

D-BUG

CONFIGURATION  
MANAGEMENT  
PLAN FOR  
TSL-KINECT

DUYGU ARALIOĞLU

BEDİA ACAR

ZÜLFÜ ULAŞ ŞAHİN

GÜLNUR NEVAL ERDEM

MIDDLE EAST TECHNICAL UNIVERSITY  
COMPUTER ENGINEERING

2012

**Contents**

- 1. Introduction ..... 3
  - 1.1. Purpose of Configuration Management Plan ..... 3
  - 1.2. Scope of the Document..... 3
  - 1.2. Definitions, Acronyms and Abbreviations..... 4
  - 1.4. Document References..... 4
  - 1.5. Document Overview ..... 5
- 2. The Organizations CM Framework ..... 6
  - 2.1. Organization ..... 6
  - 2.2. Responsibilities ..... 7
  - 2.3. Tools and Infrastructure..... 7
- 3. Configuration Management Process ..... 9
  - 3.1. Identification ..... 9
    - 3.1.1. Source Code ..... 9
    - 3.1.2. Data ..... 10
    - 3.1.3. Documentation ..... 10
  - 3.2. Configuration Management and Control ..... 11
    - 3.2.1. Software Change Request ..... 11
    - 3.2.3. System Change Implementation..... 12
  - 3.3. Configuration Status Accounting ..... 12
  - 3.4. Auditing ..... 12
- 4. Project Schedules and CM Milestones ..... 13
  - 4.1. Project Schedules..... 13
  - 4.2. CM Milestones..... 13
- 5. Project Resources ..... 14
- 6. Plan Optimization ..... 14

## **1. Introduction**

This document is the configuration management plan of TSL-Kinect project, developed by D-BUG project team. TSL-Kinect is a project which aims to solve the communication problem between speech-impaired people and others by using Microsoft Kinect tool.

### **1.1. Purpose of Configuration Management Plan**

While developing a software project, it gets harder to manage the parts and elements in the system as their numbers increase. Hence configuration management is required to manage these parts and elements which, in terms of a project, can be thought as classes, modules, and implementations. While implementing the project, classes, modules and even the interface may change in order to fulfill the needs and solve the problems faced during the implementation phase of the project. These changes may occur at any time and also may occur simultaneously. In order to be able to manage the project, to deal with these changes in a professional way, to ensure that there is a safe communication between developers, and to improve the quality of the application and its reports, a configuration management plan is essential in the project improvement process.

### **1.2. Scope of the Document**

The scope of this document is the identification of configuration management plan for the project. The plan presents configuration management activities that will be applied throughout the development process of our project and the activities explained in this document are applicable during the development and maintenance of the project.

The document contains methodologies and activities designed to manage changes and relate them. This document consists of information about organization of the team,

responsibilities of the team members regarding this project, timeline, and CM activities that are to be applied and methodologies and activities designed to manage changes.

Moreover versioning, differencing work product and reporting on all are included in this document. The related audience for this document is D-BUG software team, our assistant and instructors.

## 1.2. Definitions, Acronyms and Abbreviations

Definitions, acronyms and abbreviations are listed in the below table.

<b>CI</b>	Configuration Item
<b>CM</b>	Configuration Management
<b>CMP</b>	Configuration Management Plan
<b>SCR</b>	System Change Request
<b>CMT</b>	Configuration Management Team
<b>CCT</b>	Configuration Control Team
<b>SDT</b>	Software Development Team
<b>RCT</b>	Release Control Team
<b>TT</b>	Testing Team

## 1.4. Document References

[1]IEEE Standard for Software Configuration Management Plans (IEEE Std 828-2005)

[2] Configuration Management Plan Presentation, METU Computer Engineering, Spring 2012

[3] Detail Design Report of TSL-Kinect, this document prepared according to IEEE

Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998)

[4] <http://subversion.tigris.org/>

[5] <http://www.microsoft.com/en-us/kinectforwindows/>

## 1.5. Document Overview

This CMP document consists of six sections which are Introduction, Organization, Configuration Management Process, Project Schedule and CM Milestones, Project Resources and Plan Optimization.

The first section, Introduction, contains the purpose of the CMP, the scope of the document and the acronyms used throughout the document. Used document references are also provided in Introduction section.

In the second section, Organization, the organization and responsibilities of team members are explained. The tools that are used during the development of the project and their functionalities are also included in this section.

In the third section, Configuration Management Process, the identification of the configuration items, the methodologies that are to be followed, the plan for the audits and sub-parts are explained.

In the fourth section, Project Schedule and CM Milestones, important dates and milestones are presented.

In the fifth section, Project Resources, the project resources used for the software development process and also for coordination of the communication between team members are defined.

The last section, Optimization, contains the methods which are going to be used to optimize CMP.

## 2. The Organizations CM Framework

### 2.1. Organization

All the members of D-BUG are completely aware of the responsibilities of being a group. Therefore, all of the team members willingly contribute to CM process to develop a successful project. Each team member has equal rights on this project management process. Each member assigned to sub-teams and task assignment of each sub-team is done fairly.

Software Development Team (SDT): All group members.

Testing Team (TT): Duygu Aralioğlu, Ulaş Şahin

Release Control Team (RCT): Bedia Acar, Gülnur Neval Erdem

Configuration Management Team (CMT): Duygu Aralioğlu, Bedia Acar

Configuration Control Team (CCT): Gülnur Neval Erdem, Ulaş Şahin

Here are the explanations of the tasks assigned to each sub-team:

**Software Development Team:** SDT is responsible for implementing the modules of the TSL-Kinect project. In addition, whenever the Testing Team requests changes SDT will be in charge to make the changes. This team is also responsible for fixing bugs and creating releases.

**Testing Team:** TT is responsible for finding bugs and checking whether the modules meet the requirements. If the system requirements are not met, TT requests changes. After testing, TT gives feedback about the modules.

**Release Control Team:** RCT controls the current and next version of the project, and gives feedback to SDT about the current release.

**Configuration Management Team:** CMT is responsible for maintenance of the CM organization. Therefore, this group will keep the CMP up-to-date.

**Configuration Control Team:** The main responsibilities of CCT are to review SCRs, accept or reject SCRs and monitor SCRs. This team also supervises all the activities of other groups.

## 2.2. Responsibilities

Although each member of D-BUG is assigned to different sub-teams and has different responsibilities, there are some common responsibilities that should be taken by all the members. Here are the common responsibilities:

- Each member should strictly follow the CM schedule.
- Each member should comment on the changes he/she made on the source code before committing resources through SVN.
- If there is a SCR, other members should be informed via e-mail.

## 2.3. Tools and Infrastructure

- **Microsoft Visual Studio 2010 Express:** Microsoft Visual Studio is a multi-language integrated development environment (IDE). It is a nice environment that provides a reliable debugger, easy to use form designer and code editor and provides a good support to combine different parts of our project.
- **SVN:** Subversion is used to maintain current and historical versions of files such as source code, web pages, and documentation. Its goal is to be a mostly-

compatible successor to the widely used CVS. [4] We will be using SVN repository in order to maintain current and old versions of the implementation.

- **TRAC:** Trac is an open source, web-based project management and bug-tracking tool. It also serves as a web interface to revision control system that we use, namely SVN.
- **Google Documents:** Google Docs is a web-based office suite and data storage service. It allows us to create and edit documents concurrently together with the other members of D-BUG.
- **Google Groups:** We preferred to use Google Groups in order to have a common e-mail address in order to share literature search documentation and communication.
- **Web site:** All of the documents, project description and the project progress can be seen on our web page.
- **Kinect SDK:** It is a non-commercial software development kit for Windows 7 platform and provides capabilities to developers to build applications with C++, C#, or Visual Basic by using Microsoft Visual Studio 2010.[5]

## **3. Configuration Management Process**

### **3.1. Identification**

The Configuration Items (CI) can be separated into 3 categories namely source code, data and documentation will be explained in detail in the following sections:

#### **3.1.1. Source Code**

In a software project, source code is one of the fundamental CI. In this project, the source code files of TSL-Kinect project are stored and managed with the repository of SVN hosted by our department and the TRAC system. These tools enable group members to access and manage source code interactively. The system is developed with C# programming language.

Our source code can be divided into two parts. First part is the gesture recognition which is the main primary duty that our sponsor company demands from us. Therefore, the implementation of gesture recognition and to succeed in this part is the main objective of D-BUG. We have achieved the gesture recognition part with a limited number of gestures. However, whenever a new gesture is added to the system, we need to revise all the distinguishing properties of other gestures, such as time between states or hand angles, in order to maintain distinctive properties of each gesture. Moreover, we apply several tests after adding a new gesture and if test fails we need to modify our source code or at least add additional comparison properties to data representation structure.

The second part of our source code is the user interface part. In TSL Kinect project user will be able to control the interface via gestures so in the development of this part our main goal is to provide user a smooth, user friendly and enjoyable NUI. In order to achieve this facilities user interface implementation has to be evolved gradually. At first basic needs such as clicking a button is implemented. Other aesthetics functionalities, slide effects, controlling interface items size or alignment properties etc., will be added in time. Therefore, the design and source code of user interface is changing frequently according to the new facilities and sections added to the project.

### **3.1.2. Data**

In TSL-Kinect project, the data consists of the joint positions coming from the SDK and the gesture representations which are constant and obtained experimentally. Joint positions are temporary data, used only for calculation of angles and comparisons, when the operation is completed they are gone. However, the name, estimated hand positions, estimated hand angles, estimated state numbers and estimated time requirements of the TSL gestures that are aimed to be recognized are stored in the system.

### **3.1.3. Documentation**

Documentation is a significant CI and all documents are prepared with the contribution of all team members in the Google Docs environment. The written documents so far are;

- Project Proposal
- Software Requirements Specifications
- Initial Design Report
- Detailed Design Report
- Revised Design Report
- Configuration Management Plan
- Weekly Reports

## **3.2. Configuration Management and Control**

### **3.2.1. Software Change Request**

During the configuration process of the project any SCR is will be handled through SVN and Trac system. SVN will help the project team to see the minor changes and whereas Trac will support the environment for major changes and versioning. Minor System Change Requests will consist of:

- Related team member name
- Description of the SCR
- Date of the SCR
- Possible deadline of the SCR
- Related Module(s) of SCR
- Priority of the SCR
- Version of the SCR

When any team member reported a System Change Request, in Trac system s/he opens a ticket and it can be seen by every member.

### **3.2.2. Software Change Evaluation**

The Trac system will provide the chance of discussing and evaluating the SCR's by the tickets. Moreover, all SCRs will be discussed face to face in team meetings and during the daily communications of the team members. In each evaluation or comment, all members can emphasize their opinions freely and evaluation will be based on these opinions.

### **3.2.3. System Change Implementation**

When an SCR occurred after an evaluation, it can possibly affect other related CIs. In such condition, those CIs will be determined and changes are applied to them. Then, they can be updated through SVN.

### **3.3. Configuration Status Accounting**

Configuration items and a system for managing software changes were given in the previous section. When lots of changes occur in the project, it gets harder to control every configuration item, so, in order to deal with such situation information about related configuration items are needed to be stored. In the project, in order to keep track of development process various ways will be used to express the changes in the CI. By doing this, the communication both between team members and other people who are following the project will be improved.

The stored information about CIs includes configuration item identifications, information on the change request and information related to the details of the implementation. As the project moves toward completion, comments of the SVN commits and well-prepared weekly progress reports will be our guide to track the common changes.

Finally, the project web-page will be frequently updated in order to give information about the development process of the project and the faced problems and their solutions.

### **3.4. Auditing**

All of the team members are responsible for auditing. First of all, the implementation of any change, requesting from one of the team members, is coded after the request is approved. When the implementation is finished, the member must debug the code and correct it if

necessary. Then the latest and working version of the source code is committed by the team member to SVN repository. After this uploading stage, members of TT check the code with the trivial cases and control whether the code works properly with other modules in the system.

Team members are in touch by way of regular weekly meetings, Trac system and Google Group. We try to keep track of our project schedule by checking our works regularly.

## 4. Project Schedules and CM Milestones

### 4.1. Project Schedules

Our project schedule was determined in the Gantt Chart provided with TSL-Kinect DDR. It consists of milestones of the project and tasks which must be followed. The dates are adjusted in compliance with course schedule and especially considering to accomplish the better design of the system. CM Milestones of the project are stated in the next section.

### 4.2. CM Milestones

- **First Development Snapshot, Demo** (March 29, 2012)  
The progress on the first semester's prototype is going to be exhibited.
- **First Release Demo** (May 15, 2012)  
In this milestone, the system will be completely finished and the first version of TSL-Kinect will be released.
- **Final Package** (May 22, 2012)  
In final package milestone, TSL-Kinect product will be delivered after enhancing the first release in terms of better visual and performance considerations.

## 5. Project Resources

During the development process the following tools are used for software development and for coordination of the communication between team members:

- Visual Studio 2010 : Software Development Environment
- SVN : Revision Control System
- TRAC : Issue Tracking System
- Google Documents : Report Sharing and Editing Environment
- Google Groups: Project Resources and Literature Research Document sharing between members.
- Web site : Project Development News

## 6. Plan Optimization

This CMP is prepared to be a guide for both configuration and progress of the project TSL-Kinect during the implementation. Therefore, it is important to obey CMP for plan optimization. When an update or change is determined to be done, all group members will be notified via Trac.

Our intention is to strictly follow the predetermined schedule, however, as the project is ongoing and by considering the probability of facing unpredicted problems, it may be necessary to make some changes in the deadlines of the living schedule during the development process. Therefore, there will be weekly meetings among group members. According to these meetings, living schedule is going to be formed regularly and deadlines will be re-arranged if necessary. Any updates in the living schedule will be shared in the project web page.