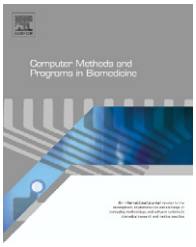
COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE 106 (2012) 114–125

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



ARDEN2BYTECODE: A one-pass Arden Syntax compiler for service-oriented decision support systems based on the OSGi platform

Matthias Gietzelt[a], *Ursula Goltz*[b], *Daniel Grunwald*[b], *Malte Lochau*[b], *Michael Marschollek*[a,1], *Bianying Song*[a], *Klaus-Hendrik Wolf*[a,*]

[a] *Peter L. Reichertz Institute for Medical Informatics, University of Braunschweig-Institute of Technology and Hannover Medical School, Muehlenpfordtstr. 23, 38106 Braunschweig, Germany*

[b] *Institute for Programming and Reactive Systems, University of Braunschweig-Institute of Technology, Muehlenpfordtstr. 23, 38106 Braunschweig, Germany*

ARTICLE INFO

Article history:
Received 10 November 2011
Accepted 11 November 2011

Keywords:
Arden Syntax
Java Bytecode
Decision support
Knowledge based systems
Compiler
Service oriented architecture

ABSTRACT

Patient empowerment might be one key to reduce the pressure on health care systems challenged by the expected demographic changes. Knowledge based systems can, in combination with automated sensor measurements, improve the patients' ability to review their state of health and make informed decisions. The Arden Syntax as a standardized language to represent medical knowledge can be used to express the corresponding decision rules. In this paper we introduce ARDEN2BYTECODE, a newly developed open source compiler for the Arden Syntax. ARDEN2BYTECODE runs on Java Virtual Machines (JVM) and translates Arden Syntax directly to Java Bytecode (JBC) executable on JVMs. ARDEN2BYTECODE easily integrates into service oriented architectures, like the Open Services Gateway Initiative (OSGi) platform. Apart from an evaluation of compilation performance and execution times, ARDEN2BYTECODE was integrated into an existing knowledge supported exercise training system and recorded training sessions have been used to check the implementation.

1. Introduction

Recently sensor enhanced health information systems have been proposed as one among many measures to alleviate the effects of demographic change [1]. Utilizing repeatedly or continuously performed measurements by sensor systems detailed information of an individual's state of health can be produced [2]. The information hidden in the gathered data could empower each individual to live a healthier life [3], to detect developing diseases earlier [4], and to effectively manage chronic diseases [5]. Giving back the patient more responsibility for his own health is expected to have a positive impact on health care costs.

One limiting factor is the patient's lack of knowledge to interpret measurements in the context of their personal health to make informed decisions. Therefore, they need to be assisted. Personalized decision support systems could help to bridge this gap.

The goal of our research was to develop components for decision support systems (DSSs) that can be seamlessly integrated into sensor enhanced health information systems. The

* *Corresponding author*. Tel.: +49 531 391 2126; fax: +49 531 391 9502.
E-mail address: Klaus-Hendrik.Wolf@plri.de (K.-H. Wolf).
[1] Present address: Peter L. Reichertz Institute for Medical Informatics, TU Braunschweig and Hannover Medical School, Carl-Neuberg-Str. 1, 30625 Hannover, Germany.

0169-2607/$ – see front matter © 2011 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.cmpb.2011.11.003

knowledge should be represented in an easy-to-understand, transportable, and preferably standardized form to support individualization of decision support.

2. Background

To archive a high level of knowledge abstraction and ease the sharing of knowledge the Arden Syntax for Medical Logic Modules (MLMs) has been chosen to represent health-related knowledge. Arden Syntax is well established as a way to represent medical knowledge. It has been standardized by Health Level 7 (HL7) [6] and incorporates a number of concepts, making it easy to express rules in the medical domain in a manner close to natural language. The promise is that experts from the domain are able to understand the expressed logic and validate the coded knowledge [7]. In the process we developed Arden2ByteCode [8], a compiler to translate MLMs formulated in Arden Syntax directly to Java Bytecode (JBC) [9]. Arden2ByteCode has been released under an open source license and is available to all interested parties to deploy, enhance and for educational purposes [8]. We hope that Arden2ByteCode will broaden the application of the Arden Syntax supporting the already existing commercial and non-commercial systems with an easy to access, use, study, enhance, and integrate alternative option. Arden2ByteCode only depends on a Java runtime environment and is thereby able to compile new or changed MLMs directly inside a target system based on Java.

The demonstrated integration of Arden2ByteCode in the OSGi Service Platform [10] shows its usability in service-oriented architectures. The resulting package is developed for and used in the research project "design of environments for ageing (GAL)" [11–13]. A first evaluation uses the Arden2ByteCode compiler in an automated training control setting, a real world use case, to show the flexibility and performance of the resulting system [14]. The ability to change the rules directly in the runtime system enables trainers to individualize the automated training control.

2.1. *Related work*

The Arden Syntax is an established programming language specifically designed for analysis of medical data [15]. Early publications of the syntax date back to 1990 [16]. In 1992 the Arden Syntax has been standardized by the American Society for Testing and Materials [17]. In 1999 HL7 took over the development of the syntax starting with version 2.0. The current version of the Arden Syntax is version 2.7 passed in 2008 [18].

Since the Arden Syntax specification was published, Arden Syntax compilers and the corresponding run time environments have been developed and deployed in many institutions. Many compilers translate or pre-process Arden Syntax into another high-level programming language, to be afterwards compiled to machine code or interpreted. Examples include pseudo code [19], C/C++[20], MUMPS [21], Java [22] and, CLIPS [23].

In 1991, Hripcsak et al. described their compiler in [19]. The authors proposed a compiler–interpreter pair for the execution of MLMs. Their system divides the execution of MLMs into two steps. At first the MLM is parsed and converted to pseudo-code. Secondly, the pseudo-codes are executed by an interpreter on a virtual machine. The interpreter or virtual machine was implemented on different architectures like a personal computer and a mainframe.

In 1993, Gao et al. developed a pre-compiling software to translate MLMs into C++[20]. One year later, a special issue about Arden Syntax was published in the Journal of Computers in Biology and Medicine. Several developed Arden Syntax compilers were reported in this special issue [24–26]. In 1996, McCauley et al. created a compiler which converts MLMs to MUMPS code [21]. In 2002, Karadimas et al. published the Arden/J architecture to increase the portability of MLMs, translating them to Java source code [22]. The described Arden/J compiler produces one Java class for each MLM.

Most of the afore mentioned implementations of Arden Syntax were developed specifically for an institution. The shortcomings in the Arden Syntax to define the interactions with the runtime environment and the access to data make sharing of MLMs between different implementations at least difficult. Therefore, it is important to mention, that the Arden Syntax has been successfully integrated into products in the health care market. Example products include Siemens Soarian, AGFA ORBIS, Health VISION, and Medexter Moni.

Besides the vendor or site-specific implementations of Arden Syntax some efforts have been made to develop independent and reusable compilers. In 2007, Medexter announced the completion of an Arden Syntax software suite written in Java [27]. The software contains an integrated development environment, an Arden Syntax compiler and some other components for the integration. It is commercially available under different license conditions and payment schemes.

Besides commercially developed Arden Syntax compilers, some open source initiatives exist. The Evidence-based Guidelines and Decision Support System (EGADSS) is an attempt to interface decision support systems (DSSs) and electronic medical records (EMRs) in order to aid physicians provide high quality care [23]. The developed Arden Syntax compiler ([28]) converts MLMs into CLIPS (C Language Integrated Production System), a LISP-like interpreted expert system shell language. This project has promoted the application of Arden Syntax, however, we have met some shortcomings in the parser (regarding e.g. arithmetic expressions) during our testing.

Besides implementation of the syntax, the language itself is under continuous development. Currently Fuzzy Arden Syntax has been suggested as an extension to formulate the uncertain knowledge in medicine [15]. A corresponding Fuzzy Arden Syntax compiler has been implemented and a systematic test of the Fuzzy Arden Syntax compiler and its use in data management system of the Vienna General Hospital is scheduled according to [15]. Arden2ByteCode does not support fuzzy constructs yet.

3. Design considerations

In the GAL research project software development is based on the OSGi platform [29]. OSGi is a service-based dynamic component model for the Java programming language and a