Dynamic Software Product Line (DSPL) engineering has proved itself as an efficient way to deal with run-time product adaptation. DSPLs have been successfully applied in domains such as smart homes, mobile devices or multimedia systems. However, existing DSPLs focus their efforts on highly adaptive products or on autonomic products. In this paper, we classify DSPLs according to their adaptation mechanisms and we also propose mixed DSPL which simultaneously address both adaptivity and autonomy. Finally, we discussed the underlying infrastructure to develop mixed DSPLs.

1 Introduction

A Software Product Line (SPL) engineering pursues the objective of producing a set of products that share a common set of assets in a specific domain. SPL engineering techniques allows to adapt a product to the customer needs decreasing its production costs and time to market. A product adaptation is performed during the SPL development, however further adaptations of a delivered product is hard to be managed.

Increasingly, software needs to dynamically adapt its behavior at run-time in response to changing conditions in the supporting computing infrastructure and in the surrounding physical environment [8]. Adaptive software is able to adapt its properties and resource requirements at run-time in response to dynamically varying user needs and resource constraints.

A Dynamic Software Product Line (DSPL) [4] is a SPL whose products are adaptative systems, i.e. a product might pro actively adapt itself when changes are performed in its environment.

When a product is produced in a SPL, a feature may be bound at different times: design time, compilation time, configuration time, runtime... Intensively using runtime binding, we would be able to produce reconfigurable products that may change their functionality even when they are deployed. However, the main difference between DSPL and SPL is the ability of changing the functionality of a product automedly and without the interaction of users. This grade of autonomy implies the usage of techniques that provide the knowledge or reasoning ability to adapt a product at runtime.

With the emergence of pervasive, mobile and service oriented computing, dynamics variation in users needs and available resources is becoming more and more widespread. For example, a pervasive system such as an smart home is highly dynamic since new kinds of entities (sensors, actuators, external software systems) can be installed in the system at any time. Furthermore, existing entities may fail or leave the system for a variety of reasons: hardware faults, OS errors, software bugs, network faults, etc. The dynamic of these systems makes SPL address the production of adaptive products.

In this paper we intend to summarize the DSPL architectures that have been proposed at date, dividing them in connected and disconnected DSPL depending on their connectivity and dependence with the DSPL development. We propose mixed DSPL as an intermediate solution that takes the benefits of both connected and disconnected DSPL architectures, giving some details about its internals.

This paper is structured as follows. In Section 2, the differences between SPL and DSPL development are remarked. Section 3 reviews the state of the art on DSPL architectures analyzing them using different aspects. In Section 4, we propose mixed DSPL as an intermediate alternative between connected and disconnected DSPL, whose architectural internals are discussed. Lastly, in Section 5, some conclusions and future work are briefly commented.