The Journal of Systems and Software 86 (2013) 1294–1306

Contents lists available at SciVerse ScienceDirect

The Journal of Systems and Software

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

Software ecosystems – A systematic literature review

Konstantinos Manikas*, Klaus Marius Hansen

Department of Computer Science (DIKU), University of Copenhagen, Denmark

ARTICLE INFO

Article history:
Received 28 March 2012
Received in revised form 8 December 2012
Accepted 8 December 2012
Available online 20 December 2012

Keywords:
Software ecosystems
Software ecosystem
Systematic literature review

ABSTRACT

A software ecosystem is the interaction of a set of actors on top of a common technological platform that results in a number of software solutions or services. Arguably, software ecosystems are gaining importance with the advent of, e.g., the Google Android, Apache, and Salesforce.com ecosystems. However, there exists no systematic overview of the research done on software ecosystems from a software engineering perspective. We performed a systematic literature review of software ecosystem research, analyzing 90 papers on the subject taken from a gross collection of 420. Our main conclusions are that while research on software ecosystems is increasing (a) there is little consensus on what constitutes a software ecosystem, (b) few analytical models of software ecosystems exist, and (c) little research is done in the context of real-world ecosystems. This work provides an overview of the field, while identifying areas for future research.

1. Introduction

It has recently been suggested that *software ecosystems* (SECOs) are an effective way to construct large software systems on top of a software platform by composing components developed by actors both internal and external (Bosch, 2009; te Molder et al., 2011). In this setting, software engineering is spread outside the traditional borders of software companies to a group of companies, private persons, or other legal entities.

This differs from traditional outsourcing techniques in that the initiating actor does not necessarily own the software produced by contributing actors and does not hire the contributing actors. All actors, however, coexist in an interdependent way, an example being the iOS ecosystem in which Apple provides review of and a platform for selling applications in return for a yearly fee and 30% of revenues of application sale.[1] This is a parallel to natural ecosystems where the different members of the ecosystems (e.g., the plants, animals, or insects) are part of a food network where the existence of one species depends on the rest.

In addition to iOS, Google's Android ecosystem is a prominent example of a (smartphone) software ecosystem. Such ecosystems are arguably gaining importance commercially: it is, e.g., estimated that in 2012, more smartphones than personal computers will be sold.[2]

While software ecosystems are thus arguably gaining importance, research in software ecosystems is in its infancy, starting in 2005 with Messerschmitt and Szyperski (2005) and now with a dedicated workshop in its third year.[3] Our own literature search (see Section 3) revealed a gross list of 420 published papers on software ecosystems. However, until now there has been no systematic literature review (SLR) of the research literature on software ecosystems, leading to potential issues in identifying research gaps and contributions.

In the context of this, we have conducted a systematic literature review in the field of software ecosystems using the approach of Kitchenham and Charters (2007). As such, the purpose of this literature review is to provide an overview of the research reported in the field and identify possible issues that existing literature is not addressing adequately. This work is intended to function as a snapshot of the research in the field by (i) identifying and analyzing the different definitions of SECOs, (ii) analyzing the growth in research reported per year, (iii) classifying the research by type of result, (iv) defining and analyzing the software architecture and structure of SECOs, and (v) analyzing to which extent research is connected to SECO industry.

1.1. Article structure

The rest of this article is organized as following: in Section 2 we specify the review protocol, in Section 3 we document the extraction of the literature, in Section 4 we analyze the literature and answer the research questions, in Section 5 we list possible threats

* Corresponding author. Tel: +45 23839917.
E-mail addresses: kmanikas@diku.dk (K. Manikas), klausmh@diku.dk (K.M. Hansen).

[1] http://developer.apple.com/programs/ios/distribute.html.
[2] http://www.slideshare.net/CMSummit/ms-internet-trends060710final.
[3] http://www.softwareecosystems.org/workshop/.

0164-1212/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jss.2012.12.026

to the validity of this work and identify areas not covered from the literature and in Section 6 we conclude.

2. Review protocol

The applied review protocol is based on the guidelines of Kitchenham and Charters (2007). The establishment of the review protocol is necessary to ensure that the literature review is systematic and to minimize researcher bias. As such, the literature review is focused on a set of research questions that serve the aim of this work and derive from the reasons that initiated this review. The review protocol is organized in a way that the research questions define the main areas this study is focusing on. Section 2.2 defines the paper literature extraction strategy including the list of resource libraries, the search query and inclusion/exclusion criteria.

2.1. Research questions

The purpose of this systematic literature review is to provide an overview of the research reported in the field of SECO. In this overview, we intent to address the following research questions:

RQ 1: *How is the term 'software ecosystem' defined?*

In order to be able to analyze the field of SECOs, we should first define the SECO as object of study. Thus, the first objective of this work is to provide an overview of how the research community defines the term 'software ecosystem'. We achieve that by looking into the SECO definitions in the literature and comparing them. This will create an understanding of what the research community means by the term SECO.

RQ 2: *What is the research output per year in the SECO field?*

By grouping the literature per publication year we are able to identify possible trends in the research invested in the field of SECOs. An increase in the number of publications per year, for example, would imply the increase in importance of the field while a decrease in the number of publications might have as a possible reason the research in the field reaching a dead end. Analyzing the trends might give an idea of how the importance of the field of SECOs is changing with time.

RQ 3: *What is the type of result that software ecosystem research reports?*

After having defined the term SECO, a question that we want to address is what kind of research this field reports. Therefore, it is of interest to classify the papers according to the contribution they make. From a software engineering perspective, Shaw's classification of research results (Shaw, 2003) has been chosen. The classification contains the following categories:

Procedure or technique: This category includes papers that are providing a concrete and implementable way to solve a SECO problem. The solutions should be in the form of a procedure or technique that can be applied and not general rules of thumb or reported experiences. For example, Kazman et al. (2012) analyze a series of traditional software design and software architecture principles and methods in the perspective of the SECOs (or software-intensive ecosystems as they are called in the paper). This results in some new or adapted methods for the software design and architecture of these software-intensive ecosystems.

Qualitative or descriptive model: Papers using models based on qualitative analysis of data or well argumentation of existing cases. Papers in this category provide an analytical or descriptive model for the problem area. As an example the analysis of two different kinds of SECO: the "as-a-service" and "on-premise" software ecosystems that derived from a comparative study of two existing SECOs presented in Hilkert et al. (2010).

Empirical model: This category includes papers that use models derived from the quantitative data collection of the problem area. A paper of this category studies empirical data and concludes some analysis or predicting model. For example, Yu et al. (2008) extract information from open source systems to assess the evolvability of software.

Analytic model: Papers using models based on automatic or mathematical manipulation for solving a specific problem. For example the paper of Capuruço and Capretz (2010) that propose a prediction of recommendations and interaction between the members of a social ecosystem based on a mathematical analysis of the member relationships.

Tool or notation: A tool or notation created or implemented applying some method or technique. For example, a tool for recovering components and their relationships in free or open source projects, proposed by Lungu (2008)

Specific solution, prototype, answer, or judgment: Papers documenting a complete solution, evaluation of a theory or comparison of different theories based on a software engineering problem. The result is addressing a specific problem. An example would be Pettersson and Gil (2010) who address reusability and adaptability issues in mobile learning systems

Report: Papers documenting knowledge and experience obtained, rules of thumb or checklists but not systematic enough to be a descriptive model. For example, the analysis of the hybrid business and revenue models that software companies can have (Popp, 2011).

RQ 4: *What is the role of architecture in software ecosystem research?*

For single systems, software architecture is seen as important in determining the quality of a system being built (Bass et al., 2003; Hansen et al., 2011). In relation to this, we analyze the extent to which SECO literature stresses software architecture. We evaluate the literature in whether it is documenting any considerations towards SECO software architecture. In doing so, our concept of software architecture is in line with Bass et al. (2003):

> "The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them."

We here extend the definition to concern software ecosystems, i.e., we define 'software ecosystem architecture' as the structure or structures of the software ecosystem in terms of elements, the properties of these elements, and the relationships among these elements. The SECO elements can be systems, system components, and actors. Relationships then include software architecture-related relationships as well as actor-related relationships such the relationship between two actors.

RQ 5: *How is the connection between research and industry in the area of software ecosystems?*

It is of interest to know how close industry and research are in the field of software ecosystems. Research benefits from realism of problems when connected to the industry while industry arguably may become more innovative and efficient when connected to research. In the case of SECOs research results are more valid when they are concerning existing SECOs, while studies of problems in existing SECO can help the industry improve.