Operational Concept Description (OCD)

Mission Science iRobot

Team #07

Jiashuo Li	Project Manager, Life Cycle Planner, Developer
Chen Li	Requirements Engineer, Software Architect, Developer
Farica Mascarenhas	Operational Concept Engineer, IV&V, Quality Analyst
Hanadi Mardah	Tester
Sergey Mukhin	Prototyper, Developer
Yun Shao	Feasibility Analyst, Developer

Version History

Date	Author	Version	Changes made	Rationale
08/20/12	SK	1.0	• Original for CSCI577; Tailored from ICSM OCD Template	• To fit CS477 course content
09/20/14	FM	1.1	• Updated the knowledge of the system's workflows and the identified Operational Concepts	• To document the progress till Valuation Commitment Review
10/12/2014	AR	2.0	• Updated information in Program Model, Benefits Chain, Diagram, System Boundary and Environment Diagram, and information on current system • Updated information on current system, System Objectives, Constraints and Priorities, Organizational and Operational Implications	• To document the improved understanding and to meet FCR expectations
10/16/2014	AR	2.1	• Fixed comments from ARB meeting	• Fixed comments from ARB meeting

11/22/2014	AR	3.0	• Reviewed priorities of Organizational goals and Levels of Service. • System boundary diagram changed to match the MMFs. • Program model and benefit chain made to link one-to-one to each other.	• Comments from FCP grading incorporated. • Readiness for Development Commitment Review.
12/6/2014	AR	3.1	• Updated some terminologies in business workflow	• As per comments received in DCR ARB.
2/10/2015	FM	3.2	• No changes	• No changes
4/5/2015	FM	3.3	• Updated sections 1.2 and 3.4.1	• To document changes for Transition Readiness Review

Table of Contents

Table of Tables

Table of Figures

1. Introduction

1.1 Purpose of the OCD

This document provides, in detail, the following:

1. Identification of the Success-Critical Stakeholders
2. Shared visions and goals of the stakeholders
3. The artifacts used and created by the system in the operating environment
4. The current business flow

The Success-Critical Stakeholders of this project are:

1. Mission Science community as the project owner and maintainer (represented by Prof. Darin Gray)
2. Undergraduate students as users (represented by Ian and Edwin)
3. Elementary school students as users
4. Elementary school teachers as users
5. Developers as guides to maintainers and undergraduate students

1.2 Status of the OCD

Difference from the previous major version: This OCD document is currently in Transition Readiness Review stage.

2. Shared Vision

Table 1: The Program Model

<table>
<tr><td colspan="4">Assumption:
1. Elementary school teachers are comfortable with the iRobot and are willing to let us do the Mission Science program at their school
2. Kids of elementary school are interested in these “hands-on” science and technology activities, projects, experiments, and demonstrations though which they will aspire for careers in Science, Technology, Engineering and Mathematics (STEM)
3. The capability of the iRobot will be greatly improved using GUI with drag and drop operation</td></tr>
<tr><th>Who?</th><th>What?</th><th>Why?</th><th>For Whom?</th></tr>
<tr><td>1. Developers

2. Undergraduate students

3. Prof. Gray</td><td>1. Develop a drag-and-drop GUI

2. Train undergraduates how to program iRobot using the GUI

3. Present elementary students and teachers how to use the GUI

4. Demonstrate successfully implemented cases</td><td>1. Simplify the control/programming of iRobot

2. Increasing elementary school students’ interests and understanding of complex function of iRobot and aspiration in STEM

3. Providing better chance of funding for Mission Science</td><td>1. Elementary School Students

2. Elementary School Teachers

3. Mission Science</td></tr>
<tr><td colspan="2">Cost:
Development Costs
iRobot Hardware
Platform computer
Elementary school teachers and students’ training
Visual Studio License etc.</td><td colspan="2">Benefits:
1. Increased number of schools adopting iRobots and GUI to make kids learn logic and programming.

2. Increased number of enrollments in STEM projects.

3. Increased number of sales of iRobot due to the easy-to-use interface rather than C programming.</td></tr>
</table>

2.1 Benefits Chain

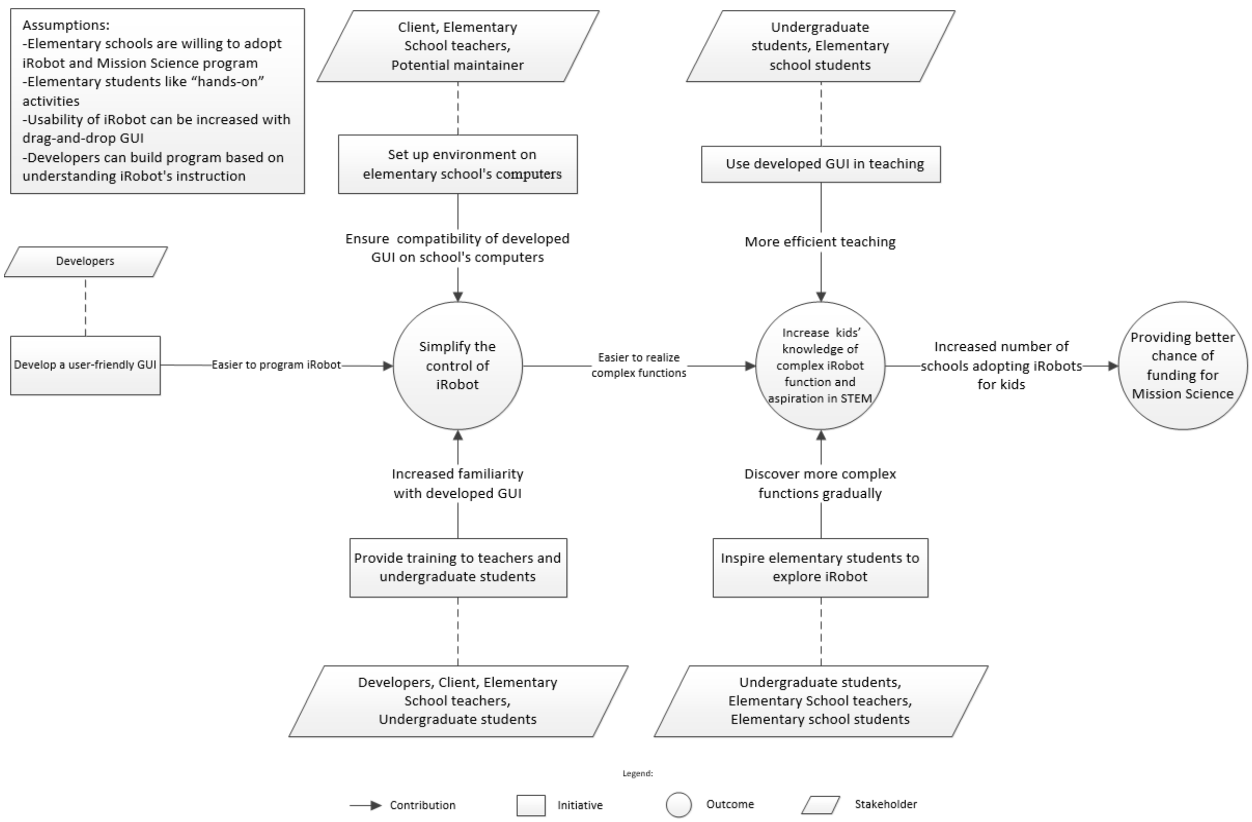


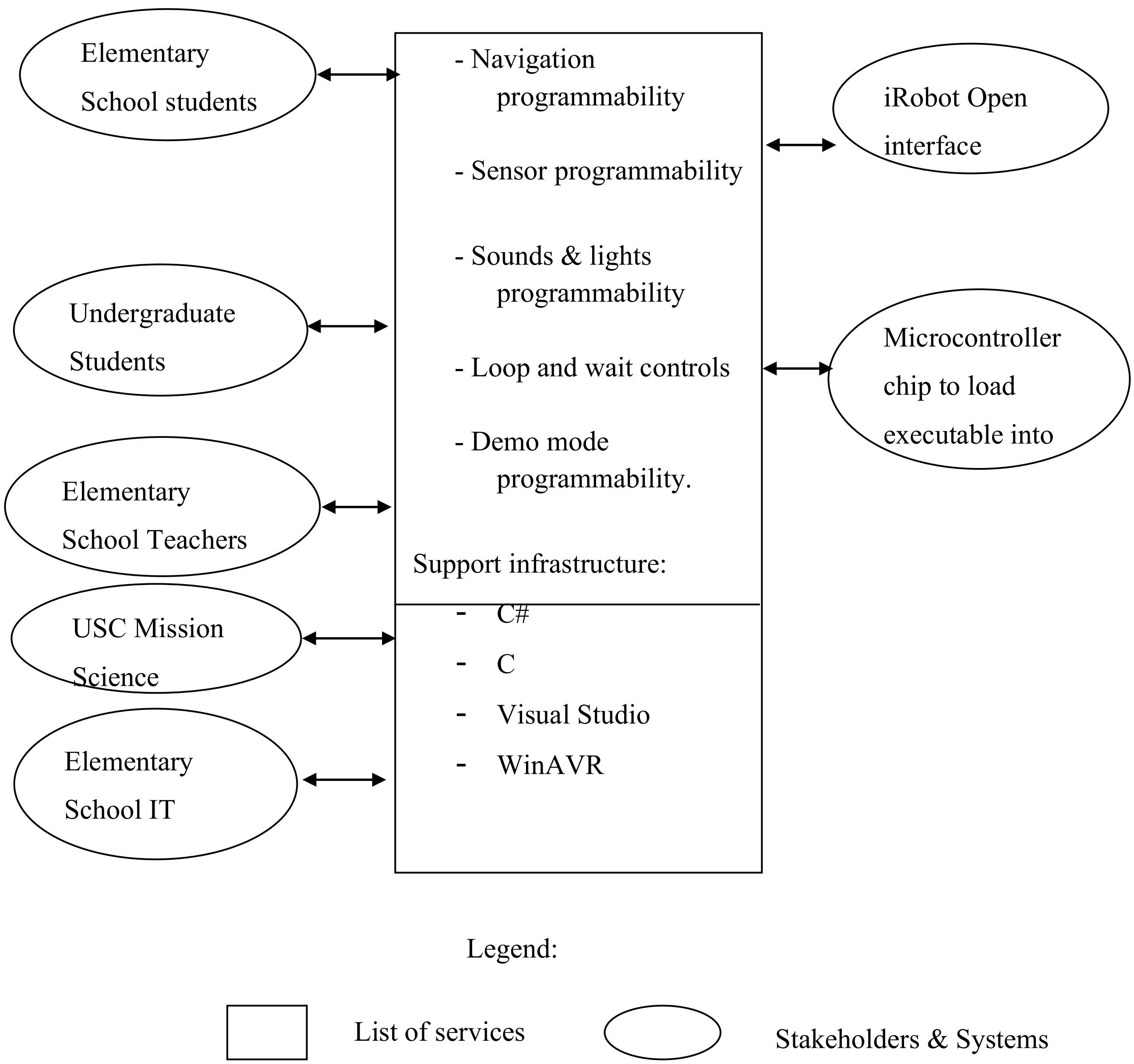
Figure 1: Benefits Chain Diagram of Mission Science iRobot

2.2 System Capability Description

The system to be built is a drag & drop, fun-to-use programming interface for the elementary school kids to learn the logic and control systems. The instruction set programmed by kids would undergo a conversion to C-code and compilation for which a backend needs to be designed. The long term goal of this project is to instill inspiration towards the computer science field and - generate higher funds for Mission Science. This system would be more intuitive and easy-to-use than its predecessor – Scribblers. iRobots with the upcoming GUI have more to offer than scribbler with their infrared sensors, bump sensors and the demo modes which are pre-programmed into the system.

2.3 System Boundary and Environment

Figure 2: System Boundary and Environment Diagram of Mission Science iRobot



3. System Transformation

3.1 Information on Current System

3.1.1 Infrastructure

What is iRobot?

iRobots are programmable machines which can perform operations like forward movement, backward movement, waiting, turning, bumping to obstacles, sensing the obstacles and heat and taking necessary actions, etc.

How does the iRobot currently work?

iRobot uses electronic and software interface called iRobot Roomba Serial Command Interface(SCI) that allows us to control or modify iRobot's behavior and remotely monitor the sensors. It is a serial protocol that allows users to control actuators, to request sensor data through the external serial port. A microcontroller (8-bit AVR micro controller), that shows high performance, having advanced RISC Architecture, Non-volatile Program and Data Memories, etc is used to support SCI.

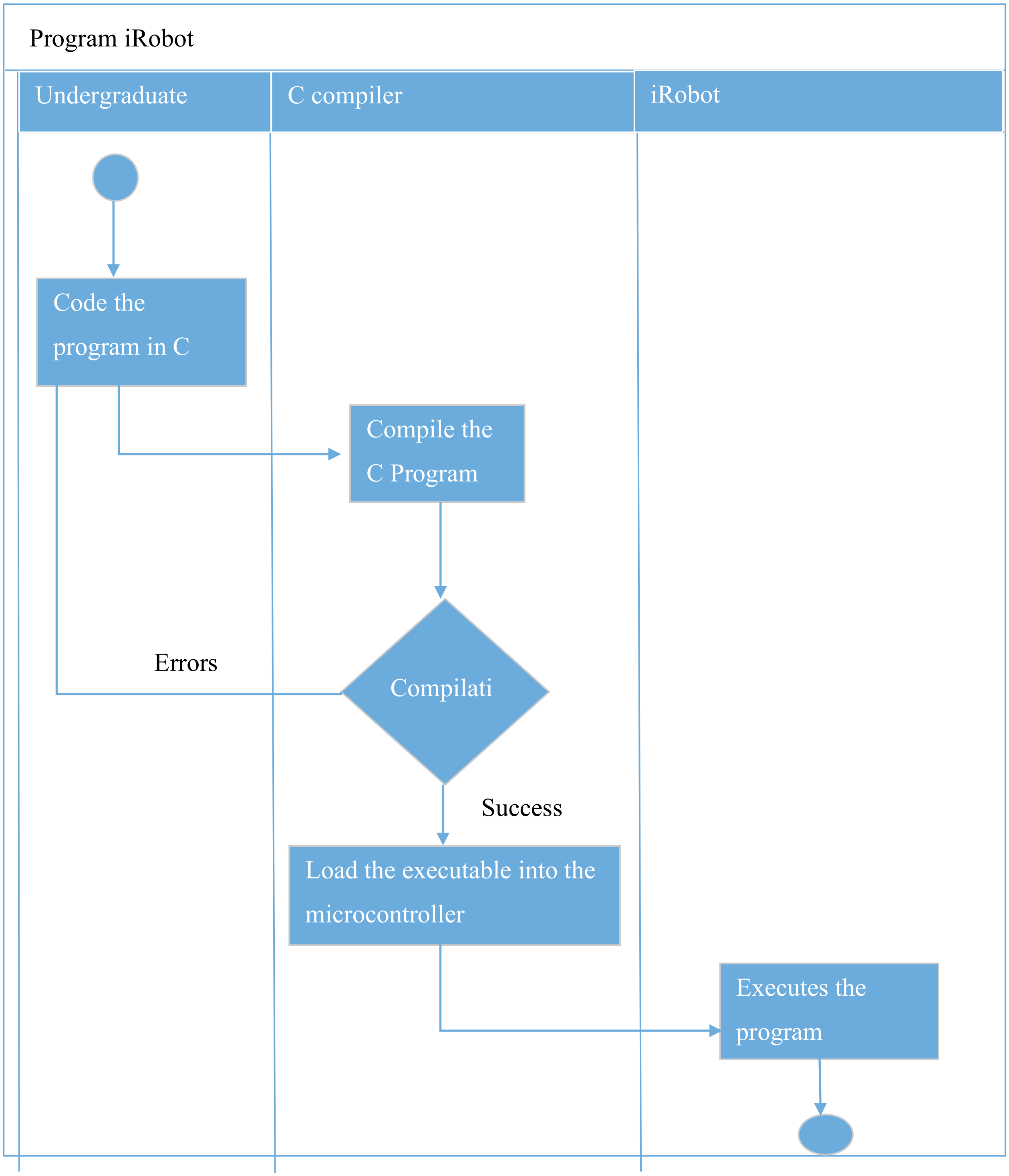
How is the iRobot currently used?

Professors use iRobot with undergraduates in their classes. Undergraduate science students program the iRobot according to the given problem statements using C language and load the executable into the iRobot using an 8 bit low power microcontroller/Mind control stick.

3.1.2 Artifacts

Our interface does not replace any artifacts. There are no artifacts that are produced or consumed by this new system. It only provides an alternate way of operating iRobot by eliminating the need to learn C programming.

3.1.3 Current Business Workflow



3.2 System Objectives, Constraints and Priorities

3.2.1 Capability Goals

Table 2: Level of Service Goals

Capability Goals	Priority Level
OC-1 Navigation programmability: Fun-to-use interface to help elementary school children learn programming to control navigation of iRobot.	<< Must have>>
OC-2 Sensor programmability: Fun-to-use interface to help elementary school children learn programming to control navigation of iRobot.	<< Must have>>
OC-3 Loop and wait constructs: The interface shall allow drag & drop of programming constructs like if-then-else, for and while loops.	<<Should have>>
OC-4 Sounds and light programmability: The interface shall allow drag & drop of musical notes and LED on/off instructions.	<<should have>>
OC-5 Demo mode programmability: The interface shall allow drag & drop of pre-programmed demo modes.	<<Should have>>

3.2.2 Level of Service Goals

Table 3: Level of Service Goals

Level of Service Goals	Priority Level	Referred WinWin Agreements
LOS-1: Seamless interoperability between GUI and compiler.	Must have	WC_3299 The system shall generate instructions for iRobot in C which is then later compiled for deployment on the microcontroller using the APIs of iRobot.

LOS-2: Detect and report ambiguous instructions in an understandable way.	Must have	WC_3297 The system shall detect and show logic errors (conflicting/inconsistent instructions) in a easy-to-read way.
LOS-3: Reasonable frequency of reading sensor data.	Must have	WC_3302 The system shall enforce a tolerance limit of +/- 2 to 3% on sensor programmability.
LOS-4: Portability above Windows 7	Should have	WC_3298 The system shall be a native windows 7 and above application.

3.2.3 Organizational Goals

OG-1: Generate more excitement toward STEM fields.

OG-2: Widen the user sector (not confined just to C programmers).

OG-3: Improved understanding in students about logic and control systems.

OG-4: Decrease time needed to program iRobot to execute complex instruction set

OG-5: Use of robots will improve funding opportunities for Mission Science.

3.2.4 Constraints

CO-1: Windows as an Operating System: The new system must be able to run on Windows 7/8/8.1

CO-2: Zero Monetary Budget: The selected NDI/NCS should be free or no monetary cost.

CO-3: **C# as a Development Language:** Visual studio- .NET will be used as a development language for the Drag & Drop GUI interface.

3.2.5 Relation to Current System

Table 4: Relation to Current System

Capabilities	Current System	New System
Roles and Responsibilities	Elementary school kids have no role to play in the current system.	Elementary school kids will be the target Success-Critical-Stakeholders in the new system.
	Undergraduate students have the role of users in the current system.	Undergraduates have an added responsibility of guiding elementary school kids in the new system
	Elementary school IT administration have no role to play in the current system.	Elementary school IT administration would have a new responsibility of maintaining the new system and using the user and troubleshooting guides.
	Elementary school teachers have no role to play in the current system.	Elementary school teachers have a new responsibility of teaching the elementary school kids to program the iRobots using the drag & drop approach in the new system.
	Mission Science current responsibility is to maintain the iRobot machine and the mind control stick.	Mission Science would have an added responsibility of maintaining the source code of GUI and back-end.
User Interactions	By coding instructions in C language.	By drag and drop GUI.
Infrastructure	Windows 7 PC, iRobot, C compiler, Mind control stick, USB, user guide.	Windows 7 PC, iRobot, C Compiles, Mind control stick, USB, Drag & Drop GUI interface, User guide, troubleshooting guide.
Stakeholder Essentials and Amenities	A sophisticated extendible system to program iRobots.	Develop a more easy and fun-to-use programming interface to expand the user sector for iRobots.
Future Capabilities	Extendible to provide GUI support.	Program complex demo modes more quickly and efficiently.

		Improve handling of ambiguous instructions.

3.3 Proposed New Operational Concept

3.3.1 Element Relationship Diagram

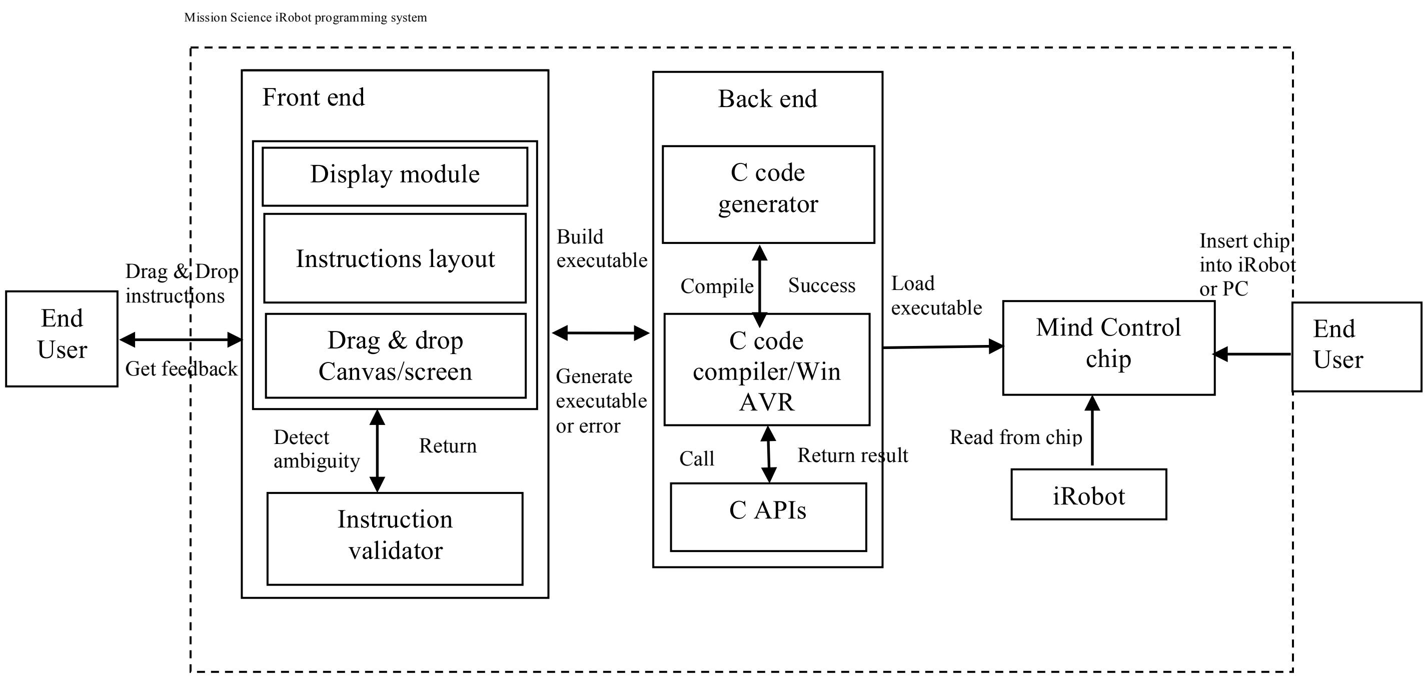


Figure 2: Element Relationship Diagram of the Mission Science iRobot project (Architected agile project)

3.3.2 Business Workflows

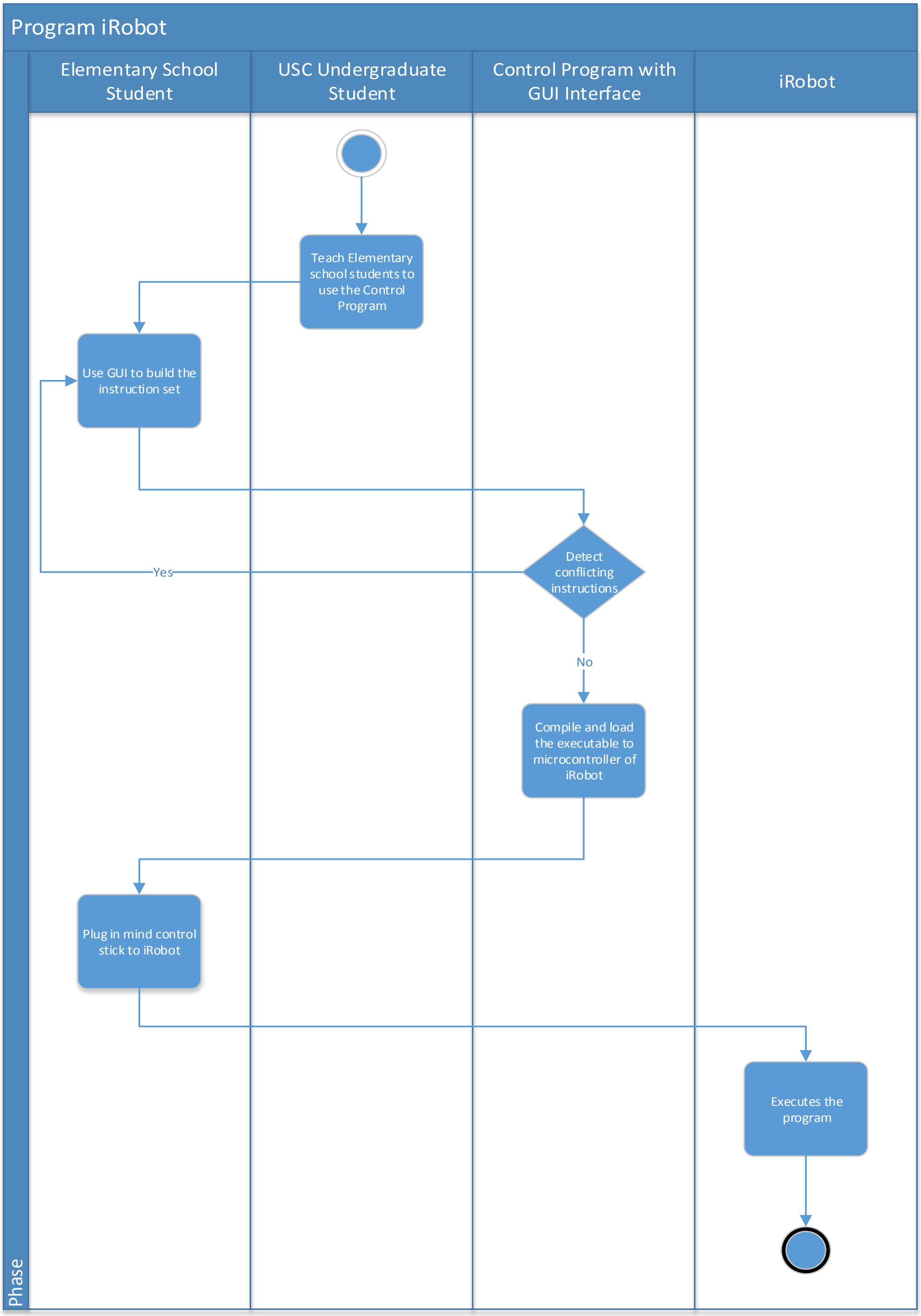


Figure 5: Business Workflow Diagram of Mission Science iRobot project

3.4 Organizational and Operational Implications

3.4.1 Organizational Transformations

- The need to train Mission Science community to maintain the system.
- Need to train the elementary school teachers to program and debug the instruction set in the new system.
- Expanded user sector for the iRobots would imply more maintenance support required.

3.4.2 Operational Transformations

- The elimination of the need to code in C language.
- Having the elementary school children use an Interface to control the iRobots.
- Having elementary school teachers and undergraduate students to guide elementary school kids on how to program the iRobot using the proposed drag & drop interface.
- Mission Science community need to maintain the source code related to drag & drop interface and the back-end C-code generator.