UNDERSTANDING THE LivMach FRAMEWORK

Shreyak Chakraborty(C) 2013

Note:- THIS DOCUMENT IS WRITTEN IN THE C++ SYNTAX WITH PROPER SYNTAX HIGHLIGHTING //(using Notepad ++).

Abstract:
We introduce the alpha version of a C++ Computational Framework to simulate life processes in the body of a living multicellular organism by virtually replicating the data flow of the actual living being in real time.
LivMach Framework is an open source project on Sourceforge.net
We use various data structures to effectively simulate all components of a living organism 's body. Due to the absence of a Graphical User Interface(GUI), we use special indicator statements to display the flow of data between various parts of the virtual body.
Using this code,one can simulate the complete physical,mental and psychological behaviour of simple and complex multicellular organisms on low cost machines.

/*
LivMach Framework is an open source C++ code that enables the user to simulate a variety of processes of a living multicellular organism in real time. The source code is distributed under GPL v2.0 and is avaliable at www.sourceforge.net/projects/livmach

This paper is a description of the features of the LivMach Framework and the data flows in the LivMach body.
*/

The LivMach contains different classes and structures for different data flow simulations.

For using the response simulation, we have three global classes defined in the code.All these classes allow bi-directional data flow. Other classes also exist globally, but will be discussed later.
The main classes for response simulations are:

```
            class NCI_SYS           (Neural Communication Interface )
            class CNS_IO            (Central Nervous System)
```

```
              class RS_IO                   (Root Sensory System)


1. class NCI_SYS //this defines the brain of the virtual organism. Contains very important
brain functions.
   The members of this class interpret the stimulus given by other body parts and generate the
   corresponding
   response.The response is then transmitted to other required cells using the brain functions
   of this class.
   Some other member functions also help the Central Nervous System class to access the brain
   functions and
   maintain proper connection between the brain and other body parts.





2. class CNS_IO //this defines the central nervous system and is publicly derived from NCI_SYS
   This class works to establish a connection for data transfer between the Root Sensory System
   and the brain.
   The class RS_IO cannot directly access the brain(NCI_SYS) but has to go via CNS_IO.



3. class RS_IO //this defines the sensory organs of the virtual body
   This class contains the receptors to get the stimuls data from the main() function. Only
   objects of this class
   are created inside main().Some member functions of RS_IO called inside main() work as
   receptor proteins.The main() function is used
   is used to declare stimulii and these stimulii are passed as areguments to the member
   functions of RS_IO.


   The connection and flow of data bi-directionally is achieved by some global functions that
   link these classes
   together.


   // SIMULATION OF LIFE PROCESSES USING THE LifeProSim INBUILT MODULE

   /*
    We briefly explain the LifeProSim Inbuilt/Internal Module for simulating life processes in
    LivMach
We now use a number of structures to simulate various life processes in the livmach body. These
data structures
generally allow unidirectional data flow. The structures behave as organs as seen earlier(in
case of response
simulations). Any cell of a given organ(a structure) is defined as an object of that structure.

The organ systems defined are
     THE RESPIRATORY SYSTEM:          struct air_input         AIn
                                      struct air_process       APro
                                      struct air_output        AOut

     THE MAIN ENERGY CONTROL SYSTEM:  struct energy_input      EIn
```

```
                                        struct energy_process    EPro
                                        struct energy_output     EOut
                                        struct energy_usage      EAv


        Similarly, other organ systems can be defined with the required properties. These
        organs and organ systems
        implement the Data Flow Replication Method(DFRM) i.e replicating the data flow flow of
        a full fledged
        living system into a computer system.

In LivMach 1.0, the life processes start at the user command at TUI but end automatically.


  LifeProSim can also use OpenMP via GCC 4.8 to simulate bothe stimulus responses and life
  processes simultaneously.
  */

   In the module definition, the OpenMP clause
   # pragma omp parallel sections
   is used to command the threads in the current team to execute a different section in parallel.
In LivMach, the life process simulation and the stimulus response mechanism is executed
simultaneously making
it more like the body of a real organism.
The LifeProSim Module uses a global function:
void LP_Do(int a,int n)
to start and end a life process effectively. The integer n represents the loop control variable
which determines
the number of times a life process is run. The value of n is provided by the user.
The alpha version only supports respiration.
We can generalise the simulation procedure for life processes(intended for developers)
LivMach Framework uses following types of functions during execution of a life process in the
virtual body.
1. void LP_Start() -type            //starts a life process
2. void LP_End()  -type             //ends a life process
3. void LP_Do()  -type              //executes life process loop
4. void E_Abs()  -type             // absorbs energy from life process and utilizes it
5. void LP_Control() -type         //member function of 'brain' which controlls the life
process fully
```