Science of Computer Programming 121 (2016) 34–54

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

Early knowledge organization assisted by aspects

Eduardo Barra *, Jorge Morato

Computer Science Department, Universidad Carlos III de Madrid, Avda. Universidad 30, 28911 Leganés, Madrid, Spain

ARTICLE INFO

Article history:
Received 22 April 2015
Received in revised form 14 July 2015
Accepted 19 July 2015
Available online 29 July 2015

Keywords:
Advanced separation of concerns
Aspect oriented programming
Ontologies
Early aspects
Requirements engineering

ABSTRACT

Throughout the lifecycle of software products, changes arise. The repercussions of these changes increase in more advanced stages of the development process. The aspect-oriented proposal is a software development paradigm that provides principles to manage changes. However, the current proposals for the development of the early stages of software do not provide an effective means for functional changes and updates. This paper presents a knowledge modeling methodology in the aspect-oriented context focused on the early stages of the software development. This methodology aims to ease the software evolution through a standardized knowledge representation, a multidimensional organization, and an appropriate advanced separation of concerns. The proposal of a conceptual modeling of early-aspect provides the means for an organization of multidimensional knowledge. The use of ontological techniques under the principles of knowledge-engineering allows specification of early knowledge. This paper presents a case study developed using the methodology and is used as a base to evaluate the methodology, comparing the efficiency in the changes against classic approaches. The results show that it is possible to ease the changes with the proposal presented in this paper.

1. Introduction

One of the major matters in software factories is related to facing the change throughout the software development lifecycle. The changes are due to the dependency on the context, the different development activities, and the diverse roles of the stakeholders. Often, the variability of such factors implies drastic different changes that can bring new problems, resulting in an unexpected final behavior and increasing the probability of failure. This situation reveals the necessity of developing models that provide the means to make changes in an efficient way on the code, the design, the analysis, and the requirements specification.

In the software engineering area, the most important advances in the development of complex systems have been achieved through several proposals based on the separation of concerns principle. In its most general form, this principle proposes specifying separately each concern of a system [1], where a concern is a specific requirement that must be taken into account to satisfy the software product goal [2]. In this sense, the complex activity about identifying, dividing in modules, encapsulating, and manipulating software product concerns, that are relevant to a concept, a goal, or a purpose, has been called advanced separation of concerns [3]. Some authors have stated that a concern generally is any matter of interest in a software system and that they are mainly conceptual [4]. Accordingly, in an advanced separation of concerns,

* Corresponding author.
E-mail addresses: ebarra@kr.inf.uc3m.es (E. Barra), jmorato@inf.uc3m.es (J. Morato).

http://dx.doi.org/10.1016/j.scico.2015.07.007
0167-6423/

the conceptualization of concerns depends on the stage of the software development lifecycle and on the viewpoint under which it is created.

A proposal that boosted the research in the advanced separation of concerns has been the Aspect Oriented Programming [5]. Such a proposal arose as a possible extension that could be applied to any programming paradigm. However, the Aspect Oriented Programming (AOP) has flourished through its symbiosis with the Oriented Object (OO) paradigm. The AOP, after being used in the codification stage, has spread to other stages, including up to the early stage of Requirements Engineering (RE). In this path several approaches have arisen trying to manage concerns that appear in different models created along the software product lifecycle [6–8]. Currently, every approach for OO software development processes, applying techniques and methods from the Aspect-Oriented (AO) paradigm, is called Aspect-Oriented Software Development (AOSD). Regarding the AOSD, the proposals with higher acceptance due to its effectiveness are focused on the final stages (codification). The proposals focused on the design stages have acquired an acceptable effectiveness in the academic environment but not in the software factories. However, in the early stages there are a lack of systematic proposals, generating some skepticism about its application.

The first step in any software development process is to capture the nature of the software product to build to its further analysis. This stage has been made concrete in the discipline known as RE and therefore has become the first part of the early stages. In this area some studies have been conducted that indicate the magnitude of the consequences of neglecting the early stages. Among the most relevant we can find are the ones undertaken by the Standish Group; these point out that in the last CHAOS report [9] an average of 39% projects met the goals of on-time delivery, on budget and with the required features and functions. Over the last few years, we can observe an important increase in the rate of projects meeting all of its requirements, compared to the 2% found by the first survey performed by GAO in 1979 [10]. However, the advances over more than 33 years, still require further improvement in order to get to an acceptable success rate. In this sense, in 2004, NASA [11] summarized the impact of the mistakes detection, and the results from nine years of studies show that the cost of solving a mistake in the design stage is 7.3 times the cost in the early stages, 25.6 times in the codification stage, and 117 times in the test stage. Therefore, the possibilities of change in the early stages of a software product are a factor with great repercussions in the failure.

The modeling resulting from the RE is oriented towards the understanding of the problem domain, but with the goal of obtaining the first solution lines. In other words, the early stages intertwine the RE specification with the description of high level abstraction solutions in the software architecture. This establishes a commitment of explicit interrelationship between the requirements specification and the logical architectonic representation.

This work studies the means provided by the advanced separation of concerns through AO in order to support a multidimensional modeling in early stages, reducing the complexity in the software product, but without endangering its semantic integrity, since the advanced separation of concerns can supply the means for an effective concerns knowledge management in order to get the efficient evolution of a software product. It was planned to start a research based on this reasoning with the goal of providing the means towards the OO software product evolution in the product software early stages because improving flexibility to change in the early stages improves the process quality, and therefore the quality of the software product.

The different proposals studied for the early stages based on AO paradigm are focused on the division and composition of the concerns, but are not oriented to the evolution of the software product. In order to get the OO software product evolution, it is necessary to manage the knowledge of it modeled in an effective way. However some problems were found in managing the concerns semantics on the diverse AO raisings in the early stages. The ontologies are the most used formalism for the representation and management of the knowledge. In this line we observed the means provided by the Ontology Engineering (OE), a discipline engendered inside the Knowledge Engineering (KE).

In this way, this article proposes an efficient knowledge modeling methodology in the early stages of the software product development. The goal is to make the software evolution easier through a standardized knowledge representation, with a multidimensional organization which allows an appropriate advanced separation of concerns in the AOSD context.

In developing this methodology, the need for AO metamodel that promotes the integration of different proposals aimed at specific stages of AOSD was perceived. However, the different studied meta-model proposals did not supply the means for the effective modeling of early stages. Therefore, we have identified the need to develop a framework for the modeling of early stages through a conceptual model.

According to these ideas, the article is structured as follows. In Section 2 the contributions and problems in the current proposals for early stages modeled of a software product based on AO paradigm are exposed. In Section 3 the conceptual modeling proposal of Early Aspect (EA) is shown. Section 4 introduces the “Early Knowledge Organization assisted by Aspects” (EKOA) methodology and the development of one case study. Finally in Section 5, we carry out an evaluation in order to demonstrate that representing the EA knowledge through the EKOA methodology is efficient.

2. Aspect oriented software development in early stages

The scope of this article is limited to the research and innovation of the early stages for a lifecycle based on the AO paradigm. We need to define the limits for these stages in software development. Likewise, we expose different current approaches of the AO paradigm for early stages modeling of a software product.