



East West University

Department of Computer Science and Engineering

Project Title

Design of a Simple Interface for Interactive Communication
between Matlab and other Programming Languages.

Supervised by

Dr. Shaikh Muhammad Allayear

Assistant Professor

Rahat Khan Mashar

2010-1-60-001

Aaquib Fehad Richie

2011-1-60-034

Abstract

Our project mainly works between two or more different programming language. It's act like an interpreter. Actually it is an Interface. In this project we are proposed an algorithm which is communicate with various programming language to get the final result.

There are many developer who are preferred various language. Our project will remove the complication of communication between programmers. In our project, we show that the input of a language is reached from the output of the other programming language. It will be a big work if we do it in big range with lots of experimental application.

Declaration

We hereby declare that, this project was done under CSE 497 and has not been submitted elsewhere for requirement of any degree or diploma or for any purpose except for publication.

Rahat Khan

ID# 2010-1-60-001

Department of Computer Science and Engineering

East West University.

Aaquib Fehad Richie

ID# 2011-1-60-034

Department of Computer Science and Engineering

East West University.

Letter of Acceptance

I hereby declare that, this thesis is from the student's own work and best effort of mine and all other source of information used have been acknowledge. This thesis has been submitted with my approval.

Dr. Shaikh Muhammad Allayear
Assistant Professor
Department of Computer Science and Engineering
East West University

Supervisor

Dr. Shamim Hasnat Ripon
Associate Professor and Chairperson
Department of Computer Science and Engineering
East West University

Chairperson

Acknowledgement

First of all we would like to thank Almighty Allah for giving us the strength and patience.

Our sincere gratitude goes to our parents for their all kindness and encouragement during our studies, without them this thesis would have been very hard to finish.

We would like to thank our supervisor Dr. Sheikh Muhammad Allayear utmost direction, constant helpful support and feedbacks.

Last but not the least, we express gratitude to our friends, seniors and juniors for providing effective suggestions and supporting us especially Pronaya Prosun Das.

Index

Chapter 1: Introduction	1-2
Chapter 2: Background	3-5
2.1 Background	4
2.2 Aims & Goals	4
2.3 Objectives	4
2.4 Motivation	5
2.5 Related works	5
Chapter 3: Model & Structure	6-8
3.1 Model	7
3.2 Explanation	7
3.3 Architecture	7
3.4 Flow	8
Chapter 4: Tools & Algorithm	9-13
4.1 Tools	10
4.2 Algorithms	11
Chapter 5: Conclusion	14-16
5.1 Summary	15
5.2 Limitations	15
5.3 Future Works	15
5.4 References	16

Chapter 1

Introduction

The idea has been generated from the complexity of group work in different programming language. It basically works as a media. It is mainly an interface. At first we have to know what an interface is.

Interface is,

A boundary across in which two independent systems meet and act or communicate with each other. In short, interface is a point where two systems, subjects, organizations, etc., meet and interact. In computer technology, there are several types of interfaces. Such as,

- User interface: The keyboard, mouse, menus of a computer system. The user interface allows the user to communicate with the operating system.
- Software interface: The languages and codes that the applications use to communicate with each other and with the hardware.
- Hardware interface- The wires, plugs and sockets that hardware device use to communicate with each other.

So an interface means to connect with or interact with.

They model key behaviors that need to be supported – by designing interfaces, you are forced to partition a big chunk of functionality into a set of behaviors. The amount of detail in each interface varies but the very act to thinking through behaviors will provide a more loosely coupled design.

rovides flexibility to change parts of a module/sub-system at anytime. This is especially relevant with projects where knowledge accumulates over time and domain understanding is unclear in during project inception. Interfaces help switch to a better implementation without adversely impacting the rest of the codebase (assuming, the contract stayed the same or only minimal changes were required).

Our work is also an interface but not for two or more system. It's an interface which is work on two or more different programming language.

- Collect data from different programming language.
- Also collect data from different dimension.
- Collect data from different types.
- Exchange the output of data between two or more languages.

Remove the distinction between two or more languages.

Chapter 2

Background

2.1 BACKGROUND

When any project are being develop, then there are many developer to solve it but there is one big problem. Every developer has their own choice of programming language to develop the project. If there is limited programmer to solve it and they prefer different language then what to do?

This is the background of our project.

2.2 AIMS AND GOALS

The aims of our project are to skip communication problem between programmers. If there is many programmer and they choose different language for doing their work portion, then our project will solve the problem. It will collect the data from the programs written in different languages and merge into a single string. Then it will pass the string data into other program which is written in other programming language through our algorithm.

2.3 OBJECTIVE

The objective of the project can be stated as follows:

1. Take input from a programming language.
2. Take input from any other programming language also.
3. Amalgamate the output of a language in a single string.
4. Convert the output into another programming language.
5. Merge the outputs from different language and show in a specific programming language.

2.4 MOTIVATION

In this project we use “Matlab” to create mediator between various kind of programming language. We think this is an innovative idea which never generate before. This is a new idea for all of us.

- Firstly-To learn a new language which is used in lots of research work and it’s challenging.
- Secondly-To get an experience to work with a real project.

2.5 RELATED WORK

As it’s a thesis based project there isn’t any significant existing project related to our project. But there are various interface related work exists.

There is a built-in MATLAB API (application programming interface) for Other Languages Interact with MATLAB[®] and MATLAB data types from other language applications; write C/C++/Fortran functions to call from MATLAB (MEX-files).

Engine applications are standalone C/C++ or Fortran programs that allow you to call MATLAB from your own programs, using MATLAB as a computation engine. Engine applications require an installed version of MATLAB; you cannot run the MATLAB engine on a machine that only has the MATLAB Runtime.

Chapter 3

Model & Structure

3.1 MODEL

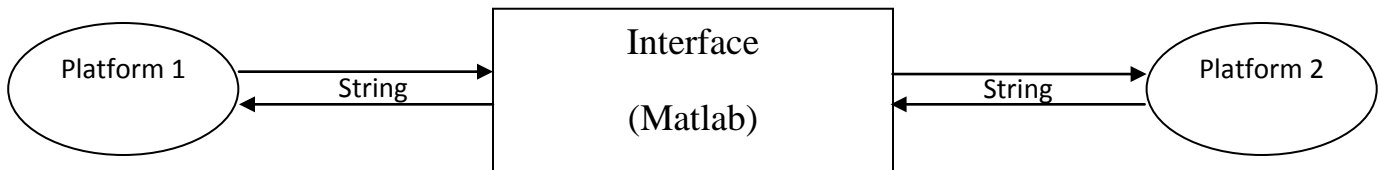


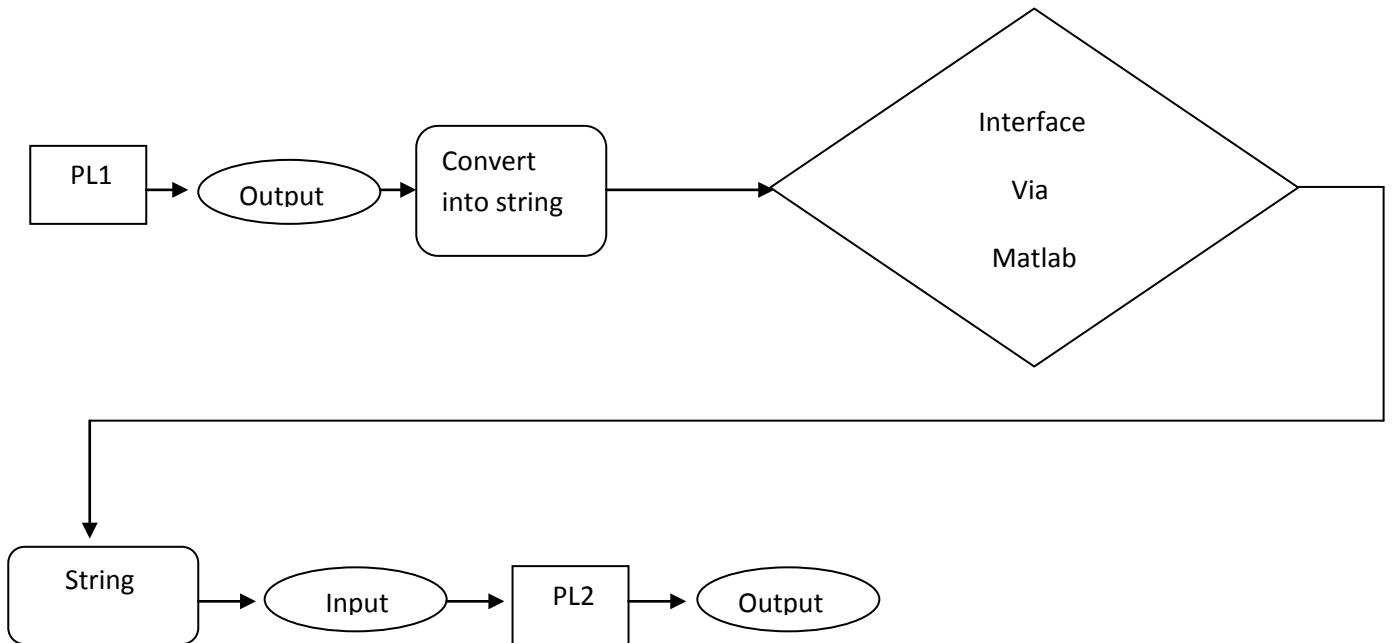
Fig 1: Model

3.2 EXPLANATION

Here platform 1 means a programming language whatever it is. Platform 2 means another programming language which is different from first programming language.

Firstly, there are some data generated by platform 1. It will convert into a single string. After that it passes through the interface. In the interface stage it converts the data which is needed in platform 2. Then the data will again amalgamate and go to the platform 2 as a string. This process is vice versa.

3.3 ARCHITECTURE:



If there are more than two languages then these output will be the input of next program of different language. As like-

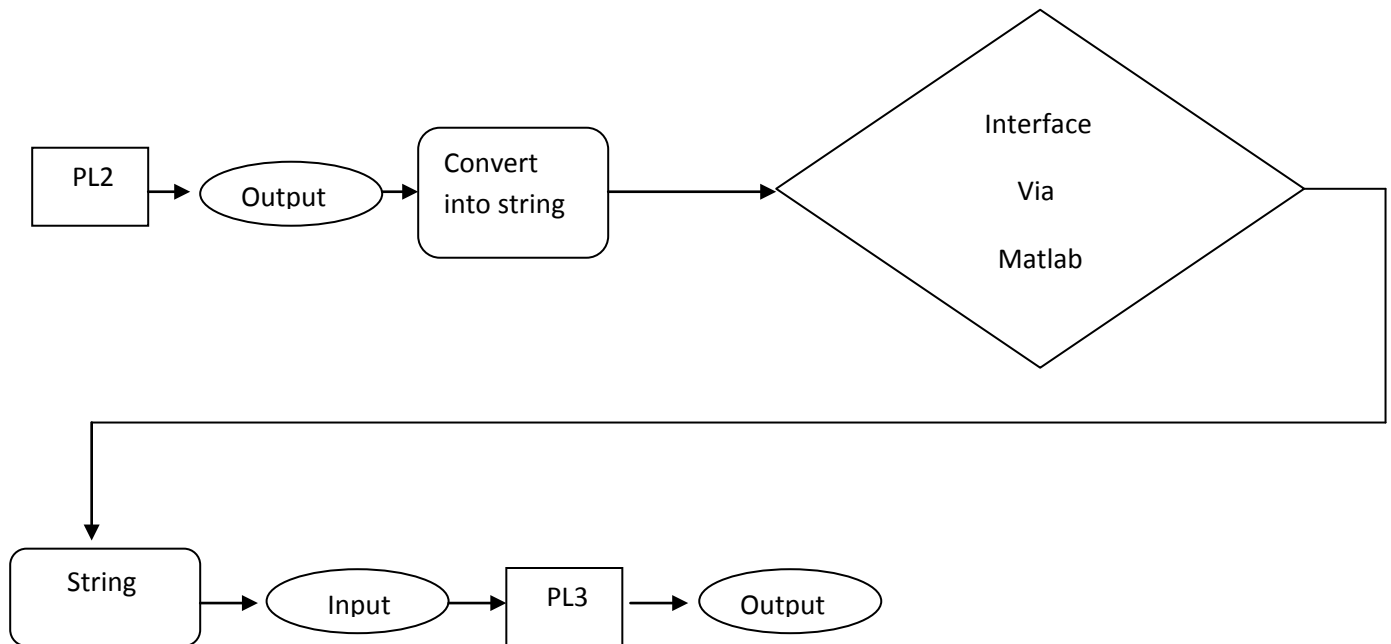


Fig 2: System Architecture

- *PL1- programming language 1*
- *PL2- programming language 2*
- *PL3-- programming language 3*

3.4 FLOW

The Flow shows how the interface works.

1. Doing work together with different programming language.
2. Group work in different programming language without knowing other language.
3. It converts the output from different platform.

Chapter 4

Tools & Algorithm

4.1 TOOLS

We have used MATLAB and Visual Studio 2010 for coding of our project.

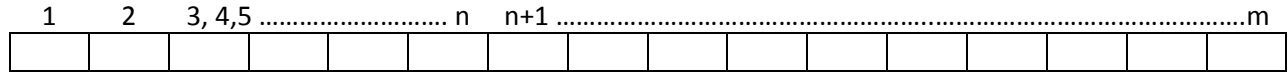
All the script files that are being used to generate random numbers and arrays and merge them into a single string are written in MATLAB. It was also used to receive the string from *.exe files and decodes the string to get raw data. The interface is mainly designed in MATLAB.

The Visual Studio 2010 is used to write the C/C++ and C# codes and generate the application files (*.exe). These applications can receive the string from MATLAB and decode it to get raw data. These can also send the data as a string to the MATLAB interface.

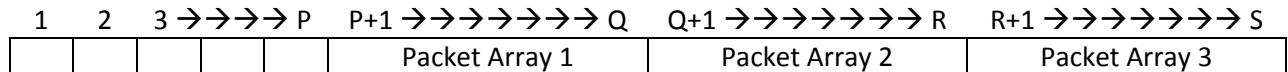
4.2 ALGORITHM

Before we start we need to know how a packet is formed.

Structure of Packet Array



Structure of Packet String



All the information stored in packet Array

```

Packet_array[1] = array_type;
Packet_array[2] = number of dimension;
Packet_array[3 to n];                               */{n=1+number of dimension}/*
Packet_array[(n+1) to m] = array_data;
Dim_size[number of array] = {packet_array[3] to packet_array[n]};
Length of (n+1) to m;                               */initially l=1/*
for (i=1 to number of dimension)
    length = length*packet_array [i+2])
*/let consider Array is the main array /*
    Array[dim_size[1]][dim_size[2]] to dim_size[number of dimension]];
for i=1 to dim_size[1]
    for j=1 to dim_size[2]
        ...
        for k=1 to dim_size[number of dimension]
            Array[i][j][...][k] = packet_array[index]
            Index++                               */where initially index=packet_array[n+1]/*

```

All the information stored in packet String

```

Packet[1] = C/C++ programs full path
Packet[2] = number of arrays
Packet[3 to p] = start index of each array           */p=2+number of array/*
Start_index[number of array] = {p,q+1,r+1,...z}     */value of number of array/*
Packet_array_1[] = {packet[p+1]to packet[q]}
Packet_array_2[] = { packet[q+1] to r }
Packet_array_3[] = {packet[r+1] to s}

```

PACKET ARRAY

Packet_array = null

1. Insert the data type into the packet array

Packet array

/ insert the data type /

0/1

2. Insert the number of array to the packet array

Packet_array[3]

/ as example number of dimension are 3/

3. Dimension

2	3	2
---	---	---

Packet_array

0/1	3	2	3	2
-----	---	---	---	---

4. Convert the data into single dimensional array

Array[][][0]

1	2	3
4	5	6

Array[][][1]

7	8	9
10	11	12

After converting this multidimensional array into one dimensional array it'll look like

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

After concating this to packet array it'll look like

0/1	3	2	3	2	1	2	3	4	5	6	7	8	9	10	11	12
-----	---	---	---	---	---	---	---	---	---	---	---	---	---	----	----	----

Element number $1 + 1 + 3 + (2*2*3) = 17$.

PACKET STRING

1. Path = "C:\...\Debug\program.exe"

Packet_string =

path

2. Number of array = 3

/lets imagine number of array is 3/

Packet_string =

path	3
------	---

3. Element of array_1=12

Element of array_2=15

Element of array_3=17

Packet_string =

path	3	12	15	17
------	---	----	----	----

4. Add the rest of packet arrays

Packet_array_1

packet_array_1

Packet_array_2

Packet_array_2

Packet_array_3

Packet_array_3

Packet_string =

path	3	12	15	17	packet_array_1	packet_array_2	packet_array_3
------	---	----	----	----	----------------	----------------	----------------

Chapter 5

Conclusion

5.1 SUMMARY

This is a thesis based project work. We can use it in broad way. We can solve communication problems between programmers by using this algorithm. In future we can make it large. That's all for us.

5.2 LIMITATIONS

Matlab is an advance language. There are many facilities which is not in other programming language. In Matlab there is an opportunity to change the dimension number of variable in run time but there is no option to change it in other programming language like C, C++. So when we work we get fixed maximum number of dimension in this project. This is the limitation of our project.

5.3 FUTURE WORK

In our project we show that how it works with C, C++, and C #programming language in one platform. In future we can work with more programming language using this algorithm. It will be a big opportunity to work together in a single project.

5.4 REFERENCES

- For all the MATLAB programming help
<http://se.mathworks.com/help/matlab/language-fundamentals.html>
- For all the C++ programming help
<https://msdn.microsoft.com/en-us/library/60k1461a.aspx>
- [3] For all the C sharp programming help
<https://msdn.microsoft.com/en-us/library/618ayhy6.aspx>
- [4] MATLAB API for Other Languages
<http://se.mathworks.com/help/matlab/programming-interfaces-for-c-c-fortran-com.html>
- [5] MATLAB Engine API for C, C++, and Fortran
<http://se.mathworks.com/help/matlab/calling-matlab-engine-from-c-c-and-fortran-programs.html>