ARCHITECTURAL SPECIFICATION

CSIR – SKETCHUP BLOCKS

CONTENTS

CHANGE LOG

Version 1.0	25/06/2013	Document created.

1. INTRODUCTION

1.1 PURPOSE

The purpose of this document is to give a high-level abstraction of the architectural strategies/tactics that will be used in the Sketchup Blocks project. This document aims to provide clear motivation for the chosen architectural design as well as an explanation of how it will serve to implement the specifications required by the Vision and Scope document for the Sketchup Blocks project.

1.2 PROJECT SCOPE

Sketchup Blocks aims to provide the client with a system that will convert physical 3D models into digital form. The system will include software and hardware components.

Hardware components:

- The physical blocks with fiducial markers attached (called 'smart blocks').
- The construction floor on which the smart block constructions are built.
- Five web cameras.

The scope of the software components will broaden from release to release. The client only requires the first release to consider the project successful. The releases following the first release will address additional concerns and implement extra features mentioned by the client.

Release 1:

The initial system will be able to recognize a small set of geometrically simple objects without any time constraints. After the blocks have been transformed into a digital model, the vertices of the model will be output to an object file for import into Google Sketchup.

Release 2:

This release aims to limit the time taken to recognize a fiducial and recreate the model it represents. The amount of constructed and recognizable blocks will also be increased. The setup of the system will be simplified to improve mobility and allow for fast setup. The removal of blocks will be detected without extra input from the user.

Release 3

This release aims to add a viewing window that will display the model as it being created thereby providing visualization before exporting to a model. The detection of the fiducials should be possible in suboptimal lighting conditions.

Optional release

Custom blocks may be imported to the smart block database. Users can view the construction of other users over the network.

1.3 RELATED DOCUMENTS

Project Vision and Scope (Sketchup Blocks project, by CICO).

2. SYSTEM DESCRIPTION

The project subscribes to the concept of the Internet of Things. The Internet of Things is centred around the idea of 'smart objects' which are physical objects and devices that are given a virtual representation and linked to information networks. These smart objects will ideally be able to capture, react to and communicate about data captured from their physical environment autonomously.

This project aims to create a system that can convert physical 3D models into digital models that can be viewed and edited in Google Sketchup. The physical model will be built with blocks that have fiducial markers attached. The locations of these blocks will be picked up on web cameras by Reactivision software and the resulting data will be used to create digital 3D models that can be viewed and exported to Google Sketchup.

3. ARCHITECTURAL TACTICS AND STRATEGIES

3.1 ARCHITECTURAL SCOPE

Session management

Users may wish to continue working on a project or start a new project. Session management of this kind may grow increasingly complex for collaborative projects.

Flat Database

Information about the smart blocks (size, id, etc.) will be stored in a flat database that is maintained by the system software. Information about other users (for collaborative purposes) will also be stored in a system-maintained flat database.

Network Management

For collaboration to take place, the system must provide network capability. This will allow users on different computers to view and export the same model.

3.2 QUALITY REQUIREMENTS

Speed/Throughput

The throughput requirements differ from release to release. The initial release imposes no time constraints on the generation of the object file. The second release requires that the object file be updated/generated within half an hour of modifying the physical model structure. The final release requires the implementation of a 3D viewer that will reflect modifications to the physical model in real time. The final release also requires that modifications be reflected in the object file within 2 minutes.

Portability

The initial release makes no demands regarding portability. The second release requires that the physical construction area should be mobile. In particular, this means that the web cameras, construction floor and blocks be relocatable and that there must be some way of recalibrating the system in its new environment. The third release does not make any particular mentions of the portability requirements, although the portability may be refined for the final release.

Resilience/Flexibility

The flexibility of the system is mainly dependent on its ability to recognize fiducial markers in different lighting conditions and after being moved. Movement of the system will require recalibration of the cameras since there may be minor changes in their alignment and relative positions. This ties in closely with the portability requirement.

The initial release makes no flexibility demands. The portability requirement for the second release implies that some level of flexibility will be required so that the system can function in different locations with normal, in-house lighting conditions. This will be implemented by a calibration phase that initiates upon startup and checks camera alignment and position.

The final release requires that the system must function in suboptimal lighting conditions i.e. with very little bright or natural light.

Integrability

The output files produced by the system must be supported by Google Sketchup. This is a core requirement of the system. The output format chosen is Collada. This is addressed in greater detail in **Section 3.3**.

Low production cost

The materials used in system construction must be affordable. Although this is not a core requirement, it is satisfied by virtue of the technologies and equipment chosen by the client. Web cameras are readily available at low cost and the technologies chosen are open source.

3.3 INTEGRATION AND ACCESS CHANNEL REQUIREMENTS

The system must be able to receive and interpret data via the TUIO protocol which will be received from reacTIVision software installed on the host computer. The system must also integrate with Google Sketchup by exporting the digital model in a supported file format, like Collada.

The system must be able to connect to other instances of the Sketchup Blocks system over the internet in order for people to successfully collaborate with others.

Local access requirements include that the user must ultimately be able to interface with the system by means of command blocks and smart blocks to be able to build models and collaborate with other people over the internet.

3.4 ARCHITECTURAL CONSTRAINTS

The system has to integrate with reacTIVision by being able to process the data sent to the system via the TUIO protocol. The system has to be implemented in “Processing”, a high-level java like programming language. The system must also be able to integrate with Google Sketchup by exporting a supported file like Collada.

4. ARCHITECTURAL PATTERNS

The Model View Controller (MVC) design pattern separates the model and the way in which it is changed from the view.

We chose this model because it supports multiple views that display the same model and compliments the natural modular layout of the project. The modular layout of the MVC pattern allows for fast development and promotes ease of development in teams. In addition to this, the development team is comfortable with this pattern and have experience in its implementation.

The model (in terms of the architecture) contains three distinct types of information:

- Information about the blocks and their fiducials.
- Information about other clients/collaborators.
- The current 3D model being constructed by the user.

The view consists of two parts:

- A real-time view of the current 3D model.
- An exporter that will export the model for use by Sketch Blocks.

The controller consists of:

- A model constructor that interprets real-world input to construct a model.
- A session manager that controls interaction between the different parts of the system.

Interaction between the user and the system is done through the use of command blocks. The controller will map the command blocks onto service requests.

The collaboration considered in the optional releases will be done with the client-server architecture with one model acting as the server and the model of the other collaborators acting as clients. The server will provide all the clients with the current model and allow clients to change their part of the model.

Abstract Class Diagram [Sketchup Blocks]
Visual Paradigm for UML Community Edition [not for commercial use]
Interpreter
SessionManager
BlockDatabase
FlatDatabase
UserDatabase
ModelViewer
Exporter
ModelConstructor
Lobby
Model
Model
View
Controller
<<use>>

Figure 4.1

5. ARCHITECTURAL STRATEGIES

Caching:

In the optional phases where user collaboration is considered the other collaborators models will be cached for display.

First-In/First-out processing:

All the processing done is equally important and the order does matter, therefore the information will be processed as is received.

6. ACCESS AND INTEGRATION CHANNELS

ReacTIVision is a cross-platform computer vision framework for fast and robust tracking of fiducial markers. ReacTIVision can be installed on the system machine or on a remote machine with network access to the system machine. The system will be getting the data of the smart blocks from reacTIVision via the TUIO protocol.

Google Sketchup is 3D modelling software used to create and manipulate 3D models. Our system will be exporting the digital models to a file format supported by Google Sketchup such that further manipulation of the models can be done within Google Sketchup.

The user must be able to interface with the system via a local access channel. The user will be able to interface with the system via command blocks and smart blocks. Command blocks have special meaning to the system and will enable the user to achieve certain tasks like creating a new model, continuing an existing model, exporting a model or joining an online lobby to view another user's progress. Smart blocks are used as normal building blocks in the system: the smart blocks will be used to build models on the construction floor.

The system must also be able to access other instances of the Sketchup Block system over the network in order to enable the user to collaborate with others by viewing their projects and helping build large projects over the internet.

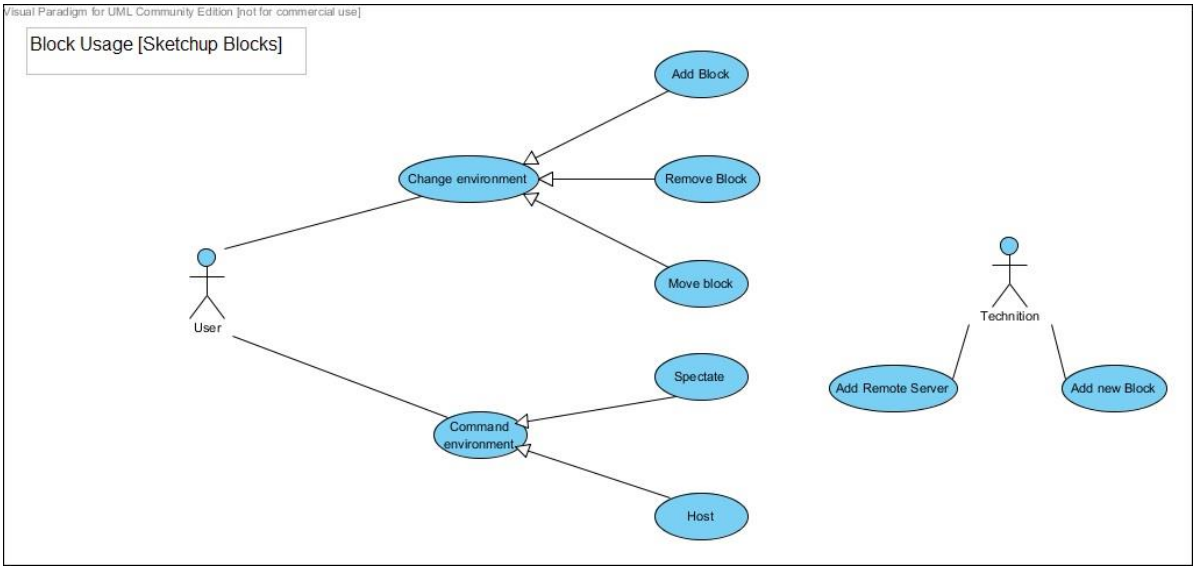

Figure 6.1

7. TECHNOLOGIES

Technologies used by the system include:

- ReacTIVision - Used to track fiducial markers.
- Processing - High-level java like programming language implemented by the system.
- Collada - File format supported by Google Sketchup.
- Microsoft Windows - Operating system the system will run on.

8. GLOSSARY

Collada	Interchange file format for interactive 3D applications that is supported by Google Sketchup.
Command block	A smart block used to control the system (as opposed to model construction).
Construction floor	The flat surface on which the physical model is built and around which the web cameras are erected.
Fiducial markers	Small markers that will be attached to the physical blocks and used to identify them.
Google Sketchup	3D modelling program.
IoT	Internet of Things
MVC	Model-View-Controller, an architectural pattern
Object file	The Collada file that contains all the data of a particular model
ReacTIVision	Open source, cross-platform computer vision framework for tracking fiducial markers that are attached to physical objects.
Smart block	A physical block with an attached fiducial marker.
TUIO	A protocol that allows the encoding and transmission of a tangible interface component abstraction, such as tokens, pointers and geometries. This is the protocol by which ReacTIVision will communicate with the Sketchup Blocks system.