

Universidad de Guadalajara

Centro Universitario de los Valles

**Software Configuration Management**

System for Event Detection from Text Data

**Master: Software Engineering**

**Third semester**

Professor: Omar Ali Zatarain Duran

Efren Plascencia JR.

222977393

Wednesday, November 29, 2023

Table of Contents

[Introduction 4](#_Toc152171079)

[Baseline of the project 5](#_Toc152171080)

[General description 5](#_Toc152171081)

[Key features 5](#_Toc152171082)

[Input module 5](#_Toc152171083)

[Preprocessing module 5](#_Toc152171084)

[Processing module 5](#_Toc152171085)

[Result module 5](#_Toc152171086)

[Administration module 5](#_Toc152171087)

[Requirements 6](#_Toc152171088)

[Functional Requirements 6](#_Toc152171089)

[Input module 6](#_Toc152171090)

[Preprocessing module 6](#_Toc152171091)

[Processing module 6](#_Toc152171092)

[Result module 6](#_Toc152171093)

[Administration module 7](#_Toc152171094)

[No-Functional Requirements 7](#_Toc152171095)

[Availability 7](#_Toc152171096)

[Devices: 7](#_Toc152171097)

[Design 7](#_Toc152171098)

[Maintainability 7](#_Toc152171099)

[Performance 7](#_Toc152171100)

[Security and Privacy 7](#_Toc152171101)

[Usability 7](#_Toc152171102)

[Requirements Analysis 8](#_Toc152171103)

[Design System 10](#_Toc152171104)

[Committee board 11](#_Toc152171105)

[Description of general functions within the project 11](#_Toc152171106)

[Change management policy. 11](#_Toc152171107)

[Change request. 12](#_Toc152171108)

[First change request 12](#_Toc152171109)

[Second change request 12](#_Toc152171110)

[Third change request 12](#_Toc152171111)

[Fourth change requests 12](#_Toc152171112)

[Status Accounting 13](#_Toc152171113)

# Introduction

Software Configuration Management (SCM) is an essential process in software development that involves a series of tasks and responsibilities intended to control and manage changes in a software project. This document focuses on the changes and processes involved in SCM, providing a detailed overview of each of them applied to a project.

The document will address the following key processes:

**Configuration identification (CI)**: this process involves the identification and definition of the elements that constitute the software at a given point in time. This may include source code, design documents, tests, data, among others.

**Configuration Control (CC)**: This process deals with the evaluation, coordination, approval, or disapproval of changes to the software. It is also responsible for the implementation of approved changes.

**Configuration Accounting**: This process involves recording and reporting the information necessary for the effective management of the software. This includes the status of configuration items and requested changes.

**Configuration Audit**: This process verifies that changes have been implemented as approved and that all configuration elements are correctly identified and described.

**Release management and delivery**: This process deals with the preparation and delivery of software to customers, including release planning, release design, and release delivery.

Throughout the document, we describe how each of the process involved in SCM are applied in the project call *System for event detection from text data,* highlighting their importance in effectively managing changes in a software project. The goal is to provide a clear understanding of how changes to the software can be effectively managed.

We hope that this document will be useful guide and valuable resources for those involved in software projects.

# Baseline of the project

In the context of Software Configuration Management (SCM), a project baseline is an agreed description of the attributes of a product at a specific point in time, which servers as a basis for defining change.

In this section we show the version 1 of the baseline, this means first attempt the describes the baseline for the system call: System for Event Detection from Text Data.

## General description

System with the ability to identify events present in texts written in English, and to determine the possible relationships that exist between the identified events and make a graphic representation of the relationships found.

Where entries to the system will be through Wikipedia links, files in txt format and by writing text in the system.

## Key features

The following section shows the key features that are expected to be developed in the system.

### Input module

* Users can enter Wikipedia link and the system will fetch the data from the web site.
* Users can upload text files (.txt) where the content will be shown to the user and be able to edit the text in the system.
* User can enter text directly to the system (write and edit).

### Preprocessing module

* Creation of a set of tokens which we refer to as a sequence.
* Identification and extraction of stop words in the sequences.
* Identification and extraction of concepts in the sequence.
* Identification and extraction of operators in the sequence.
* Identification of relative and specific dates present in the sequence. (Anchors and intervals)

### Processing module

* Identification of relationships between concepts (implicit)
* Identification of relationships between concepts (explicit)

### Result module

* Tabular event reporting
* Generation of reports of events graphically (timeline / graph)

### Administration module

* Management of users and documents processed for storage.

# Requirements

In this section we show the requirements for the system based on the key featured described in the description of the system, we classified by functional and non-functional. Where we will use the acronyms FR to refer to functional requirements and NFR for non-functional requirements.

## Functional Requirements

### Input module

FR 1: The system will be able to receive text files.

FR 2: The system will be able to show data from the text files to the user.

FR 3: The system will be able to tolerate changes in the data from the data.

FR 4: The system will be able to receive Wikipedia links by the user.

FR 5: The system should be able to fetch the Wikipedia links.

FR 6: The system should be able to get the Wikipedia without HTML and CSS tags (web scrapping)

FR 7: The system will be able to receive data written in the system by the user.

### Preprocessing module

FR 1: The system will be able to create a sequence of tokens.

FR 2: The system will be able to identify stop words in the sequence.

FR 3: The system will be able to identify concepts in the sequence.

FR 4: The system will be able to identify operators in the sequence.

FR 5: The system will be able to identify relative and specific dates in the sequence.

### Processing module

RF 1: The system will be able to identify relationship between concepts explicit with anchors or intervals.

RF 2: The system will be able to identify relationships between concepts implicit with anchors or intervals.

### Result module

FR 1: The system will be able to generate tables representing the concepts and the relationships between concepts.

FR 2: The system will be able to generate tables representing the isolated concepts.

FR 3: The system will be able to generate reports of concepts with a graph.

FR 4: The system will be able to generate reports of concepts with timelines.

### Administration module

The following requirements are specified for the web version.

FR1: User must be able to create account by providing their personal information such as full name, email, username, and password.

FR2: User should be able to edit their personal information and contact information.

FR3: User should be able to log in using their registered credentials.

FR4: Users should be able to recover forgotten password through a reset process.

FR5: The system must validate the uniqueness of email to avoid duplicate registration.

FR6: The system must verify the authenticity of the email during the registration process.

FR7: The system should be able to save files and the result of the analysis performed if the user wishes.

## No-Functional Requirements

### Availability

NFR 1: The system must be available most of the time for the needs of the final user.

### Devices:

NFR 1: The program must work consistently across different operating systems (Windows or Linux).

NFR 2: The program must work consistently across different internet browsers.

### Design

NFR 1: Responsive user interface for use on desktop and web applications.

NFR 2: The system must allow informative messages about errors.

### Maintainability

NFR 1: The code source should be well structured in some pattern of software development to facilitate future update and changes.

NFR 2: Documentation on the code and architecture to facilitate maintenance by future developers.

### Performance

NFR 1: Loding time should be adequate to avoid user frustration.

NFR 2: Processing time for texts needs to be the more efficient possible.

### Security and Privacy

NFR 1: The system must follow regulations and privacy standards.

NFR 2: The system access must have some security mechanisms.

### Usability

NFR 1: User interface must be intuitive and easy to navigate for the final user.

# Requirements Analysis

In this section, we analysis the requirements and we present a use case diagram for the general representation of main actors that intervene in the system with the intention demonstrating the purpose for the system and the understanding for the development team and the stakeholders.

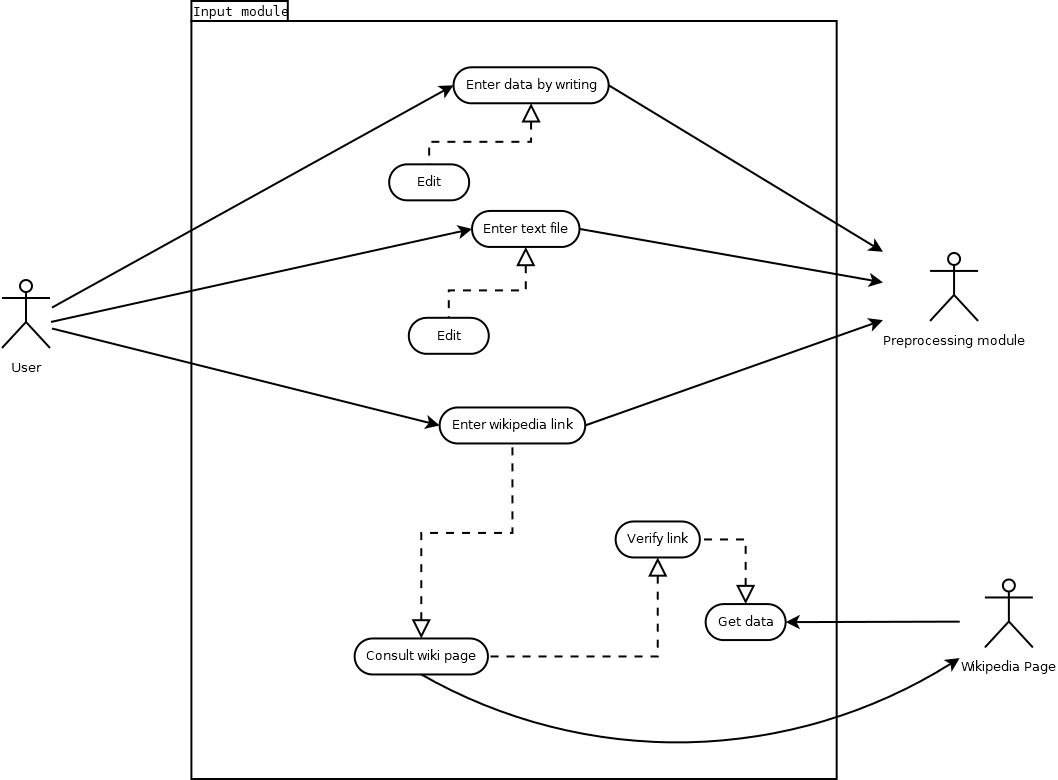


Figure 1 Use case diagram input module.

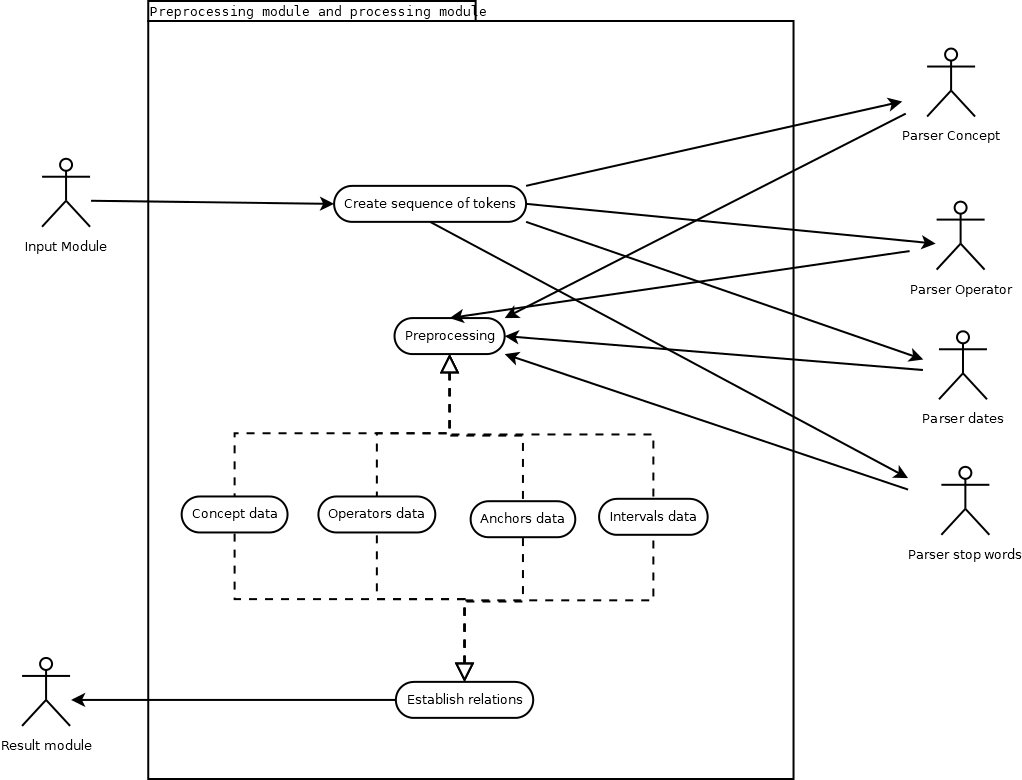


Figure 2 Use case diagram preprocessing and processing module.

# Design System

In this section, we describe the flow of the system figure 3 where it seeks to show the different stages involved in the system.

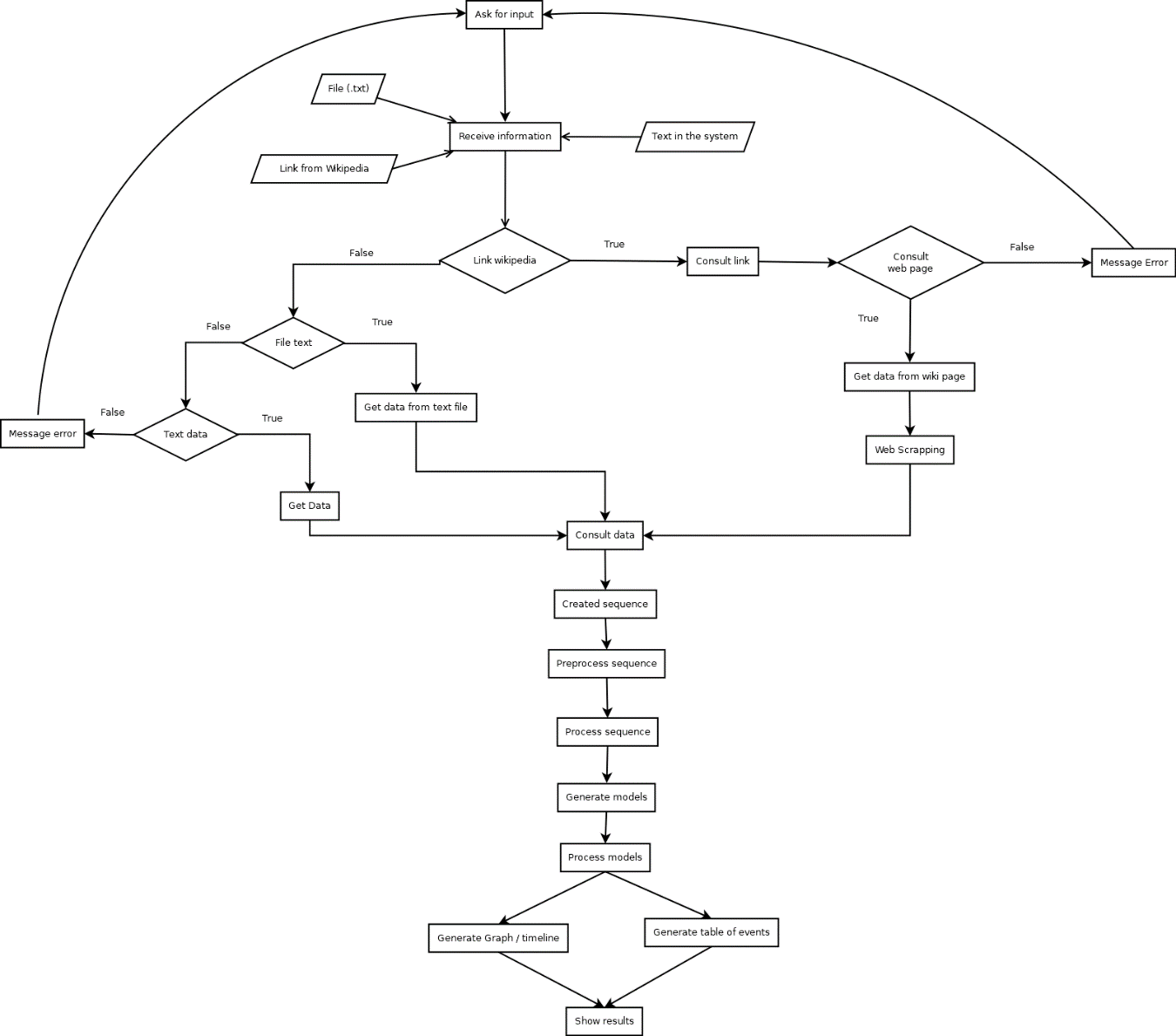


Figure 3 Flowchart of the system.

# Committee board

In this section we describe the software development committee board that is an essential for the SCM framework. This board comprises a group of key stakeholders, including project managers, lead developers, and quality assurance leads, among others, who are responsible for overseeing and guiding the software development process.

The board’s primary role is to ensure that the software development process aligns with the organization’s objectives and standards. They are involved in critical decision-making process, such as approving changes to the software, managing risks, and resolving conflicts that may arise during the development process.

## Description of general functions within the project

# Change management policy.

In this section we describe the change management policy that is a critical component of SCM. It outlines the procedures for making changes and managing the effects of these changes in a software project. The policy is designed to handle changes in a controlled manner, ensuring that no unnecessary updates are made, while necessary changes are not overlooked. It involves identifying, documenting, approving, or rejecting changes to the project, with the aim of controlling costs, reducing unnecessary risks, and providing a basis for project evaluation and accountability.

The change management policy includes guidelines on how to submit change requests, how these requests are approved and implemented, and how changes are tracked and reported. It also defines the roles and responsibilities of each member of the software development team, ensuring that everyone understands their part in the change process.

# Change request.

In this section we talk about the change request that is a formal proposal for an alteration to be made in a software project and is a fundamental aspect of SCM. It is typically raised by any stakeholder involved in the project, such as a team member, a project manager, or even a client, when they identify a need for a change in the product.

The change request includes detailed information about the proposed change, including the reason, the benefits it will bring, any potential risks, and the resources required to implement it.

In this section we show the desired modification to the software systema and can range from a simple request for a minor adjustment to a significant architecture alteration. Once a change request is submitted, it undergoes a thorough review process by the change control board or the Software Development Committee Board.

This board evaluates the impact of the proposed change on the project’s scope, cost, schedule, and quality. If approved, the change is then implemented, tested, and verified before being incorporated into the software product.

## First change request

The client requires that a new functionality needs to be developed. The system should also detect geographic relations within the text and generate a spatial-temporal graph that describes the following directions: north, south, east, west, above, below, left, right inside, and outside.

## Second change request

In addition to the initial CR1, the client wants a module that differentiates semantics according to the context: “I look forward to hearing from you” and “We moved forward towards the tower.” One speaks about the future and the other describes a direction.

## Third change request

The client requires a module that perceives the phonetics from a person and translates it into text.

## Fourth change requests

The client desires a new Screen for interacting with the system. It requires web design for people with disabilities.

# Status Accounting

In this section we talk about status accounting of SCM. It involves the process of recording and reporting the necessary information on the status of the development process. This includes tracking all elements of the software configuration, documenting their state during the different stages of the software development lifecycle, and maintaining a history of the changes that have been made. It provides visibility into the progress of the project, allowing stakeholders to make informed decisions based on the current state of the software.

Status Accounting helps in understanding the composition of different versions and builds of the software. It also aids in tracing the changes made to each component over time. This information Is vital for managing the software effectively, diagnosing issues, and planning future actions.

The criterial to evaluate in the aspect of documentation is shown in the following table.

|  |  |
| --- | --- |
| Documented Process Criteria | |
| Check Mark | Task description |
|  | Record the current approved configuration documentation and configuration identifiers associated with each configuration item (CI) |
|  | Record and report implementation status of authorized changes. |
|  | Provide the traceability of all changes from the original released configuration documentation of each configuration item (CI) affected. |
|  | Record all document and software that has been delivered to the stakeholder or the main repository. |

The criterial to evaluate in the aspect of code is shown in the following table.

|  |  |
| --- | --- |
| Code Process Criteria | |
| Check Mark | Task description |
|  | Each commit has a unique identifier, a timestamp, the person who made the change, and a message describing the change. |
|  | All function develop has a backup. |

The criterial to evaluate in the aspect of time is shown in the following table.

|  |  |
| --- | --- |
| Criteria table for time | |
| Evaluated in days or month (depends on the project or the change request) | Aspect to consider: |
|  | Amount of time for the approve a change. |
|  | Amount of time for the conception of a feature to its implementation. |
|  | Amount of time that developers spend working on the project. |
|  | Amount of time from the start to the completion of the task. |

# Reference

[1] E. H. Bersoff, "Elements of Software Configuration Management," in IEEE Transactions on Software Engineering, vol. SE-10, no. 1, pp. 79-87, Jan. 1984, doi: 10.1109/TSE.1984.5010202.

[2] Plascencia JR., E. GitHub. Retrieved November 20, 2023, from <https://github.com/Efr3nJR/SCM_Class2023B>