The Journal of Systems and Software 92 (2014) 128–142

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Architecture for embedded open software ecosystems

Ulrik Eklund[a,*], Jan Bosch[b]

[a] Volvo Car Corporation, SE-405 31 Gothenburg, Sweden
[b] Chalmers University of Technology, Software Engineering Division, Department of Computer Science & Engineering, SE-412 96 Göteborg, Sweden

ARTICLE INFO

Article history:
Received 4 December 2012
Received in revised form
13 December 2013
Accepted 4 January 2014
Available online 25 January 2014

Keywords:
Software architecture
Embedded software
Software ecosystem

ABSTRACT

Software is prevalent in embedded products and may be critical for the success of the products, but manufacturers may view software as a necessary evil rather than as a key strategic opportunity and business differentiator. One of the reasons for this can be extensive supplier and subcontractor relationships and the cost, effort or unpredictability of the deliverables from the subcontractors are experienced as a major problem.

The paper proposes open software ecosystem as an alternative approach to develop software for embedded systems, and elaborates on the necessary quality attributes of an embedded platform underlying such an ecosystem. The paper then defines a reference architecture consisting of 17 key decisions together with four architectural patterns, and provides the rationale why they are essential for an open software ecosystem platform for embedded systems in general and automotive systems in particular.

The reference architecture is validated through a prototypical platform implementation in an industrial setting, providing a deeper understanding of how the architecture could be realised in the automotive domain.

Four potential existing platforms, all targeted at the embedded domain (Android, OKL4, AUTOSAR and Robocop), are evaluated against the identified quality attributes to see how they could serve as a basis for an open software ecosystem platform with the conclusion that while none of them is a perfect fit they all have fundamental mechanisms necessary for an open software ecosystem approach.

1. Introduction

Software is prevalent in many embedded products; cars, washing machines, mobile phones, airplanes and satellites (Ebert and Jones, 2009). Typically these products are developed in large, and sometimes very complex, industrial projects where the embedded software may be critical for the success of the product but the manufacturing and delivery of the product may be a heavier investment than the software budget. This in turn tends to drive the entire R&D process, and software just follows the process logic of the mechanical development and manufacturing setup.

Original equipment manufacturers (OEM) of embedded products may view software in their products as a necessary evil rather than as a key strategic opportunity and business differentiator. One of the central reasons for this can be extensive supplier and subcontractor relationships and the cost, effort or unpredictability of the deliverables from external subcontractors are experienced as a major problem.

Ebert and Jones (2009) mention factors contributing to complexity in their survey of the present state of embedded software development: “combined software/hardware systems equipped with distributed software, computers, sensors, and actuators” which points to the integration aspects of these systems. They list “high demands on availability, safety, information security, and interoperability” as typical quality attributes.

Broy (2006) states about automotive software “The speed of the development, the complex requirements, the cost pressure and the insufficient competency in the field bring enormous challenges and risks”. We will hereafter use the automotive industry as a characteristic context of embedded systems since cars are arguably the most complex product with embedded software, both in terms of conflicting requirements and subcontractor relationships.

We thus define the domain of large industrial development of embedded systems in general, and the automotive domain in particular, by five characteristics:

- Deep integration between hardware and software for significant parts of the functionality.

* Corresponding author. Present address: Malmö University, School of Technology, Department of Computer Science, SE-205 06 Malmö, Sweden.
Tel.: +46 40 6657293.
E-mail addresses: ulrik.eklund@mah.se (U. Eklund), jan.bosch@chalmers.se (J. Bosch).

0164-1212/$ – see front matter
http://dx.doi.org/10.1016/j.jss.2014.01.009

• Strong focus on manufacturing aspects of the product in the development.
• Strong supplier involvement.
• Some parts realise safety-critical functionality.
• Long production life-time, i.e. mass-production.

But these mass-produced embedded systems (MPES) also exhibit some inherent problems (Eklund and Bosch, 2012): Heavy reliance on external developers and subcontractors complicates coordination through process. Outsourcing of significant parts of development to suppliers causes expensive communication and coordination delays during integration. Exponentially growing feature content severely complicates "big-bang" integration.

Bosch (2009) described how companies transition from software product lines to software ecosystems. This transition is driven by a need to deliver functionality to customers faster than what can be built in a reasonable amount of time by a single organisation, and with less R&D investment. He also stated that "extending the product (which includes the platform) with externally developed components or applications provide an effective mechanism for facilitating mass customisation."

Bosch and Bosch-Sijtsema (2010) presented five approaches to large-scale software development, ranging from integration-centric to composition-oriented in an open software ecosystem. The studied cases suggested the companies used a too integration-oriented approach.

We will investigate an open software ecosystem as a sustainable approach to develop software also for embedded systems, and in the rest of the paper we elaborate on the necessary properties of an embedded platform and design a reference architecture to facilitate a successful establishment and growth of ecosystems for embedded software. This is the second key activity proposed by Eklund and Bosch (2012) for establishing ecosystems for embedded software.

Software ecosystems is a new research area, Jansen et al. (2009) proposed a research agenda regarding ecosystems, and this paper is a response to the "software vendor challenge 3 – architecting for extensibility, portability, and variability" in that agenda. We have only found a single study addressing this research challenge for the embedded domain (Papatheocharous et al., 2013).

The research questions investigated are thus:

1 What are the key design decisions for an architecture in an open software ecosystem for mass-produced embedded systems?
 (a) What qualities must such an architecture satisfy?
 (b) What architectural patterns are suitable for a reference architecture?
 (c) Which existing platforms could serve as a basis for an ecosystem platform?

2. Research methodology and problem

This paper follows the design research methodology (DRM) proposed by Blessing and Chakrabarti (2009), with the four stages depicted in Fig. 1. The resulting artefact from the design research process being a reference architecture.

The research clarification stage of DRM, *C* in Fig. 1, explorers the context of embedded software development and identifies some problems in this context in Section 3. An open software ecosystem was identified as a viable solution mitigating some of the identified problems.

The first descriptive study stage, *DS I* in Fig. 1, identifies a set of quality scenarios based on what is required of a reference architecture and associated embedded platform for an ecosystem. These prerequisites are described in detail in Section 4.

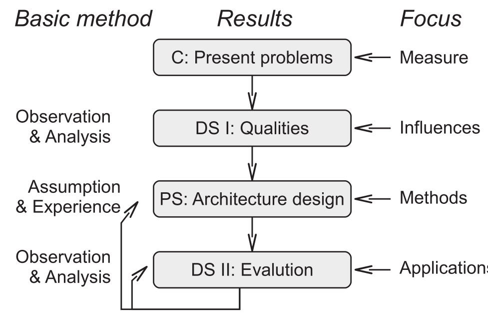

Fig. 1. The research process, applied from the design research methodology (DRM) framework by Blessing and Chakrabarti (2009).

The prescriptive study phase, *PS* in Fig. 1, defines the solution in terms of 17 architectural decisions together with a set of architecture patterns. The reference architecture and platform are described in detail in Section 5.

The second descriptive study stage, *DS II* in Fig. 1, evaluates the architecture through (I) observations in an industrial case study, (II) architectural analysis of four existing architecture frameworks against the scenarios identified in the first descriptive stage. This is described in Section 6.

2.1. Data collection and analysis

All data collected in the study was qualitative, and the evaluation in the second descriptive stage (DS II) was qualitative. Thus any scientific conclusions are also of a qualitative nature, i.e. answered that the design of the artefact (i.e. the reference architecture) works, not that it was optimal according to some quantitative measure. This type of qualitative assessment is common practice in industry (e.g. McGee et al. (2010), Ameller et al. (2012)).

The case data in Section 6.1 consist of working documents of both product owner and Scrum team triangulated with personal notes from the first author who was a participant/insider observer in the project at the time.

The artefacts were evaluated through observational, analytical and descriptive methods as categorised by Hevner et al. (2004):

• Observation in the case study in Section 6.1.
• Architectural analysis of designed artefacts based on available documentation in Sections 6.2 and 6.3.
• Informed argument based on insider knowledge to show the artefacts' utility in Section 6.1
• Scenarios describing the use of the artefact in the context, described in Section 3.2

3. The context of embedded software development

Today manufacturers of mass-produced embedded products range from focusing on efficient manufacturing of products with the embedded software as difficult necessity to seeing software as a key business differentiator. Many embedded domains, e.g. automotive, have extensive supplier and subcontractor relationships, often in many levels.

The most common approach to development of embedded software is to use an integration-centric approach according to the mapping study by Eklund and Bosch (2013). The study surveyed existing literature to identify approaches to embedded software development used in industry, excluding papers describing academic proof-of-concept prototypes. The study was performed similarly to a systematic literature review (Kitchenham and Charters, 2007), and resulted in 28 cases described in 23 papers from 2003 to 2012, of which the authors were involved in 4 papers