Introduction to the Software Product Lines and Platform Ecosystems: Engineering, Services, and Management Minitrack

Timo Käkölä
University of Jyväskylä, Finland
timokk66@gmail.com

Andrea Leitner
Virtual Vehicle Research Center, Austria
andrea.leitner@v2c2.at

Software has become the key asset for competitive products and services in all industries. Competitiveness in software development, maintenance, and services can be increased through (1) internal strategies such as the strategic creation and reuse of software platforms and (2) external strategies such as outsourcing software development, maintenance, and/or services from third party service providers and acquiring off-the-shelf components from providers and open source communities. A third strategy is to develop both strategies in parallel. This minitrack focuses on the first and third strategy.

Software product line engineering (SPL) is a methodology for developing software-intensive systems and services faster, at lower costs, and with better quality and higher end-user satisfaction. It differs from single system engineering, as:

1. It needs two development processes to work optimally: domain engineering and application engineering. Domain engineering defines and realizes the common and variable features of the product line by establishing and governing a common, relatively stable software platform. Application engineering derives applications by exploiting the commonality and binding variability built into the platform.
2. It needs to explicitly define and manage variability. During domain engineering, variability is introduced in all assets of the platform such as requirements, architectural models, components, and test cases. It is used during application engineering to mass-customize applications to the needs of customers.

The SPL body of knowledge has mainly been created by the software engineering community to enable industrialized software production. It covers fairly coherently and comprehensively issues such as

- methods and tools for software product line engineering and variability management and
- evolution of product line assets.

Other relevant bodies include but are not limited to economics, marketing, and industrial organization. Long-term product portfolio planning, organizational learning, and investments are typically needed to fully leverage the SPL strategy. The strategy tends to involve a geographically distributed ecosystem of stakeholders. The ecosystem needs to be orchestrated and the knowledge needs to be managed and shared amongst the stakeholders. When open source communities take responsibility for platform development and evolution, the leadership and governance of the ecosystem become increasingly distributed and emergent, making SPL management challenging. The products need to be serviced and service systems need to be built to increase customer satisfaction, generate new revenue streams, and ensure the feedback from the customers is effectively leveraged.

The minitrack is run for the 10th time in 2018. It has included app. 40 accepted papers and presentations over the years. After a rigorous review process, two papers were accepted for publication this time.

Tzeremes and Gomaa investigate the domain of smart spaces (e.g., smart homes) in which domain engineers (called EU SPL designers) develop product lines (called EU SPLs) from which end users (e.g., home owners) can mass-customize software applications for their own use. They contribute by presenting a systematic approach for EU SPL designers to develop EU SPLs and end users to derive software applications for their spaces, an end user development environment that supports EU SPL development and application derivation, and a testing approach for testing EU SPLs and derived applications.

End users typically create ambiguous and incomplete requirements descriptions in the fuzzy front end of software development. End users could benefit from software tools to alleviate such deficits but few suitable tools are available for them. Bäumer and Geierhos investigate this problem and contribute by developing a tool that helps end users to create unambiguous and complete requirements descriptions by combining existing expert tools and controlling them using automatic compensation strategies.

URI: http://hdl.handle.net/10125/50608
ISBN: 978-0-9981331-1-9