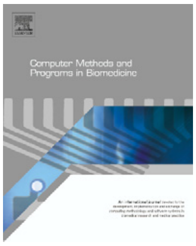
COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE 98 (2010) 191–203



journal homepage: www.intl.elsevierhealth.com/journals/cmpb

AIBench: A rapid application development framework for translational research in biomedicine

D. Glez-Peña[a,*], M. Reboiro-Jato[a], P. Maia[b], M. Rocha[b], F. Díaz[c], F. Fdez-Riverola[a]

[a] Dept. Informatics, University of Vigo, Campus Universitario As Lagoas s/n, 32004 Ourense, Spain
[b] Dept. Informatics, Universidade do Minho, Campus Gualtar, 4710-057 Braga, Portugal
[c] Computer Science Dept., University of Valladolid, Escuela Universitaria de Informática, Plaza Santa Eulalia 9-11, 40005 Segovia, Spain

ARTICLE INFO

Article history:
Received 13 June 2009
Received in revised form
11 November 2009
Accepted 10 December 2009

Keywords:
Scientific software development
Biomedical informatics
Open software
Reusable component model
Application framework

ABSTRACT

Applied research in both biomedical discovery and translational medicine today often requires the rapid development of fully featured applications containing both advanced and specific functionalities, for real use in practice. In this context, new tools are demanded that allow for efficient generation, deployment and reutilization of such biomedical applications as well as their associated functionalities. In this context this paper presents AIBench, an open-source Java desktop application framework for scientific software development with the goal of providing support to both fundamental and applied research in the domain of translational biomedicine. AIBench incorporates a powerful plug-in engine, a flexible scripting platform and takes advantage of Java annotations, reflection and various design principles in order to make it easy to use, lightweight and non-intrusive. By following a basic input–processing–output life cycle, it is possible to fully develop multiplatform applications using only three types of concepts: *operations*, *data-types* and *views*. The framework automatically provides functionalities that are present in a typical scientific application including user parameter definition, logging facilities, multi-threading execution, experiment repeatability and user interface workflow management, among others. The proposed framework architecture defines a reusable component model which also allows assembling new applications by the reuse of libraries from past projects or third-party software.

1. Introduction

Over the last few years, numerous authors have discussed potential and critical needs regarding the application of computing to biomedicine, emphasizing the necessity of exploiting the synergies between both disciplines to address current limitations and to promote relevant developments in both biomedical discovery and translational medicine [1]. In this context, an interesting summary of computing opportunities and challenges motivated by biomedical research and healthcare needs was summarized in the CRA-NIH 2006 Computing Research Challenges in Biomedicine Workshop Recommendations.[1] This report revealed a significant need to support both the development of new software tools and to provide the necessary support for software infrastructure and software engineering for biomedical researchers and healthcare professionals.

In response to these challenges, the clinical and translational research informatics domain is rapidly evolving from sparse and unrelated initiatives to two major well-established areas: (i) clinical research informatics (CRI), dedicated to

---

* *Corresponding author*. Tel.: +34 988 387015; fax: +34 988 387001.
E-mail address: dgpena@uvigo.es (D. Glez-Peña).
URL: http://sing.ei.uvigo.es/ (D. Glez-Peña).

[1] http://www.bisti.nih.gov/docs/CRA-NIH-Workshop-Recommendations-Final.pdf.

0169-2607/$ – see front matter © 2009 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.cmpb.2009.12.003

the development, use and evaluation of standards, models, processes and systems to improve the design, conduct and dissemination of clinical research [2], and (ii) translational research informatics (TRI), more concerned with the application of informatics theory and methods to translational research [3]. Although both areas overlap considerably, the former is more focused on developing practical applications for computer-aided medicine while the main goal of the latter is to provide fundamental support to translational research.

However, the rapid development of successful feature-rich applications containing advanced functionalities in the field of biomedical and clinical research still remains a major demand for smaller institutions lacking both human and financial resources. The situation worsens if we consider the software development effort required to deliver highly specialized applications usually demanding sophisticated user interfaces. Moreover, developing applications in an interdisciplinary and applied research context also presents a large number of particular requisites ranging from computational requirements to usability. Specific issues include (i) sharing of heterogeneous data, (ii) integrating third-party or previously developed algorithms, (iii) cross-platform compatibility, (iv) ability to repeat workflows while changing a few parameters or input data, (v) extensive use of logging messages to monitor the progress of long processes, (vi) establishing values for a high and variable number of parameters before running experiments and (vii) taking the maximum advantage of multithreading capabilities in highly demanding tasks, among others.

In the global context of computer science and software development, a typical approach to cope with these kinds of problems is to make use of an application framework, which can be seen as a semi-finished application and a reusable architecture design [4]. Therefore, in recent years frameworks have become very popular, especially in web application development where Ruby on Rails,[2] Symfony,[3] Spring,[4] JSF[5] or Apache Struts[6] are examples of some of the most successful alternatives for deploying scientific applications as web services [5]. Nowadays, there are general frameworks for almost any kind of software including object-oriented desktop applications (MFC,[7] Netbeans,[8] Eclipse[9]), software testing (JUnit[10]), compiler generation (Bison,[11] Javacc[12]), multimedia (WindowsMedia,[13] ffmpeg,[14] GStreamer[15]), virtual reality (Vega Prime,[16] VR Juggler,[17] CAVELib[18]) and middleware (CORBA,[19] EJB[20]). Nevertheless, the actual benefits of applying such general scalable software environments to the development of specific biomedical applications are clearly insufficient, mainly due to the special requirements of computer-assisted biomedical and clinical research areas.

With the aim of giving a more adequate support to the particular needs of several areas belonging to the theoretical and clinical biomedicine domain, different focused frameworks were also successfully developed in the C++ language during the last few years. This endeavour was particularly evident in the area of medical imaging. In this context, the Medical Imaging Interaction Toolkit (MITK) implements a free open-source software system for the development of interactive medical image processing software [6]. MITK combines the Insight Toolkit (ITK[21]) and the Visualization Toolkit (VTK[22]) for currently offering functionalities for data visualization, processing and interaction. In the same line, MeVisLab [7], a development environment for medical image processing and visualization, as well as IGstk [8], a high-level component-based framework providing common functionality for image-guided surgery applications, make use of ITK and VTK toolkits (among others) for giving support to specific biomedical applications. Another example of a successful software platform in this area is JULIUS [9], an extensible framework for medical data processing and visualization. From a different perspective, and more focused in the development of research based image-guided navigation software, is the SIGN framework [10], which provides the developer with a platform specifically designed for image-guided therapy and aids the rapid development of new applications.

From a broader perspective, but also related with the goal of giving support to the development of software techniques and processes used to manage images of the human body for clinical purposes, there are two successful application frameworks: the Multimod Application Framework (OpenMAF) [11] and MARVIN [12]. OpenMAF implements an open-source framework for rapid development of multimodal applications coded in the C++ language. OpenMAF supports several types of biomedical data where its interactive visualization approach helps the user to interpret complex datasets. In addition, the framework supports different input–output hardware devices being based on a collection of portable libraries. A different approach, but also coded in the C++ language, is the MARVIN project. MARVIN implements a medical research application framework where different modules can be plugged together in order to provide the functionality required for a specific scenario. In the MARVIN framework, application modules work on a common patient database that is used to store and organize medical data. As in the case of OpenMAF, MARVIN

---

[2] http://rubyonrails.org/.

[3] http://www.symfony-project.org/.

[4] http://www.springframework.org/.

[5] http://java.sun.com/javaee/javaserverfaces/.

[6] http://struts.apache.org/.

[7] http://msdn.microsoft.com/en-us/library/d06h2x6e(VS.80).aspx.

[8] http://platform.netbeans.org/.

[9] http://wiki.eclipse.org/.

[10] http://junit.org/.

[11] http://www.gnu.org/software/bison.

[12] https://javacc.2dev.java.net/.

[13] http://www.microsoft.com/windowsmedia/.

[14] http://ffmpeg.mplayerhq.hu/.

[15] http://gstreamer.freedesktop.org/.

[16] http://www.multigen.com/products/runtime/vega_prime/.

[17] http://www.vrjuggler.org/.

[18] http://www.mechdyne.com/integratedSolutions/software/products/CAVELib/CAVELib.htm.

[19] http://www.corba.org/.

[20] http://java.sun.com/products/ejb/.

[21] http://www.itk.org/.

[22] http://www.vtk.org/.