University of Puerto Rico
Mayagüez Campus

Programming Language for Health Care Protocols

ProtoX

Renier Velazco Santiago (801-12-9620)
Cristian F. Torres Collazo (802-13-8525)
Bryan N. Bonilla Rodríguez (802-16-0764)
Bryan J. Sanchez Maldonado (802-07-8188)
ICOM 4036 Programming Languages
Dr. Wilson Rivera

Introduction

We are living in an era of frameworks to solve different computational problems. Technologies are merging to simplify our jobs. Hence, there are more tools to be used when trying to have a solution to a real problem. The perfect moment to contribute to the area of healthcare is now. When chaos presents itself in a hospital, there could be numerous options to alleviate the situation. How do we know that we are taking the most optimized solution for the problem? That's when computation and engineers can contribute to the system. We know that human emotions could be part of the foreground when dealing with victims that suffer from an arbitrary condition or an accident. ProtoX is a programming language whose purpose is to provide a framework to solve complex scenarios when decisions are vital to resolving the conflict. We have decided to create a programming language to simplify the development of surgery protocols in hospitals. We have talked about ProtoX purpose but what about its implementation? The main functionality of ProtoX and where it excels is the simple creation of protocols for a given situation and its retrieval. The user needs to follow a guideline when creating this protocol to ensure its correctness. What constitutes these guidelines? Some are clarity, brevity and a specific set of instructions. When using the programming language users can create protocols. Users can also access these protocols via decision making from a given input stored in variables to use them in an emergency. These protocols are merely an algorithm to follow for a proper situation. The goal of this new programming language is to help new doctors and nurses to act quickly in the medical process to prioritize human life from a bad decision or misstep in practice.

Language Features

The language has three types of “objects”: hospitals, procedures and protocols. A hospital can perform many different procedures and each procedure has associated protocols. Our ProtoX language can perform the following operations:

- Create/delete a hospital.
- Show all hospitals.
- Add a procedure that is done to each hospital.
- Fetch the list of protocols related to a procedure.
- Fetch the list of protocols by hospital depending on which procedures they offer.
- List the procedures done in each hospital for all procedures.
- Create/delete a protocol.
- Edit an existing protocol.
- Assign a protocol to a procedure.
- Remove protocol from a procedure.

Example of a program

ProtoX 1.0: Examples of how the code will be implemented.

>>> add hospital (“Hospital Perea”)

>>> show hospitals

Hospital Perea

>>> add procedure (“Hospital Perea”, “CPR”)

>>> list protocols (“CPR”)

Cardiac life support

Airway Management

>>> list hospital procedures (“Hospital Perea”)

CPR

>>> list hospital protocols (“Hospital Perea”)

Ethics- Patient Consent

Cardiac life support

Airway Management

>>> show protocol (“ Airway Management“)

First priority is establishment or maintenance of airway patency. 1. Talk to the patient A patient who can speak clearly must have a clear airway. Airway obstruction by the tongue in the unconscious patient is often a problem. The unconscious patient may require assistance with airway and/or ventilation. If you suspect a head, neck or chest injury, protect the cervical spine during endotracheal intubation...

>>> edit protocol (“ Airway Management“)

Opens protocol text file in Vim. Saves edited protocol.

Implementation requirements and tools

Development tools:

- **PyCharm** as the IDE for development in Python.
- **GitHub** for version control.
- **PLY (Python Lex-Yacc)** for lexical parsing.

Project plan and timeline

Phases

Phase 1 - Project Plan (Feb 14)

This phase will consist in planning all parts of the project. It is basically developing a project plan document that will serve as a guide during development.

- Introduction
- Language Features
- Example of a Program
- Implementation Requirements and Tools
- Project Plan and Timeline

Phase 2 - Language Translator (Apr 23)

This phase will focus on developing the language and interpreter software components.

- Lexical analyzer (scanner)
- Syntax analyzer (parser)

- Intermediate Code

Phase 3: Final report and demo (Final Exam Day)

The final phase will dedicate itself to polishing the final product. In addition, the final report will be done and this will contain the following:

- Introduction
- Language tutorial
- Language reference manual
- Language development
    - Translator architecture
    - Describe the interfaces between the modules.
    - Describe the software development environment used to create the Translator.
    - Describe the test methodology used during development.
    - Show programs used to test your translator.
- Conclusions

Gantt Chart

Build a Programming Language

Programming Language for Health Care Protocols : **ProtoX**

Team Members: Bryan J. Sanchez Maldonado, Renier Velazco, Cristian Torres, Bryan N. Bonilla Rodríguez

Project Start: Mon, 2/11/2019

Display Week: 2

TASK	ASSIGNED TO	PROGRESS	START	END
Phase 1 - Project Plan				
Research	All	100%	2/11/19	2/19/19
Introduction	Renier Velazco	100%	2/11/19	2/19/19
Language Features	Bryan	100%	2/11/19	2/19/19
Example of Program	Bryan	100%	2/11/19	2/19/19
Implementation Requirements and Tools	All	100%	2/11/19	2/19/19
Project Plan and Timeline	Cristian F. Torres	100%	2/11/19	2/19/19
Phase 2 - Language Translator				
Lexical Analyzer	TBD		2/19/19	3/12/19
Syntax Analyzer	TBD		3/10/19	3/30/19
Intermediate Code	TBD		3/30/19	4/23/19
Phase 3 - Final Report and Demo				
Introduction	TBD		4/24/19	4/26/19
Language Tutorial	TBD		4/26/19	4/28/19
Language Reference Manual	TBD		4/29/19	5/1/19
Language Development	TBD		4/25/19	5/10/19
Translator Architecture	TBD		4/27/19	5/10/19
Description of Interfaces	TBD		4/28/19	5/3/19
Description of Software Development Environment	TBD		5/1/19	5/4/19
Description of Test Methodology	TBD		5/5/19	5/6/19
Working Program Samples	TBD		5/8/19	5/10/19
Conclusions	TBD		5/10/19	5/12/19