Entertainment Computing 4 (2013) 131–142

Contents lists available at SciVerse ScienceDirect

Entertainment Computing

journal homepage: ees.elsevier.com/entcom

Building and mining a repository of design pattern instances: Practical and research benefits

Apostolos Ampatzoglou, Olia Michou, Ioannis Stamelos

Department of Informatics, Aristotle University, Aristotle University Campus, PO Address 54124, Thessaloniki, Greece

ARTICLE INFO

Article history:
Received 10 February 2012
Revised 15 June 2012
Accepted 1 October 2012
Available online 12 October 2012

Keywords:
Software engineering
Computer games
Design patterns
Repository

ABSTRACT

Design patterns are well-known design solutions that are reported to produce substantial benefits with respect to software quality. However, to our knowledge there are no scientific efforts on gathering information on software projects that use design patterns. This paper introduces a web repository of design patterns instances that have been used in open source projects. The usefulness of such a repository lies in the provision of a base of knowledge, where developers can identify reusable components and researchers can find a mined data set. Currently, 141 open source projects have been considered and more than 4500 pattern instances have been found and recorded in the database of the repository. The evaluation of the repository has been performed from an academic and a practical point of view. The results suggest that the repository can be useful for both experienced and inexperienced users. However, the benefits of using the repository are more significant for inexperienced users.

1. Introduction

A repository is a shared database about engineered artefacts, such as software, documents and information systems [10]. In the field of software, several repositories of open-source projects that can be used as means of knowledge and experience sharing are available. Such repositories (e.g. Sourceforge, CodeHouse), provide data to the fast developing open source research area [44] and to software engineering researchers in general. Even though such repositories contain a great amount of information, sometimes it is difficult to browse, due to scalability problems, data integrity problems, etc. Reuse repositories and especially open source repositories are the basic infrastructure for software reuse, due to its significant economic impact. Nevertheless, they have introduced difficulties in finding and locating reusable software artefacts.

Additionally, design patterns are software engineering techniques that are considered to aid in software system development. In [22], the authors attempted to identify whether there are common problems among different system designs and if common solutions might be introduced. Moreover, a qualitative evaluation of the proposed solutions implies that the use of design patterns leads to adaptability and extensibility, as each design pattern supports independence of possible future changes [22]. The possibility of creating reusable components from design patterns is discussed in [7,26], where the authors introduced the idea of componentizing and reusing design patterns. Furthermore, in [3], the authors investigated the possibility of selecting design pattern instances as reuse granules, in white box reuse.

Despite the fact that during the last years the research on design patterns appears to flourish, no repositories concerning design pattern instances in software projects have been identified in the literature. In this paper we present a web repository (Percerons – PattErn Repository and Components Extracted from OpeN Source software) of object-oriented design pattern instances and a web tool which is used to mine it [18]. Currently, the repository contains more than 4500 pattern instances, extracted from 141 Java open source projects. Up to this point, we have investigated and recorded all stable computer games identified in a well known source code repository[1]. The selection of games was based on the fact that game development is an active topic in OSS communities [44] but game developers are more probable to write code without prior design activities [27]. Thus, a repository that will enhance the design process might prove useful. The benefits from such an attempt are expected to assist both practitioners and academics in the following ways:

- Practitioners
  - The re-users will be given a smaller search domain than forges, that will provide application specific components

E-mail addresses: apamp@csd.auth.gr (A. Ampatzoglou), stamelos@csd.auth.gr (I. Stamelos).

[1] www.sourceforge.net

1875-9521/$ - see front matter
http://dx.doi.org/10.1016/j.entcom.2012.10.002

- The components that are recorded in the database of the repository involve design pattern participants[2] and therefore, the rationale of their design is documented
- Similarly, some internal quality attributes of the extracted components, such as coupling, cohesion and complexity, are expected to improve compared to any other non-pattern component [2]

• Academics
- The software engineering researchers can easily obtain datasets on design patterns, without having to mine vast source code forges.

The user of Percerons is expected to have two possible motivations for accessing the search engine. (a) He/She is interested in implementing a game requirement, but does not know where to look. Percerons will provide a set of classes that implement the complete or a part of the desired functionality. (b) He/She is interested in retrieving several instances of a specific design patterns for any possible reason (education, research etc.). In order to validate the abovementioned benefits we conducted an experiment, in order to evaluate the usefulness of the repository from the practitioner's and researcher's point of view. In the next section, the research state of the art on reuse repositories is presented. Section 3, presents an overview on game development and game design. In Section 4, the structure of repository is presented. Section 5 deals with the empirical validation of the repository, from both an academic and a practical point of view. In Sections 6 and 7 discussion and threats to validity are provided. Finally, Section 8 presents conclusions resulting from this work and proposals for future research.

2. Reuse repositories

Current trends in software reuse indicate that the design and implementation of open source repositories follow a rather holistic approach, since not only programmers, but software managers and users are also influenced. Repositories should support the 'produce → manage → consume' cycle and its stakeholders [6,15]. In [15], a generic architecture for software repositories which contains modules that satisfy the needs of all involved parties is presented. Accordingly there is a production module (for component producers), a management module (for reuse managers), a consumption module (for component reusers or consumers) and an infrastructure module (providing common services to the other modules). More specifically, in case of the management module, the repository should provide the means to manage the reuse process at the organizational level by enabling reuse managers to observe specific quantifiable metrics associated with the reuse of the available components. Consequently, a set of numerous requirements should be satisfied for the repository to be considered as successful. Dominant role play search, retrieval and the emerging technology of semantic web that is used for component description are discussed in [14,32,40]. Moreover, requirements such as specification publication, user notification, version management, dependency management should also be included in the set.

As the main challenge that a component approach will meet is dealing with change, the substitutability of its parts can be achieved only if components are properly specified. Faceted classification as a representational method for classifying reusable artefacts is an interesting approach, according to which it organizes the search space in search-related facets [17,35]. A faceted classification provides a vocabulary for cataloguing reusable components and an organization mechanism for these components. The effectiveness can be assessed through various metrics including the well known precision (high precision means that a high fraction of retrieved instances are relevant) and recall (high recall means that only few relevant components are not retrieved), search effort, user satisfaction, and ease of use [20,31,36]. The role of the semantic technology is prominent as metadata are ideal for component description through the facets. In literature in the domain of software design, quality and reuse, several approaches have been found concerning software engineering repositories. On the other hand none of these approaches deals explicitly with design patterns.

Firstly, in [1] a widely accessible repository of software development artefacts (e.g. code, models and test cases) is proposed. The repository would enable software developers and researchers to assess software engineering tools and techniques. In [16], a repository of FLOSS data and analyses is proposed. This repository named OSSmole is a collaborative project which collects, shares and stores compatible data and analyses of FLOSS development for research purposes. It aims to mine FLOSS source code repositories and provide resulting data and summary analysis at open source products.

ROSE is another software engineering repository based on open source products. It is developed to aid students and educators during software engineering courses or lessons on open source software development [29]. Another type of software engineering repository is introduced in [24], where among others a proof-of-concept prototype metric repository is described. In [33,47], the importance of software repositories aiming at reuse is proved as the first introduces a knowledge-based repository scheme for storing and retrieving business components, whereas the latter explores component classification and retrieval methods with an overriding concern for automation. In [30], the authors introduced a benchmark repository of faulty and correct software that would enable unification of diverse experimental results regarding software testing techniques. Agora system described in [42] is a software prototype that allows automatic discovery and retrieval of software components from the web and their characterization with introspection. The object of this work is to create an automatically generated and indexed worldwide database of software products classified by component model. Agora combines introspection with Web search engines to reduce the costs of bringing software to, and finding components in, the software marketplace.

Finally, Maracatu [23] is another approach for retrieving software modules from CVS repositories. The Maracatu service has the following modules: (a) a CVS module for accessing the repositories for reusable components, (b) an Analyser module for analysing the code to determine if it is suitable for indexing, (c) a Download module for downloading (checking out) the source code, and (d) a Search module for searching using queries. This functionality is provided with an Eclipse plug-in.

3. Game development

We have chosen to validate the usefulness of the proposed repository in the domain of game development, because computer game design is one of the most modern and fast growing trends in computer science [37]. Additionally, it is a common sense that games play a very important role in modern societies concerning economy and lifestyle.

The game industry is now considered to be one of the most powerful in the business spectrum [21,50]. The fact that computer games currently rival television and films in both market size and cultural impact [21] can justify why game development is

[2] As pattern participant we mean every class that plays a specific role in a design pattern. A definition of the pattern participants is given in [25].