Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 37 (2014) 457 – 464
Procedia Computer Science

The 1st International Workshop on Meta-modelling for Healthcare Systems (MMHS)

Archetypes based meta-modeling towards evolutionary, dependable and interoperable healthcare information systems

Gunnar Piho[a,b,*], Jaak Tepandi[b], Douglas Thompson[a], Triinu Tammer[b], Marko Parman[b], Viljam Puusep[b]

[a]*Clinical and Biomedical Proteomics Group, University of Leeds, Beckett St, Leeds LS9 7TF, UK*
[b]*Department of Informatics, Tallinn University of Technology, Akadeemia St. 15A, Tallinn 12618, Estonia.*

Abstract

Information systems have to work correctly and securely as expected (dependability); have to be able to communicate and understand each other's data (interoperability); and have to be able to change in an evolutionary way similarly to how the organizations and their business processes are changing (evolutionary). To write software we must know the requirements; to prescribe the requirements we have to understand the domain; to understand the domain we have to analyze and model one. The domain or application domain (e.g. banking, healthcare, clinical laboratory and etc.) can be anything to which computing can be applied. In analyzing and modeling domains we use archetypes and archetype patterns as meta-models. We explain what the archetypes and archetype patterns are and how we utilize them in the development of domain models, requirements, and software in order to meet the dependability, interoperability and evolutionary criteria.

Peer-review under responsibility of the Program Chairs of EUSPN-2014 and ICTH 2014.
Keywords: theory and practice of meta-modelling with application in healthcare; case studies of meta-modelling in healthcare applications; archetypes and archetype patterns; laboratory information management systems; dependability, interoperability and evolutionary criteria of information systems;

1. Introduction

We present a framework for the development of domain models, requirements and software. This framework is a part of Sentry (sample entry) software for CBPG (Clinical and Biomedical Proteomics Group, Leeds Institute of Cancer and Pathology, University of Leeds) and uses a combination of different model-driven development technologies like archetypes and archetype patterns [1], triptych software development [2], Zachman Framework [3]

* Corresponding author. Tel.: +372 511 1236; fax: +372 620 2311.
E-mail address: gunnar.piho@computer.org

Peer-review under responsibility of the Program Chairs of EUSPN-2014 and ICTH 2014.
doi:10.1016/j.procs.2014.08.069

and software factories [4] to cope with the dependability [5], interoperability [6] and evolutionary [7] issues of software.

Dependability, interoperability and evolutionary criteria are extremely important for laboratory software. Any laboratory software has to work correctly and securely as expected (dependability); has to be able to communicate and understand data from diverse software systems (interoperability); and has to be able to change in an evolutionary way similarly to how the organizations and their business processes are changing (evolutionary). In research laboratories like CBPG, the business processes are constantly changing and different research groups within the same research laboratory may require different business processes and different or differently organized data. This is why we decided not only to develop just laboratory software, but decided to move towards the software factory concept.

By its nature Sentry is a LIMS (Laboratory Information Management System) software factory. According to ASTM (American Society for Testing and Materials) LIMS Standard Guide [8], LIMS represents a class of computer systems designed to manage laboratory information. Software factory [4] defines domain-based development artefacts (models, languages, tools, and others) that can be used to automatically generate software. In software factory all the models are code (and not only documentation) artefacts.

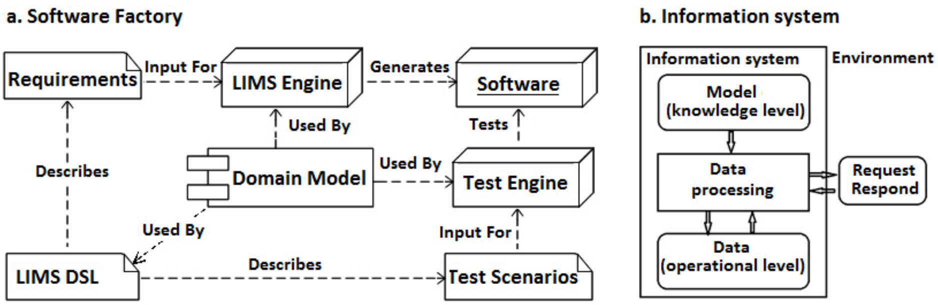

Fig. 1. General Architecture of (a) Sentry (LIMS software factory) and (b) information systems (adapted from [7])

The general architecture of Sentry is depicted in Fig.1a and includes LIMS DSL (domain specific language), LIMS Engine and a Test Engine. The requirements are specified using LIMS DSL and LIMS Engine is used to generate or change the software according to these requirements. The Test Engine validates these requirements with respect to the domain model and verifies the generated software. The central part of this architecture is a domain model of the laboratory. This laboratory domain model contains archetypes and archetype patterns (AAP) and some laboratory specific knowledge designed using AAP as meta-models. By software we mean an information system (Fig.1b), which processes requests from an environment and also responds to the environment. In doing so, the data (operational-level information) processing occurs in accordance with the model (knowledge-level information). This model includes all of the knowledge described either by AAP, domain model or requirements.

The paper is organized as follows. Section 2 introduces AAP. Section 3 exemplifies the use of AAP in a real life project. Section 4 explains how in our understanding with AAP it is possible to move towards dependability, interoperability and evolutionary information systems. In Section 5 the related works are outlined and we conclude in Section 6.

2. Archetypes and archetype patterns (AAP)

AAPs were originally introduced by Arlow and Neustadt [1]. Business archetype patterns (namely *Product*, *Party*, *Order*, *Inventory*, *Quantity* and *Rule*), composed of business archetypes (e.g. *Person's Name*, *Address*, *Phone Number*, etc.) are the universal information models describing the universe of discourse of businesses.

For instance, the *Party Archetype Pattern* is illustrated in Fig.2. The *Party* archetype is used to describe persons (e.g. person name, DoB, gender and address) and organizations. As persons and organizations can be in different roles (e.g. the same person may be mother, patient, etc.) in different relationships (e.g. she may be mother of Paul,