



52565

Uzma Afzal

41. Collaborative Configuration Management

Abstract

With the maturity of software development industries the old concept of product development is modified and brought product line approach in demand that shares common feature to satisfy the need of particular customers market. Because of the size of product line, its product derivation is a complex task requiring the involvement of many heterogeneous stakeholders. Taking their different roles and needs into account is essential to exploit the possible benefits of product lines.

Some issues related to this collaborative configuration are:

How to split configuration domain into configuration units

How to analyze the dependencies between different configuration units.

How to avoid conflicts that evolved at the time of compilation of smaller configuration units into a large configured product

The main focus of the research is conflicts [features\requirements\decisions] identification. Collaborative configuration management is intended to support the activities of projects that are configuring by multiple programmers. An unstable product configures if conflicts are not identified or identified in the late phases of software product configuration so an approach is required to capture these conflicts in earlier stage.

If these conflicts are not identified or late identified:

Release date of the product will be affected.

Cost of the product will increase.

Complexity of product increased.

Customer expectations and reliability on product can affect.

Moral of the development team will fall down

Many techniques and solutions are available to resolve the conflicts. In [17] a technique called CURE is described that diagnose errors and conflicts in configuration. CURE's diagnose specially recommend the minimum or cost optimal set of features that should be selected and

deselected in faulty configuration. The conversion from the faulty configuration to constraint satisfied product is complex and time consuming. In [19] an approach to collaborative product configuration that supports the splitting of the feature model into smaller units called configuration spaces and the arrangement of such spaces in a workflow-like plan is proposed it showed that, because CPC plans can contain errors that may cause invalid product specifications to be produced, validation rules are required to enforce the correctness of the plans. Even with visualization support from specialist tools such as pure::variants the visual representation of very complex model structures is a not a completely solved problem. Larger feature models can have several hundred features and the solution space can have several thousands or more constituents. Thus it can be hard to understand the implications of modifications to these models just through use of model diagrams [21].

To solve the problem we proposed a Layered based configuration repository (shared) architecture that reduces the configuration complexity by capturing conflicts (Requirements conflicts, features conflicts, decision conflicts) at earlier stage.

Some benefits of our solution are:

Reduced time-to-market.

reduced configuration complexity

Higher quality

Decreased cost

A “Product Configuration Tool” (PCT) is developed to support the proposed architecture. PCT has two views one for the population of configuration repository and other for the product derivation. Business pattern data of an ERP system is used to validate the repository architecture and its supportive tool.