International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 3– No.2, July 2012 – www.ijais.org

Inception of Medi Java: an Open Source Library for Medical Science

Subrata Sinha
Centre for Bioinformatics Studies, Dibrugarh University
Dibrugarh-786004, Assam, India

Paran Sundar Das
Centre for Bioinformatics Studies, Dibrugarh University
Dibrugarh-786004, Assam,India

Supriya Chakraborty
Centre for Bioinformatics Studies, Dibrugarh University
Dibrugarh-786004, Assam,India

Pallavi Dutta
Centre for Bioinformatics Studies, Dibrugarh University
Dibrugarh-786004, Assam,India

Bishnu Prasad Deori
Centre for Bioinformatics Studies, Dibrugarh University
Dibrugarh-786004, Assam,India

ABSTRACT

Medical Software development process needs valuable time and efforts of the developers and medical science experts of the respective field. As the software developers need to write the code from scratch. So there has to be some medical science library functions in the all frequently used programming languages to facilitate faster and efficient software development process. So an attempt has been made to investigate the existence of any such library functions in the existing programming language and to propose the possibility of such library in Java as Java API. Medi Java (Name proposed by the first author) is expected to cover all important areas of medical science. MediJava is intended to be an open source project.

General Terms

Object Oriented Programming; Open source.

Keywords

Open Source Library; Medi Java; Medical Science Library.

1. INTRODUCTION

The majority of research and development (R&D) in Medical Informatics can be divided into three branches. The first branch develops algorithm, the second branch facilitates the development of software's based in the algorithm developed and the third branch includes those who work on the software [1]

For rapid and economical software development developers prefer to use inbuilt libraries. Designing tools, libraries, and frameworks is one of the highest forms of design and programming. Thus, providing a tool, language, framework, etc., that makes the result of such work available to thousands is a way for programmers and designers to escape the trap of becoming craftsmen of one of a kind artifacts [2].

Development of any kind of software need inbuilt library, like Bioinformatics software developer if use Java , then there is a huge collection of Java library in the form of Bio Java package and classes for faster and reliable Bioinformatics Software development . Biojava is an open source project headed by Prof. Andreas Pralic, Senior Scientist, University of California, San Diego. Many open source bioinformatics packages are widely used by the research community across a wide variety of applications. Open source bioinformatics software has facilitated rapid innovation, dissemination, and wide adoption of new computational methods, reusable software components, and standards [3].

2. MATERIALS AND METHODS

2.1 MATERIALS

To design the class diagram of the open source library we used StarUML software which is a freely available Unified Modeling Language generator/ MDA platform running on win32 platform [4]. It shows the classes of a system, their interrelationship, the operations and attributes of the classes [5]. JDK 1.7.1 is a program development environment which is used to code, compile, debug, and test the library written in the java language [6]. The JDK tools used in library are java is the loader, javac is the compiler, and jar is the archiver [6]. The NetBeansIDE is open source and is written in the programming. It provides the services common to creating desktop applications. The NetBeans platform and IDE are free for commercial and non-commercial use and they are supported by Sun Microsystems [7].

2.2 Methods

1. The library creation starts from generating the UML class diagram where the different specialists of medical science and their equations are classified into an hierarchy of classes and interfaces. For library designing, classic life cycle model of SDLC is followed [8].

2. In the design phase we constructed the UML class diagram and accordingly the library is constructed using JDK 1.7.1

3. The library is made available to the developers by creating jar file.jar file can be created by-

```
jar cvf medi.jar *
```

Which mean the jar will be created for all .class files present in the current directory.

4. To test the library java application is developed using Net Beans IDE [9].

5. Library is added to the Net Beans IDE by going to the following options

Tools →Libraries →Library manager →Add jar →Press OK

6. Test case has been prepared for input to the developed application.

7. The test result is verified with MedCalc3000 calculator.

3 RESULTS

3.1 UML Diagram

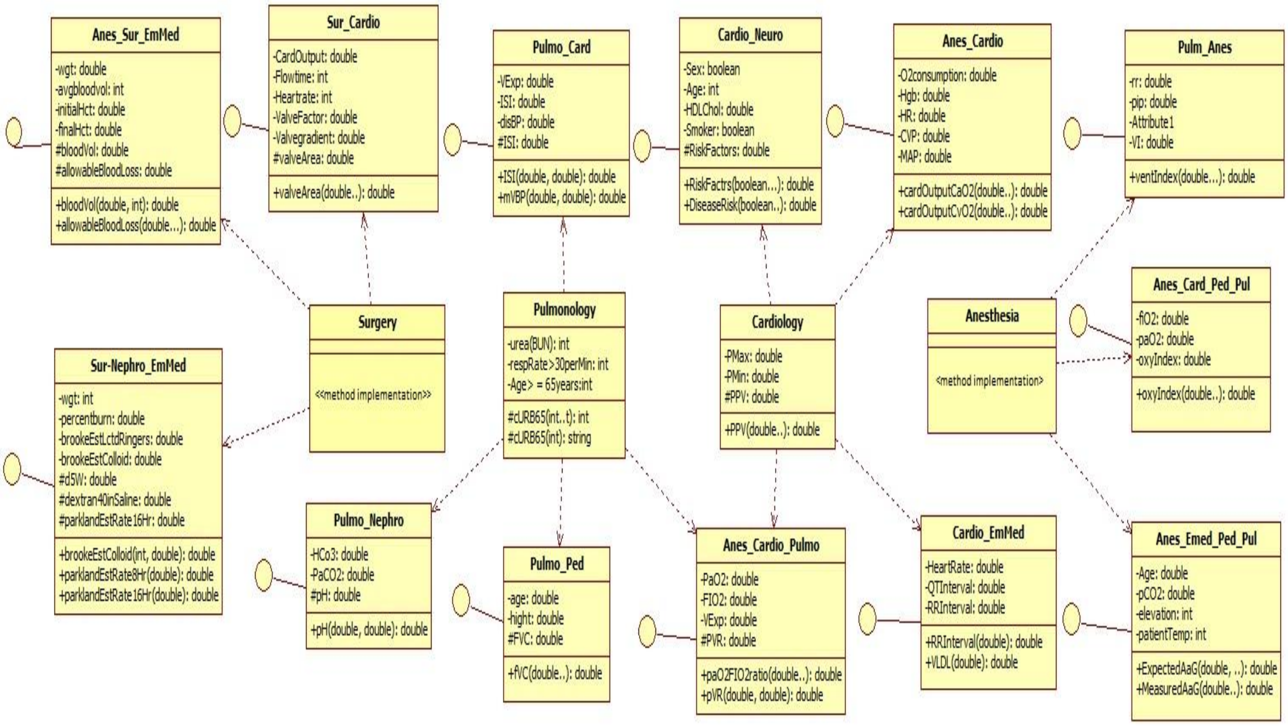


Fig 1: UML Class Diagram of the Librar

3.2 Library

Library with 14 interface and 4 classes with 47 methods has been created but the code is not included in the paper but the method declaration of the interfaces and classes of the library are given below.The library is made available to developers as . jar file.

3.2.1 Classes and Interfaces of the MediJava Library

3.2.1.1 Interfaces and Class for Anesthesia package:

interface Anes_Card_Ped_Pulm :

double oxyIndexOI(double FIO2,double meanAirwayPressure,double PaO2);

interface Anes_EmMed_Ped_Pulm :

double aAGrExpectedAaG(double Age);

double aAGrMeasuredAaG(double pCO2,double respQuot,double percentInspiredO2,double paO2,int elevation,int patientTemp);

interface Anes_Cardiac :

double cardiacOutputCaO2(double hgb,double o2sat,double paO2);

double cardiacOutputCvO2(double hgb,double o2vsat,double pvo2);

double cardiacOutputCo(double o2consumption,double CaO2,double CvO2);

double pressureAdjustedHeartRatePAdjHR(double HR,double CVP,double MAP);

double systolicPressureVariationSPV(double SysMax, double SysMin);

interface Pulm_Anes :

double ventIndexVI(double rr,double pip,double peep,double CO2);

The class Anesthesia implements the methods of
Anes_Card_Ped_Pul, Anes_EmMed_Ped_Pul, Anes_Cardiac, Pul_Anes.

3.2.1.2 Interfaces and Class of Cardiology package:

interface Anes_Card

double cardOutputCaO2(double Hgb,double O2Sat,double PaO2);

double cardOutputCvO2(double Hgb,double O2vSat,double PvO2);

double cardOutputCO(double O2Consumption,double CaO2,double CvO2);

interface Card_Emerg

double intervalCorrectRRInterval(double HeartRate);

double intervalCorrectQTICorrected(double QTInterval,double RRInterval);

double veryLowDensitylipoVLDL(double Triglycerides);

interface Anes_Pul_Card

double respQuotientR(double VCO2,double VO2);

double respQuotientVCO2(double VExp,double FECO2,double FICO2);

double respQuotientVO2(double VExp,double FIO2,double FEO2);

double oxyConsumpVO2(double VExp,double FIO2,double FEO2);

interface Card_Neuro

double CardVascularDiseaseRiskFactors(boolean sex,int age,double SysBP,double TotalChol,double HDLChol, boolean hypertnsnmedicatn,boolean Cig,boolean DM);

double CardVascularDiseaseRisk(boolean sex,int Age,double SysBP,double TotalChol,double HDLChol,boolean hypertnsnmedicatn,boolean Smoker,boolean diabetes);

The class Cardiology defines the following method-

public double pulPresVariatPPV(double PMax,double PMin)

The class Cardiology implements the methods of
Anes_Card,Anes_Pul_Card,Card_Emerg,Card_Neuro.

3.2.1.3 Interfaces and Class of Pulmonology package:

Interface Anes_Card_Pulmo

double modsCalcpaO2FIO2(double paO2,double FIO2) throws InvalidFIO2Exception

double pulmVasResistPVR(double meanPulmAtrialPresr,double lftAtrialPresr,double pulmflw);

interface Pulmo_Card

double co2ProdVCO2(double VExp,double FECO2,double FICO2);

double meanVascPrsrMVBP(double SysBP,double DiasBP);

double intNormRatioPrtmTmINR(double PtntProtm,double CtrlProtm,double ISI);

interface Pulmo_Nephro

double hndHsslbchEqnpH(double HCO3,double Paco3);

interface Pulmo_EmMed

interface pulmo_Ped

double pftPrdctValusforBoysMCalc(double Hight,double Age);

The class Pulmonology defines the following method-

public int CURB_65PneumoniaSeverityScore(int confabbrevMentalTestScore,int ureaBUN,int RespRate,

int diastolic,int systeolic,int Age)

The class Pulmonology implements the methods of Anes_Card_Pulmo, Pulo_Card, Pulmo_Nephro, Pulmo_EmMed.

3.2.1.4 Interface and Class of Surgery package:

interface Sur_Nepro_EmMed

double brookeEstLctdRingers(int wgt,double percentburn)throws InvalidPercentBurnException;

double brookeEstColloid(int wgt,double percentburn)throws InvalidPercentBurnException;

double D5W();

double dextran40inSaline(int wgt);

double dextranFreshFrozenPlasma(int wgt);

double evanEstNormalSaline(int wgt,double percentburn);

double evanEstColloid(int wgt,double percentburn);

double modifiedBrookeCrystalloidEstLctdRingers(int wgt,double percentburn);

double slaterEstLctdRingers();

double pThrxDgrColpsPerPnumo(double lungDmtr,double HmthrxDmtr);

double slaterEstFreshFrozenPlasma(int wgt);

double parklandEstTotalCrystalloid24Hours(int wgt,double PercentNonsuperficialBurnArea);

double parklandEstRate8Hr(double parklandEstTotalCrystalloid24Hours);

double parklandEstRate16Hr(double parklandEstRate8Hr);

interface Anes_Sur_EmMed

double bloodVol(int wgt,int AvgbloodVol);

double allowBloodloss(double bloodVol,double initHct, double finalHct);

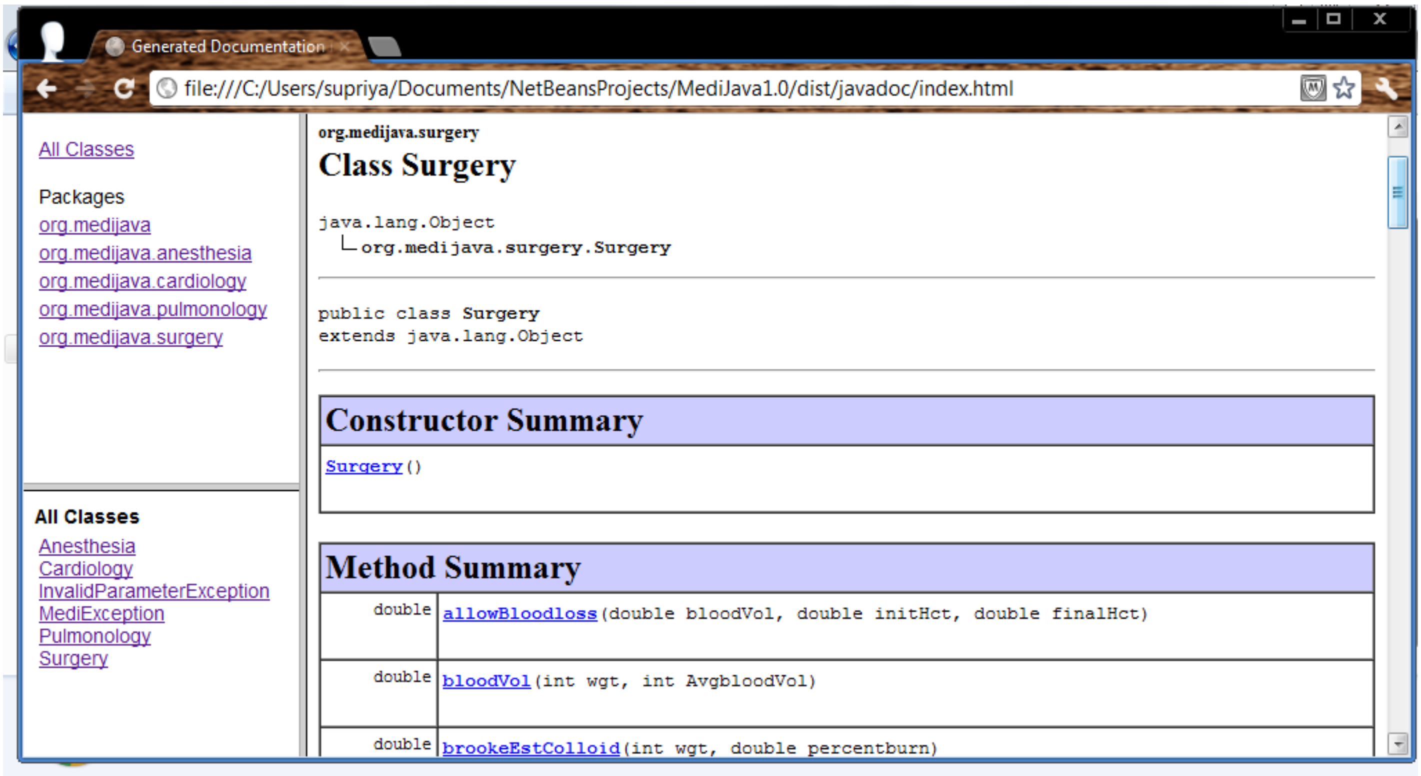
interface Sur_Cardio

double valveArea(double CardOutput,int Flowtime,int Heartrate,double ValveFactor,double Valvegradient);

The class Surgery implements the methods of Sur_Nephro_EmMed, Anes_Sur_EmMed and Sur_Cardio.

3.3 Documentation of Medi Java Library

The documentation is generated from NetBeansIDE which shows only four packages and their classes,which is shown in the Fig.2



file:///C:/Users/supriya/Documents/NetBeansProjects/MediJava1.0/dist/javadoc/index.html

All Classes

Packages
org.medijava
org.medijava.anesthesia
org.medijava.cardiology
org.medijava.pulmonology
org.medijava.surgery

All Classes
Anesthesia
Cardiology
InvalidParameterException
MediException
Pulmonology
Surgery

org.medijava.surgery

Class Surgery

```
java.lang.Object
  └org.medijava.surgery.Surgery
```

```
public class Surgery
extends java.lang.Object
```

Constructor Summary
Surgery()

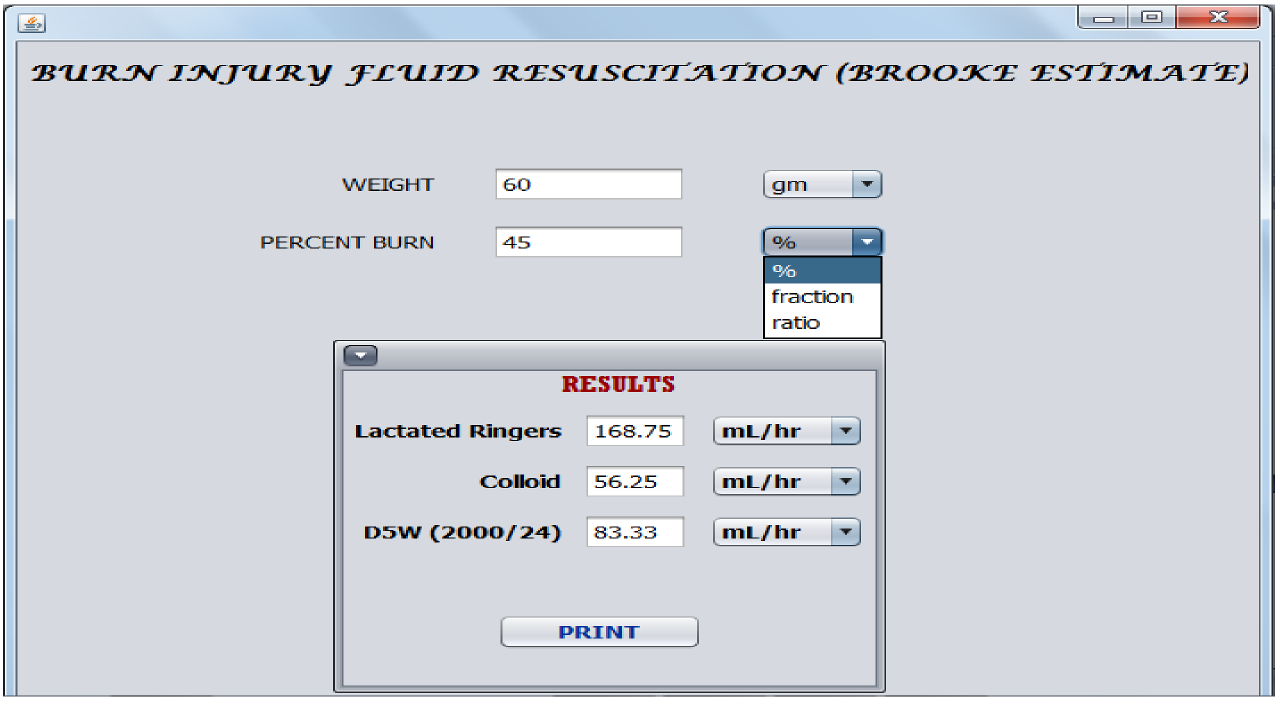
Method Summary	
double	allowBloodloss(double bloodVol, double initHct, double finalHct)
double	bloodVol(int wgt, int AvgbloodVol)
double	brookeEstColloid(int wgt, double percentburn)

Fig 2 : Snapshot of java documentation of the Medi Java Library

3.4 Library test result

To test the library a java application has been created using Net Beans IDE [10] which comprise the fields like Cardiology, Surgery, Anesthesia, and Pulmonology. The Snapshot of one of the interface is given below which test the burn injury function of that class After the application has been executed with some test cases and the result has been verified with MedCalc3000.

Total 47 test cases has been designed out of which one test case is given below in Table 1.



BURN INJURY FLUID RESUSCITATION (BROOKE ESTIMATE)

WEIGHT 60 gm

PERCENT BURN 45 %

RESULTS

Lactated Ringers 168.75 mL/hr

Colloid 56.25 mL/hr

D5W (2000/24) 83.33 mL/hr

Fig 3 : Interface of burn injury fluid resuscitation (Brooke Estimate)

Table 1 : Test Case with verified result

Test cases of Burn Injury Fluid Resuscitation(Brooke Estimate)			Library generated results	MedCalc results
A. Lactated Ringers	Weight	Percent Burn		
A1	60	40	150	150
A2	60	70	262.5	262.5
A3	65	90	365.625	365.625
B. Colloid	Weight	Percent Burn		
B1	60	40	50.00	50.00
B2	60	70	87.50	87.50
B3	65	90	121.88	121.88

4. DISCUSSION

The project Medi Java is still undergoing, the results shown in the paper is just a small portion of the entire project. In this paper only four specialization has been covered namely Anesthesia, Surgery, Pulmonology and Cardiology

4.1 Need

Lack of standard library for medical science has lowered the speed of development for Medical Science Software's; also the lack of such library has increased the cost of development.

4.2 Scope

Medi Java Library is an open source library, so there is tremendous amount of scope for further development and growth of the library

4.3 Limitation

The medical science is a very fast area of study, encompassing even one speciality will take much effort, however an initiative has been taken, to show the possibility of such library for medical science software developers.

5. CONCLUSION

Lack of standard library for medical science has lowered the speed of development for Medical Science Software's; also the lack of such library has increased the cost of development. The medical science is a very vast area of study, encompassing even one specialty will take much effort, and however an initiative has been taken, to show the possibility of such library for medical science software developers. The major challenge in the development of the library is to extract information from Medical Science Professionals for the design of methods for the Medical Equations and Clinical Criteria. Also creating library for medical science specialties not only signify about creation of methods for medical equations and clinical criterion. The present work has only considered medical equations and clinical criteria of only four specialization of medical science. There are total 31 specialties and in each specialty there are medical equations, clinical criteria, decision tree. The library can be designed for all specialties. Medi Java Library is an open source library, so there is tremendous amount of scope for further development and growth of the library. The study has not considered library for embedded system for medical equipments. The device programming will be considered as a continuation of Medi Java Project.

6. ACKNOWLEDGMENT

I would like to acknowledge Prof. John Hancock, Visiting Scientist at University of Cambridge, United Kingdom for assuring me for assessment and looking at the scope of the library and making suggestions for additions. Prof. Andreas Parlic, Senior Scientist, University of California, San Diego (UCSD) for providing feedback and Prof. Louis Leff, Clinical Professor of Medicine , University of Pittsburgh School of Medicine for providing single user licence of his Medcalc3000, even before payment for the software. Apart from that I would like to acknowledge Prof. Lefteris G. Gortzis, Adjunct Professor University of Patras, School of Medicine for showing his interest in my project. Dr K Narain, Scientist,RMRC-ICMR for his support.

7. REFERENCES

[1] An Introduction To Open Source Software Development, Diploma thesis of Steffen Evers of Technische Universität Berlin. P.19-21.

[2] The C++ Programming Language, Third Edition by Bjarne Stroustrup. Addison Wesley Longman, Inc. ISBN 0201889544 , p. 73

[3][Web Site] http://www.eaglegenomics.com/2011/04/eagle-genomics-sponsors-bosc-2011/

[4] Lee, Hyunsoo Kim, Jeongil Kim,Jangwoo Lee. StarUML 5.0 User Guide [Internet] , 2005[Updated 2005; cited May 2012] Available from: http://staruml.sourceforge.net/docs/user-guide%28en%29/toc.html

[5] Scott W. Ambler. The element of UMLTM 2.0 Style. Cambridge,USA: Cambridge University Press; 2005.P.60.

[6] Herbert Schildt.Java The Complete Refference.7th Edition.NewYork.McGrew-Hill publication ISBN: 978000 36774, p.75-80.

[7] Charlie Abela. Introduction to NetBeans Europe: University of Malta Press.

[8].Elias M. Awad. System Analysis and Design. 2nd Edition. India: Galgotia Publications pvt.ltd;2008,ISBN: 978-817 515 6180, p .40-48,201-202.

[9] Webopedia: Online Computer Dictionary for Computer and Internet. IT Business Edge Network. [Internet]. QuinStreetInc. 2012.[updated 2011Oct 22; cited 2012 June 12].Available from:www.webopedia.com/TERM/documentation.html

[10] E.Balaguruswamy, "Programming with Java A Primer" 4th Edition,McGrew Hills Pub;lication,New Delhi 2010,ISBN-13:978-0-07-014169-8.