Information and Software Technology xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Information and Software Technology

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

Characteristics of software ecosystems for Federated Embedded Systems: A case study

Jakob Axelsson [a,b,*], Efi Papatheocharous [a], Jesper Andersson [c]

[a] Swedish Institute of Computer Science (SICS), SE-164 29 Kista, Sweden
[b] School of Innovation, Design and Engineering, Mälardalen University, SE-721 23 Västerås, Sweden
[c] Department of Computer Science, Linneaus University, SE-351 95 Växjö, Sweden

ARTICLE INFO

Article history:
Received 3 July 2013
Received in revised form 21 March 2014
Accepted 25 March 2014
Available online xxxx

Keywords:
Software ecosystems
Embedded Systems
Systems-of-Systems
Architecture
Case study

ABSTRACT

Context: Traditionally, Embedded Systems (ES) are tightly linked to physical products, and closed both for communication to the surrounding world and to additions or modifications by third parties. New technical solutions are however emerging that allow addition of plug-in software, as well as external communication for both software installation and data exchange. These mechanisms in combination will allow for the construction of Federated Embedded Systems (FES). Expected benefits include the possibility of third-party actors developing add-on functionality; a shorter time to market for new functions; and the ability to upgrade existing products in the field. This will however require not only new technical solutions, but also a transformation of the software ecosystems for ES.
Objective: This paper aims at providing an initial characterization of the mechanisms that need to be present to make a FES ecosystem successful. This includes identification of the actors, the possible business models, the effects on product development processes, methods and tools, as well as on the product architecture.
Method: The research was carried out as an explorative case study based on interviews with 15 senior staff members at 9 companies related to ES that represent different roles in a future ecosystem for FES. The interview data was analyzed and the findings were mapped according to the Business Model Canvas (BMC).
Results: The findings from the study describe the main characteristics of a FES ecosystem, and identify the challenges for future research and practice.
Conclusions: The case study indicates that new actors exist in the FES ecosystem compared to a traditional supply chain, and that their roles and relations are redefined. The business models include new revenue streams and services, but also create the need for trade-offs between, e.g., openness and dependability in the architecture, as well as new ways of working.

1. Introduction

In many industries developing technical products, such as automotive, aerospace, or process automation, Embedded Systems (ES) and software play an increasingly important role [1]. Traditionally, the ES has been an integral part of the physical product and being in charge of controlling and monitoring the product during its operation. Some of the key characteristics of many ES are that they have to be cost-efficient, reliable, robust, safe, and secure, and hence they are typically tailored for a certain product.

The software of the ES is usually monolithic in the sense that it is handled in the product as one piece, without the possibility to replace only parts of it, or add new parts on top of the present ones. The software is installed during production, and upgrades normally require physical connection to the ES. Often, the ES is developed by an external supplier based on a specification from the manufacturer responsible for integrating the final product.

With the arrival of affordable communication technologies, in particular wireless, it is becoming feasible to provide external communication capabilities to the ES. This eventually allows updates of software to be carried out remotely even though the flexibility of such upgrades would be limited due to the monolithic nature of the ES software. Recent research [2] has however demonstrated how a software plug-in mechanism can be utilized, that allows

* Corresponding author at: Swedish Institute of Computer Science (SICS), SE-164 29 Kista, Sweden. Tel.: +46 72 734 29 52.
E-mail address: jakob.axelsson@sics.se (J. Axelsson).

http://dx.doi.org/10.1016/j.infsof.2014.03.011
0950-5849/© 2014 The Authors. Published by Elsevier B.V.

the installation of add-on functionality in the ES and changes its monolithic nature. Such a plug-in mechanism, in combination with external communication, opens up many new possibilities for ES in particular and product development in general:

- It shortens dramatically the time to market for introducing new or extended features, and allows addition of new features into products that are already deployed, something which was not feasible before.
- It allows the creation of Systems-of-Systems (SoS), by letting plug-ins in different products connect to each other and exchange data through communication channels, thus forming a Federated Embedded System (FES), in ways that were not thought of at the time of design of the individual systems.
- It provides opportunities for third-party developers to supply add-ons, and thereby fosters open innovation formed by openness in collaborations in a way that was never seen before in the area of ES.

As an illustrative example of a FES, let us consider the case of car-to-car and car-to-infrastructure communication. In such a scenario, one could imagine a service where information is passed to cars about the recommended speed to be able to pass road intersections without having to stop at red lights. The cars and the servers in the road infrastructure form a federation to achieve this service. The recommended speed could be fed into the car's cruise control system to automatically adapt the car's speed. Thus, the service could decrease fuel consumption and travel time, and improve traffic flow. The information gathered by this service could further be used in real-time services that estimate the travel time, but also in aggregated services that show traffic patterns for route guidance, or even for long-term road network planning improvements. It is in this combination of direct improvements to the individual ES products and the creation of novel services that the real potential of FES resides.

These possibilities lead to many challenges on a technical level. In addition, they also have large implications on how the development process is instantiated and executed, and changes the relations between the different parties involved. The supply chain model traditionally used for ES, with a manufacturer integrating parts ordered from suppliers, will need to expand into a much more dynamic software ecosystem [3], where many new actors enter the ecosystem and interact. This perspective involves expansion beyond and across organizational boundaries, exposing platforms and opening access to reusable assets, increasing collaboration in the software ecosystem, but also involving the need of management of an economic, business and social environment [4].

The contribution of this paper is an empirical study of how such an ecosystem for FES should be set up and supported, resulting in a characterization of the business aspects, development issues, and architectural strategies. This ecosystem includes the ES with its hardware and built-in software; the software plug-ins that are added dynamically to the ES; the software running on servers that is needed to create services in the FES; and also to some extent, the mechanical products on which the ES are installed. Ecosystems in the ES domain are different than typical software ecosystems because the processes carried out need to support building multiple complex units of distributed functionality with real-time properties and constraints.

1.1. Research questions

The possibility of creating FES is currently emerging, and many companies in different industries are struggling with how to position themselves in this new ecosystem. Often, it will require the companies to make major revisions to their core business strategies, as well as their technology platforms, to remain competitive. The *goal* of this research is to provide support for such decisions by providing a more complete picture of how an ecosystem for FES should be organized. The underlying hypothesis of the research is:

H. The business related aspects are just as important as technology to the industry actors in order to make FES successful.

Therefore, two primary *Research Questions (RQs)* have been studied:

1. *What actors are needed in an ecosystem of Federated Embedded Systems (FES), and what are their relations?*
2. *What are the key elements of (a) the business models, (b) the product architecture, and (c) the processes, methods, and tools, needed to make the ecosystem effective?*

As a side effect of studying these RQs, we expected that various other relevant areas related to software ecosystems for FES would be discovered. Even though the development of FES technology is at a relatively early stage, we find it essential to have an early look at the non-technical aspects around it, since business models and development processes must be in place to make the technology effective. The way actors in the ecosystem interact will most likely also influence technology, by requiring different mechanisms to support development and operation of the system, and this research aims at discovering those mechanisms up front in order to ensure that they are properly reflected in future technical solutions.

1.2. Research method

Ecosystems for FES have only recently started to emerge, and therefore it is hard to find good examples of mature constellations to study empirically. Therefore, our research is highly explorative in its nature, and the best we can do to gain empirical data is to study similar ecosystems in related areas and try to extract and extrapolate information from there. The goal is thus to formulate a conceptual model which can be thought of as a set of hypotheses describing the important elements and relations in an ecosystem for FES. Naturally, these hypotheses have to be validated in future research.

There are several candidate research methods that could suit this kind of research, including case studies, grounded theory, and ethnographic studies. Grounded theory is useful when there is a need to generate new theories, but in this case there are already established theories for software systems, and it did not appear likely that the extension to FES would require a whole new set of concepts. As for ethnographic methods, it was difficult due to practical reasons. It would have required us to follow a certain ecosystem within it for some time, and we did not have access to a suitable object of study to allow this.

The research method we have chosen to apply is therefore an explorative case study, details of which can be found in [5]. In this research, the context is the relevant industry developing technical products and services in general. The focus is on the ecosystem as a whole, rather than on an individual company. The study includes a number of different companies, and is thus, according to the terminology of [5], an embedded case study with multiple units of analysis.

Since the study is not based on a pre-defined theory and it is performed for exploratory purposes, it is important to be open-minded and open-ended in the data collection process, and therefore a research method based on semi-structured interviews was considered suitable. In this way, an a priori idea of the most important

Please cite this article in press as: J. Axelsson et al., Characteristics of software ecosystems for Federated Embedded Systems: A case study, Inform. Softw. Technol. (2014), http://dx.doi.org/10.1016/j.infsof.2014.03.011