Available online at www.sciencedirect.com
ScienceDirect
Journal of Biomedical Informatics 41 (2008) 106–123
Journal of Biomedical Informatics
www.elsevier.com/locate/yjbin

caCORE version 3: Implementation of a model driven, service-oriented architecture for semantic interoperability

George A. Komatsoulis [a,*], Denise B. Warzel [a], Francis W. Hartel [a], Krishnakant Shanbhag [a], Ram Chilukuri [c], Gilberto Fragoso [a], Sherri de Coronado [a], Dianne M. Reeves [a], Jillaine B. Hadfield [a], Christophe Ludet [b], Peter A. Covitz [a]

[a] *National Cancer Institute Center for Bioinformatics (NCICB), 2115 E. Jefferson St., Suite 5000, Rockville, MD 20852, USA*
[b] *Oracle Corporation, 600 Oracle Parkway, Redwood Shores, CA 94065, USA*
[c] *Semantic Bits, LLC, 12359 Sunrise Valley Dr Suite 260 C, Reston, VA 20191, USA*

Received 2 November 2006
Available online 2 April 2007

Abstract

One of the requirements for a federated information system is interoperability, the ability of one computer system to access and use the resources of another system. This feature is particularly important in biomedical research systems, which need to coordinate a variety of disparate types of data. In order to meet this need, the National Cancer Institute Center for Bioinformatics (NCICB) has created the cancer Common Ontologic Representation Environment (caCORE), an interoperability infrastructure based on Model Driven Architecture. The caCORE infrastructure provides a mechanism to create interoperable biomedical information systems. Systems built using the caCORE paradigm address both aspects of interoperability: the ability to access data (syntactic interoperability) and understand the data once retrieved (semantic interoperability). This infrastructure consists of an integrated set of three major components: a controlled terminology service (Enterprise Vocabulary Services), a standards-based metadata repository (the cancer Data Standards Repository) and an information system with an Application Programming Interface (API) based on Domain Model Driven Architecture. This infrastructure is being leveraged to create a Semantic Service-Oriented Architecture (SSOA) for cancer research by the National Cancer Institute's cancer Biomedical Informatics Grid (caBIG™).
Published by Elsevier Inc.

Keywords: Semantic interoperability; Model Driven Architecture; Metadata; Controlled terminology; ISO 11179

1. Introduction

The rise of high-throughput biomedical research tools has given scientists and clinicians data of unprecedented depth, timeliness and diversity to attack the problem of alleviating human cancer. However, this diversity brings with it the problem of integration and interoperation; that is, making disparate data sets created by a wide range of individuals available to others for the purposes of performing analyses that span multiple data types. For example, many researchers would like to correlate the results of high-throughput gene expression microarray experiments with clinical outcomes or toxicology results, or correlate loss of heterozygocity with tumor susceptibility to certain therapeutic compounds. Even though there are often multiple sources of this sort of data, they are often inaccessible to information systems at runtime (so-called data 'stovepipes'), or the data is insufficiently annotated to determine if successful reuse is possible.

The solution to this problem is to build data systems utilizing an architecture that facilitates such interoperability. The IEEE Standard Computer Dictionary defines interoperability as the "ability of two or more systems or components to exchange information and to use the information that has been exchanged" [1]. From this

[*] Corresponding author.
E-mail address: komatsog@mail.nih.gov (G.A. Komatsoulis).

1532-0464/$ - see front matter Published by Elsevier Inc.
doi:10.1016/j.jbi.2007.03.009

definition it is possible to decompose interoperability into two distinct components: the ability to exchange information, and the ability to use the information once it has been received. The former process is denoted as 'syntactic interoperability' and the latter 'semantic interoperability'. A small example suffices to demonstrate the importance of solving both problems. Consider two persons who do not share a common language. They can speak to one another and both individuals will recognize that data has been transferred (they can also probably parse out individual words, recognize the beginning and end of message units, etc.). Nevertheless, the meaning of the message will be mostly incomprehensible; they are syntactically but not semantically interoperable. Similarly, consider a person who is blind and one who is deaf, but who both utilize a single language. They can attempt to exchange information, one by speaking and one by writing, but since they are incapable of receiving the messages, they are semantically but not syntactically interoperable.

The creation of an interoperable data system, therefore, requires several elements, including a convenient mechanism that provides a clear and consistent interface into a data repository and a source of terminology whose meaning is clear and unambiguous to those who would record and use the data maintained in that repository. Object-Oriented Application Programming Interfaces (APIs) built using the Model Driven Architecture (MDA) paradigm can begin to address the first problem, and controlled terminology available at runtime can help with the latter. There is however, a third piece to this problem; a mechanism to bind the controlled terminology to a model driven, object-oriented data system. Such descriptive information, commonly referred to as semantic metadata, provides a formal description of the meaning of the types of data that are supplied by the individual classes and attributes that are part of the data system as well as what constitutes a valid value for each attribute of the classes.

The creation of an infrastructure that supports such a 'semantically annotated' data system is the central innovation of caCORE version 3. The caCORE infrastructure provides a runtime-accessible terminology service and a standards-based, runtime accessible metadata repository that is used to bind the terminology to a domain model-based information system. This infrastructure has been successfully used to create a reference implementation, cancer Bioinformatics Infrastructure Objects (caBIO), but is fully extensible to any arbitrary data system, regardless of subject.

1.1. Introduction to components of caCORE

Version 3 of caCORE consists of three major parts: (1) a primary technology stack encompassing three major components depicted in the center of Fig. 1; (2) two major enabling technology components; and (3) one supporting technology. This is diagrammed in Fig. 1. At the top of the primary technology stack are the cancer Biomedical Informatics Objects, caBIO, the interoperable data system;[1] at the bottom of the stack is the Enterprise Vocabulary Services or EVS, supplying the controlled terminology that is leveraged to provide semantics for caBIO (or any other data system that utilizes the caCORE methodology). Between these two components is the cancer Data Standards Repository or caDSR, a system for storing semantic metadata, that acts as the glue between the object-oriented data system and the controlled terminology. The supporting technology component is a Common Security Module or CSM, which is designed to be readily integrated into systems designed along caCORE lines. The CSM contains a user provisioning tool for managing rights given to users within the system. Finally, there are two pieces of enabling technology, (1) the caCORE Software Development Kit [2] that is used to generate 'caCORE-like' systems, and (2) the Semantic Integration Workbench, an end-user application with a graphical user interface (GUI) that assists in creating the semantic metadata that is stored in the caDSR. This manuscript focuses primarily on the integration of the three components of the primary technology stack to enable semantic interoperability. The caCORE infrastructure is distributed under a non-viral open source software license which allows for any commercial or non-commercial reuse of the software, its components, or source code. Each component is described in further detail in upcoming sections.

2. Creating an interoperable, service-oriented architecture

2.1. Model Driven Architecture

All caCORE components utilize a software development strategy known as Model Driven Architecture or MDA, that is developed by the Object Management Group (OMG http://www.omg.org). The MDA approach posits that prior to creating an object-oriented software system, the designer should create a graphical model of the functions, components, and behavior of the system that is independent of the computer language that the system will ultimately be created in. This model is then used to create the implemented software system, sometimes using programs called code generators to create either a skeleton of or the complete software system. Within the MDA paradigm, a Platform Independent Model (PIM) is captured using the Unified Modeling Language (UML). As a result, all caCORE systems are modeled in UML prior to implementation.

[1] Any model driven, object-oriented data system could be at the top of the caCORE primary technology stack. In a sense, the caBIO system is a 'reference implementation' of 'caCORE-like' data system. We include it in the discussion of caCORE because it is the example implementation, and because the caBIO system is developed and released concurrently with the other components of caCORE.