The Journal of Systems and Software 000 (2017) 1–19

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Mining domain knowledge from app descriptions

Yuzhou Liu[a,b], Lei Liu[a,b], Huaxiao Liu[a,b,*], Xiaoyu Wang[a,b], Hongji Yang[c]

[a] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, China, Changchun 130012, China
[b] College of Computer Science and Technology, Jilin University, Changchun, China
[c] Centre for Creative Computing, Bath Spa University, Corsham SN13 0BZ, England United Kingdom

ARTICLE INFO

Article history:
Received 1 November 2016
Revised 19 July 2017
Accepted 12 August 2017
Available online xxx

Keywords:
Domain analysis
Feature extraction
App descriptions
Data analysis

ABSTRACT

Domain analysis aims at gaining knowledge to a particular domain in the early stage of software development. A key challenge in domain analysis is to extract features automatically from related product artifacts. Compared with other kinds of artifacts, high volume of descriptions can be collected from App marketplaces (such as Google Play and Apple Store) easily when developing a new mobile application (App), so it is essential for the success of domain analysis to gain features and relationships from them using data analysis techniques. In this paper, we propose an approach to mine domain knowledge from App descriptions automatically, where the information of features in a single App description is firstly extracted and formally described by a Concern-based Description Model (CDM), which is based on predefined rules of feature extraction and a modified topic modeling method; then the overall knowledge in the domain is identified by classifying, clustering and merging the knowledge in the set of CDMs and topics, and the results are formalized by a Data-based Raw Domain Model (DRDM). Furthermore, we propose a quantified evaluation method for prioritizing the knowledge in DRDM. The proposed approach is validated by a series of experiments.

1. Introduction

With the development of software industry, there always exist some products similar or related to the software to be developed. Whether these products come from inside of the company itself or extensive markets, they provide the software development with reference assets, such as requirements, use cases, architecture and frameworks (Lian et al., 2016b). Based on these assets, domain analysis is conducted to gain knowledge in the early stage of software development, helping the developers understand and define a particular domain accurately. This supports a quick start of the development process and benefits the reuse of source code or other higher level life cycle artifacts. In this way, the success of domain analysis can effectively improve the product competitiveness, such as less time-to-market and high quality products (Lisboa et al., 2010).

In domain analysis, feature is a basic notation used to describe a user-perceived characteristic of a system and feature extraction is one of core activities to gain such information (Mefteh et al., 2016b). In the process of feature extraction, texts related to the products are always taken as an important analyzing object because they are a kind of main assets recording information of software system. In general, software related texts can be classified into three types (Bakar et al., 2015): 1) given by the developers to support or record the software development process, such as requirement specifications; 2) also given by the developers but intended for potential users to introduce the software, such as descriptions, brochures; 3) given as feedback by software users, which is often expressed as reviews or comments. Initially, most of domain analysis methods focus on utilizing the first type of texts, and rely upon analysts reviewing them to gain domain knowledge. Thus, it is very labor-intensive (Kang et al., 1990; Kang et al., 1998). In recent years, texts of the second and third type from various sources have grown fast, more and more data can be collected easily. Such high volume of texts makes it difficult to understand or analyze manually. By introducing data technology into the field of domain analysis, many researches propose approaches that (semi-)automatically generate feature diagrams, clustered requirements, keywords or direct objects through systematic or quantitative processes of data analyzing. So as to mine domain knowledge from such natural language-based documents to better support the subsequent software development process.

Mobile application (App) is a software system carried on mobile terminals and grows in popularity rapidly (Martin et al., 2016). App descriptions are the introductions of App products and belong to the second type of texts introduced above. Compared with the first type of texts, App descriptions can be collected more easily

* Corresponding author.
E-mail address: liuhuaxiao@jlu.edu.cn (H. Liu).

http://dx.doi.org/10.1016/j.jss.2017.08.024
0164-1212/

Table 1
An example of App descriptions.

Instagram is a simple way to capture and share the world's moments. Follow your friends and family to see what they're up to, and discover accounts from all over the world that are sharing things you love. Join the community of over 500 million people and express yourself by sharing all the moments of your day––the highlights and everything in between, too.
Use Instagram to:
• **Post photos and videos, edit them with filters and creative tools, and combine multiple clips into one video.**
• Share multiple photos and videos (as many as you want!) to your story. Bring them to life with text and drawing tools. They disappear after 24 hours and appear on your profile grid or in feed.
• Watch stories from the people you follow in a bar at the top of your feed. View them at your own pace.
• Discover photos and videos you might like and follow new accounts in the Explore tab.
• Send private messages, photos, videos and posts from your feed directly to friends with Instagram Direct.
• Instantly share your posts to Facebook, Twitter, Tumblr and other social networks.

due to the existence of App marketplaces. For example, when we want to develop an App for social communication, there are hundreds of Apps within the same domain in Google Play. Meanwhile, because App descriptions are given by App providers, they contain denser domain knowledge than user feedbacks. Therefore, developers can gain the main feature information of Apps by analyzing their descriptions. Considering the description of an App 'Instagram' shown in Table 1, it contains the main features of the App, such as the common features 'discover accounts' and 'Post photos and videos', and the variable feature 'combine multiple clips into one video'. Similarly, more features with their attributes can be extracted from a huge number of related App descriptions. By using these information synthetically, the domain analysis can be performed to effectively support the development of a specific App. Hence, App descriptions, as a kind of abundant and useful data, cannot be ignored in the domain analysis of App products. However, due to their huge amount and lack of expression standard, intensive efforts are needed to analyze App descriptions.

To solve this problem, we propose an approach that automatically mines domain knowledge, including features and their relationships, from App descriptions. We define Concern-based Description Model (CDM) and Data-based Raw Domain Model (DRDM) to formally describe the knowledge in a single App description and overall knowledge in the domain separately. In addition, a quantified method is given to prioritize the knowledge for facilitating their application in practice. For the purpose of evaluation, we conducted a series of experiments with 574 App descriptions from Google Play.

Firstly, we conduct a quantitative evaluation of each method in our approach by comparing them with several well-established methods. The results show that our feature extraction method has a precision of 86.15% and a recall of 83.45% on average, which indicates the good performance of our method. Also, our topic modeling method can obtain more meaningful and understandable results than LDA, and our feature clustering method can achieve about 10% improvement over *K*-means in purity. These results verify that our approach can gain domain knowledge from App descriptions effectively.

Secondly, we give a case study and surveys to compare the usefulness of report generated in our approach with raw data of App descriptions and the feature model constructed by the approach proposed in (Hariri et al., 2013). By analyzing the results of the surveys with statistical methods, we find that the participants using our report can complete the tasks with significantly shorter time than the ones using the other two materials. Furthermore, we also find that our approach are more adaptive for overall analyzing the whole domain and generating creative ideas, which are important tasks in domain analysis.

The paper is organized as follows. Section 2 presents the related work. Section 3 gives problem statement in our research and an overview of our approach. In Section 4, the automatic proceeding of feature extracting and modeling from App descriptions is introduced. The integration and prioritization of domain knowledge are provided in V. Finally, our experiments and the conclusion are shown in Sections 6 and 7 respectively.

2. Related work

In early researches on domain analysis, the methods usually need to be done manually, such as Feature-Oriented Domain Analysis(FODA) (Kang et al., 1990), Feature-Oriented Reuse method (FORM) (Kang et al., 1998), and Organization Domain Modeling (ODM) (Coplien et al., 1998). Despite some tools (Benavides et al., 2007; Antkiewicz and Czarnecki, 2004; Sharma and Sharma, 2005) that have been proposed to support these methods, most of them can only help analysts to manage the process of feature extraction, whereas feature extraction itself and establishment of constraints still rely on intensive human interaction. This makes these methods only adapt to analyze relatively low volume of data.

With the abundance of data resources in recent years, the limitation of methods mainly depending on human analysis is obvious. Thus, techniques in data science, such as LDA, Clustering, N-gram, SVM, have been introduced to domain analysis to support the researches on automatic or semi-automatic methods. Based on the types of texts introduced in Section 1, we classify research works related to our work into three categories.

(1) To support the development process, the first type of texts is expressed normatively and records complete information of software, so we define it as the inputs with high quality. Many researches focus on utilizing this type of texts to construct Feature Models (FM) for Software Product Line Engineering (SPLE) (Bakar et al., 2015). Some of these researches take the formal texts as the input. For example, from tabular data files, Acher et al. propose a semi-automated method to extract FMs through a dedicated language and a specific merging algorithm (Acher et al., 2012). They further propose a tool-supported approach to extract and manage the evolution of software variability (Acher et al., 2014). Additionally, other researches analyze this type of texts expressed in natural language and extract features by introducing data analyzing techniques. From text-based software requirements specifications (SRSs), Mu et al. extract functional requirements by analyzing the linguistic characterization of SRSs (Mu et al., 2009); Bagheri et al. propose an approach that employs natural language processing techniques to identify potential features and integrity constraints in the domain document for support domain engineering lifecycle (Bagheri et al., 2012); in Niu et al. (2014), Niu Nan propose a systems-oriented approach based on information retrieval (IR) techniques to extract functional requirements profiles automatically; based on natural language semantics analyzing and classification technology, Mefteh et al. propose a fully top-down method to extract potential features and group similar ones for structuring the FM (Mefteh et al., 2016a); applying a sequence of machine learning steps, Rahimi et al. present a data mining approach for extracting and modeling quality

Please cite this article as: Y. Liu et al., Mining domain knowledge from app descriptions, The Journal of Systems and Software (2017), http://dx.doi.org/10.1016/j.jss.2017.08.024