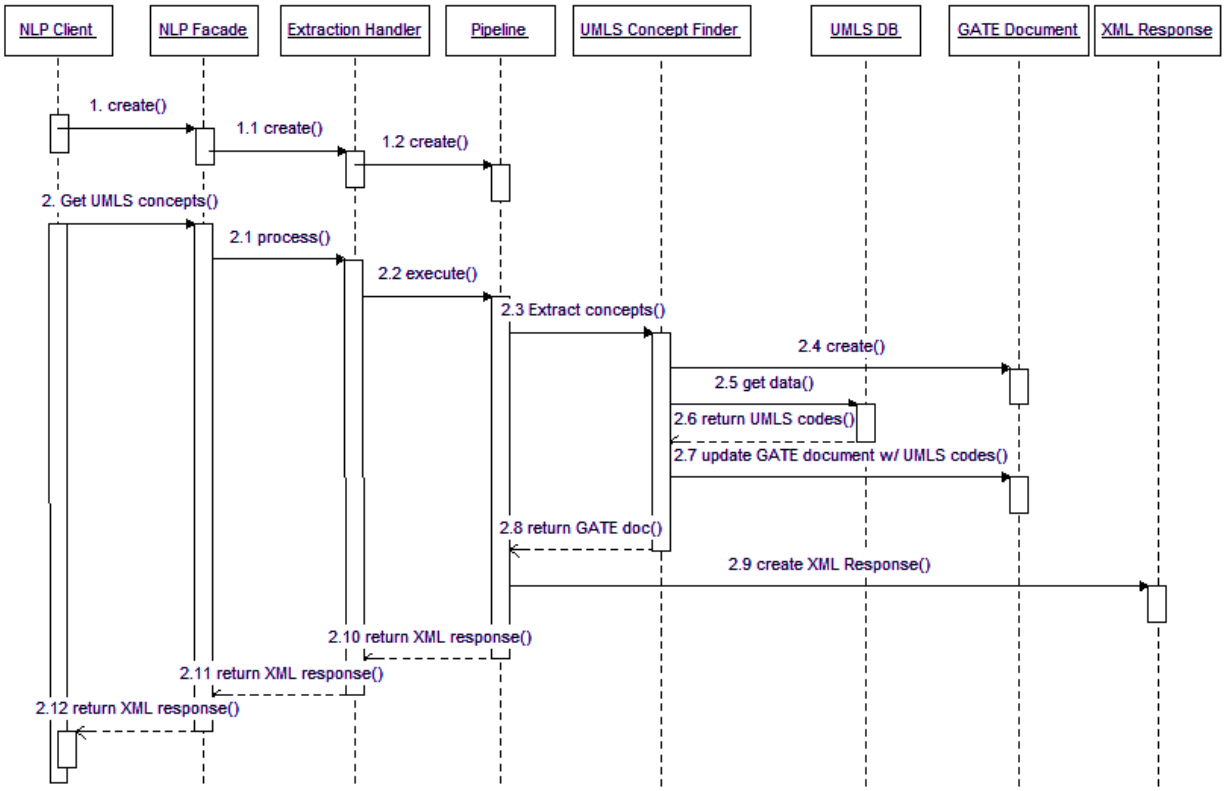
Extraction Handlers	Business Object	Delegates NLP requests to data access object layer (HITEx core) to perform natural language processing operations.
HITEx Core	Module	Supports clinical information extraction from unstructured text documents.
NLP Pipelines	Business Object	Represents list of NLP components in the specific order required to carry out specific NLP operation.
GATE NLP Framework	Supporting Library	The NLP engine behind all NLP operations.
GATE Document	Transfer Object	Represents the input unstructured text document with added NLP metadata.
UMLS Concept Finder	Data Access Object	Supports access to the UMLS database.
N-Gram Finder	Supporting Library	Supports n-gram extraction
UMLS Database	Data Repository	Contains UMLS data required by the UMLS Concept Finder
Static Component Resources	Data Repository	Contain static configuration resources for NLP core system
WEKA Data Mining Library	Supporting Library	Supports smoking classification functionality

6.2.2.3 Context Diagram



6.2.2.4 Sequence Diagram

Below is the UML sequence diagram representing the principal diagnoses extraction operation. Other operation sequence diagrams – discharge medication extraction, smoking status extraction, all concepts extraction and custom concepts extraction – are similar to this diagram.



6.3 Mapping of Styles

The following table is a mapping between the elements in the Component and Connector Client-Server view shown in section 6.1.1, and the Modules Decomposition and Uses views shown in sections 6.2.1 and 6.2.2.

The relationship shown is is-implemented-by, i.e. the elements from the Component and Connector view shown at the top of the table are implemented by any selected elements from the Modules views, denoted by an “X” in the corresponding cell.

	NLP Server	UMLS Database
NLP Module	X	
NLP Façade	X	
NLP Web Service	X	
Extraction Handlers	X	
HITEx Core	X	
NLP Pipelines	X	
GATE NLP Framework	X	
UMLS Concept Finder	X	X
UMLS Database	X	X

N-Gram Finder	X	
Static Component Resources	X	
WEKA Data Mining Library	X	

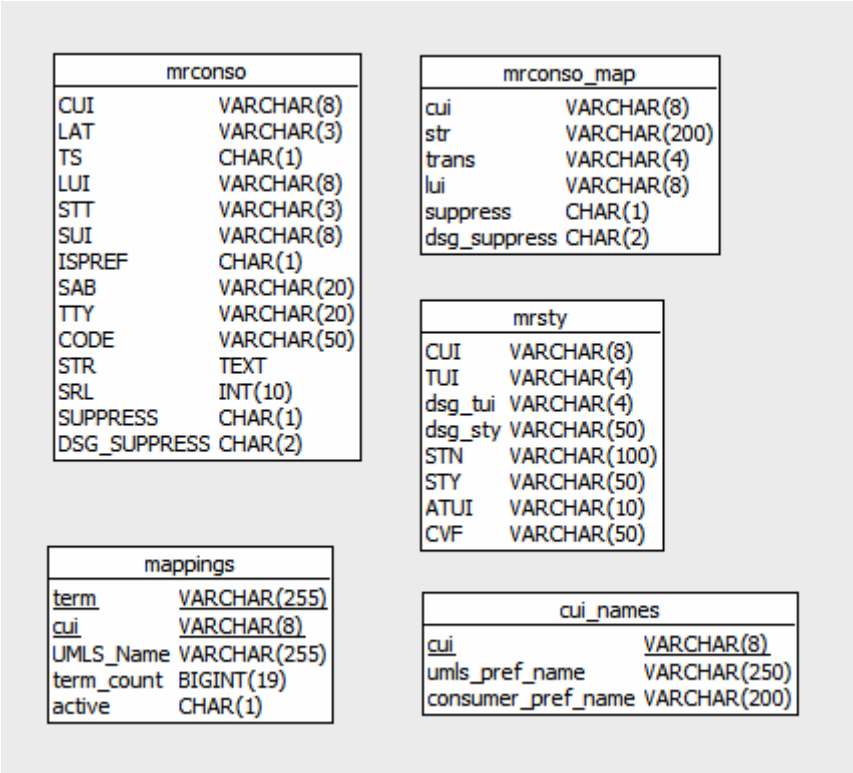
7. Data View

7.1 Data Elements

There are 3 means on communication between a client requesting NLP services and NLP server: UMLS Concept Data Object, Medication Data Object and Smoking Data Object. These objects are derived from GATE documents resulting from any kind on server-side NLP processing. Responses contain a collection of these data objects representing the results of processing an unstructured medical record sent with the client's request.

7.2 Schemas

NLP system uses umls_2004aa database to provide mappings to UMLS concepts. The following tables reside in the umls_2004aa schema:

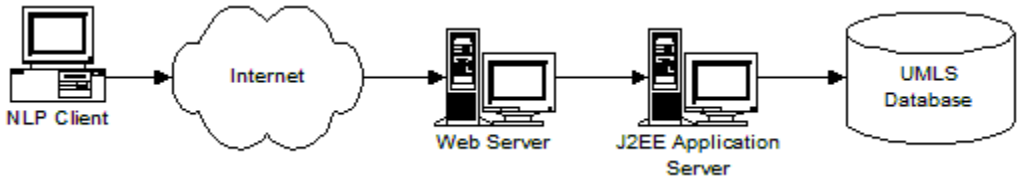


- **mrconso_map** – used to map terms to UMLS concepts by the UMLS concept finder module in the NLP pipelines. This table contains mappings for different term transformations (such as stemmed terms), and different levels of suppression (mild, medium or strong). For example, this table is used for principal diagnoses extraction.
- **mrconso** – contains additional information about UMLS concepts, such as vocabularies the concept is found in, and others.
- **mrsty** – contains data about semantic types of UMLS concepts.

- **cui_names** – contains UMLS cui to UMLS Preferred Name mappings. This table is used to provide extra information about UMLS concept codes.
- **mappings** – contains invalid term to cui mappings that are suppressed by the NLP system.

8. Deployment View

8.1 Global Overview



8.2 Detailed Deployment Model