Software Requirements Specification

for

Natural Language to Cypher Query (NL2CQ)

Version 1.1 approved

Prepared by Kevin Feddema

Huawei

Feburary 22, 2018

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Kevin Feddema	22-02-2018	Iteration/Sprint 1	1.0
Kevin Feddema	07-03-2018	Iteration/Sprint 2 Document Update	1.1

1. Introduction

1.1 Purpose

The purpose of this document is to give a detailed description of the software requirements for the Natural Language to Cypher Query (NL2CQ) system. It will indicate any system constraints, restrictions, and a declaration for the development phase. This document will be used to illustrate the project proposal to a customer or client to gain client approval for the development of the first version of the system as described in this document.

1.2 Document Conventions

Throughout this Software Requirements Specification document, the following terminology will be used:

Term	**Definition**
NL2CQ	Natural Language to Cypher Query
NLP	Natural Language Processing
SRS	Software Requirements Specification
User	An individuals who interacts indirectly with the system through the use of the application that the NL2CQ system is implemented into
Developer	An individual who aids in the development process and construction of a system
TB Tags	Tree Bank Tags: Variable length word type identification tags, eg. Adjective: JJ, Determiner: DT, Adverb: RB
Cypher	A declarative graphical database querying language used in the NL2CQ system
Python	A high-level programming language used in the development of the NL2CQ system
Neo4j	A Database Management System developed by Neo4j that is utilized by the Natural Language to Cypher Query system

DBMS	Database Management System

1.3 Intended Audience and Reading Suggestions

This document is intended for project managers, developers, users, testers, and documentation writers. It is suggested that any reader refer back to 1.2 Document Conventions regularly throughout the analyzation of this document. This SRS document includes 6 different sections. The first section provides a document overview and describes briefly the NL2CQ system. The second section identifies and explains the technical side of the NL2CQ system. The third section will describe any interface and user functionality aspects of the system. Section 4 describes features of the system that are currently implemented. Section 5 glances at system performance as a non-functional requirement.

1.4 Product Scope

The Natural Language to Cypher Query system is a Python based natural language to cypher translator or converter. The system is designed to be implemented into various applications which require a means of converting natural English speech or text into a cypher query. The cypher query, then, being used by the application to effectively query a graphical database. The system receives a statement, question or string and returns a computational cypher query.

The system is designed to convert a statement into multiple cypher queries, which are then tested for translation accuracy by querying the graphical database. If the test causes errors, the query is rejected and the errors are returned. If the test passes, both the test results and the queries are returned.

For the purpose of portraying the functionality of the system, NL2CQ is a web based application, which a user, who wishes to review the cypher translation of natural speech, can interact with.

2. Overall Description

2.1 Product Perspective

The NL2CQ Project started as a course project at The University of British Columbia Okanagan (UBCO) for CoSc 310 – Software Engineering. It is an extension of a Natural Language Processing Project being carried out by The Université Paris-Est Créteil (UPEC) which is being supported by Huawei Technologies Cooperated. A similar research project is being explored by students at The Okanagan Collage, however, the NL2CQ project is considered separate from the Okanagan Collage research. NL2CQ is also considered as an extension or follow-on member of the work being done by UPEC.

2.2 Product Functions

The high-level functions that the NL2CQ system is projected be able to perform are:

- Receive a string of text, statement, or a question from a user as input
- Analyze the input text (assign TB tags, scrub for stop words, etc)

- Produce a set of cypher queries
- Test queries for validity against a Neo4j database

2.3 User Classes and Characteristics

The system is designed to be used by either users who are experienced with cypher querying or inexperienced users, however NL2CQ is not projected to provide additional functionality to either user class. The primary goal is to provide satisfactory functionality to a single, general user, class.

2.4 Operating Environment

NL2CQ is designed as a web based application that can be run on any platform. The system uses PHP 5.6 and Python 2.7 to operate, both of which are implemented on the host server. It is also being primarily designed for use on Google Chrome as multi-browser implementation is not a strong priority in the development phase.

2.5 Design and Implementation Constraints

System constraints include:

- Potential runtime latency between the HTML/PHP and Python modules
- *Execution time of python script
- System is developed using Chrome as the primary execution browser
- English is the only supported translation language

*Performance wise, python is reported to be slower than languages such as Java, C#, and C. NL2CQ also requires the importation of added libraries (Natural Language Toolkit) which can also reduce runtime performance.

2.6 User Documentation

A Software User's Manual is provided for user's to refer to for easier understanding of the NL2CQ software.

2.7 Assumptions and Dependencies

The system requires the use of an external Python library called nltk (Natural Language Toolkit) to function and translate natural language to the graphical database query language, Cypher. The DBMS that will be used in the initial version of NL2CQ is Neo4j, along with a test database that will be used to develop and test the software. The implementation of an external database with NL2CQ operates on the assumption that the implemented database contains a list of attributes that can be defined in order to produce an effective query.

3. External Interface Requirements

3.1 User Interfaces

The interface of the NL2CQ system is designed for extreme simplicity and ease of access. The home page contains the project logo and a text input bar in the center of the page which is what a user interacts with to return a query. The user inputs there statement that they wish to convert to cypher and then simply press the submit button. The submit action then redirects to the results page where the statement is processed and the queries returned.

The web application also contains links to The NL2CQ Developers About Us page, the Contact page, as well as the Documentation page, which links the project documentation to the application to be viewed by the user.

3.2 Hardware Interfaces

NL2CQ is platform independent, it can, therefore, be run on any device, however, the web application has only been designed for the Google Chrome browser. A user may experience page layout errors, result errors, or execution errors if the application is run in an unsupported browser. The only restriction on the viewing device is the screen size must be larger than 250x200p for the site layout to perform optimally.

3.3 Software Interfaces

The NL2CQ main user interface functions by linking to executable python scripts through PHP. The Python implementation is constructed with the use of the Python Natural Language Toolkit (nltk), which is utilized to analyze the user input string and return the queries. Query error and validity checking is accomplished by testing the returned queries against the Neo4j database.

3.4 Communications Interfaces

The system is implemented on an Apache based Web Server with Python script handling using the Secure Hyper Text Transfer Protocol.

4. System Features

4.1 Converting Natural Speech to Important Words with associated TB Tags

4.1.1 Description and Priority

The system receives a statement or question from the user, and separates the words and assigns Tree Bank tags to each word. This function will later be altered to output translated cypher queries but the current version portrays the first step in reaching that stretch goal. This feature has been given a high priority because it is one of the only primary function of the NL2CQ system.

4.1.2 Stimulus/Response Sequences

This conversion feature is instantiated by the user upon submission of an input string. The user first loads the NL2CQ web application, then inputs a question, and presses submit. This then redirects the user to the results page where the input is processed and the results are displayed.

4.1.3 Functional Requirements

REQ-1: Web application must have an uninterrupted internet connection
REQ-2: Web application should be view in Google Chrome to ensure correct functionality

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The NL2CQ System must return a result within a time frame that corresponds with the amount of time it takes for the submission to transmit the input to the internal script, establish a connection with the database (implemented in later version), validate the results (implemented in later version), and return the results. For simplicity, no longer than 4 seconds.