DECIMAL and PLFaultCAT: From Product-Line Requirements to Product-Line Member Software Fault Trees*

Josh Dehlinger[1], Meredith Humphrey[1], Lada Suvorov[1], Prasanna Padmanabhan[2] and Robyn Lutz[1, 3]
[1]*Iowa State University*, [2]*Citrix Systems Inc.*, [3]*Jet Propulsion Laboratory/Caltech*
{dehlinge, mjh, suvorov}@iastate.edu, p.prasanna@mailcity.com and rlutz@cs.iastate.edu

Abstract

PLFaultCAT is a tool for software fault tree analysis (SFTA) during product-line engineering. When linked with DECIMAL, a product-line requirements verification tool, the enhanced version of PLFaultCAT provides traceability between product-line requirements and SFTA hazards as well as semi-automated derivation of the SFTA for each new product-line system previously verified by DECIMAL. The combined tool reduces the effort needed to safely reuse requirements and customize the product-line SFTA as each new system is constructed.

1. Introduction

Software product-line engineering supports reusability by developing a set of products sharing core commonalities and differing via a set of variabilities [3]. DECIMAL [2] and PLFaultCAT [1] provide management, traceability and automated verification of product-line requirements and the creation, derivation and analysis of product-line SFTAs, shown in Figure 1.

Using DECIMAL, requirements engineers can:

- Document product-line requirements
- Automatically verify consistency of a new product-line member with the product line's dependencies

Engineers can utilize PLFaultCAT to:

- Construct product-line SFTAs and
- Link the nodes of the SFTAs to the product-line requirements defined in DECIMAL
- Perform automated safety analyses on the product-line SFTAs to identify failure points, safety-critical requirements and new product-line constraints
- Automatically derive the set of product-line member SFTAs from a DECIMAL-verified product-line member

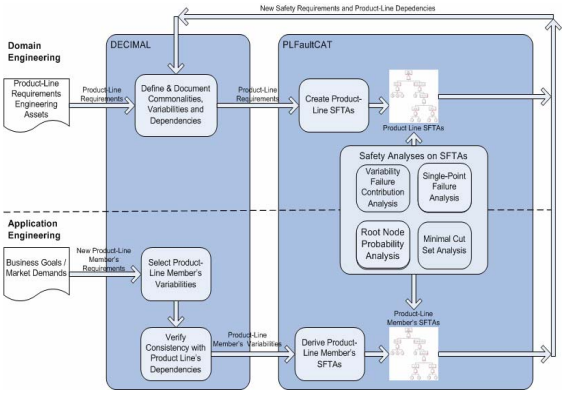

Figure 1. DECIMAL and PLFaultCAT's role in software product-line engineering

2. Domain Engineering

The domain engineering phase develops those software engineering assets relevant to the entire product line. The product line's requirements are documented and safety analysis assets are created for the entire product line. A product line must define its commonality, variability and dependency requirements. DECIMAL provides facilities to fully document and store a product line's requirements [2].

PLFaultCAT provides a graphical SFTA editor to construct a product-line fault tree [1]. PLFaultCAT reads the requirements of a product line from DECIMAL so that a user can associate a DECIMAL-defined requirement with the leaf nodes of a product-line SFTA, thus associating the product-line requirement(s) that can contribute to the causes of the hazard described in the leaf node. In addition to lessening the manual effort required to construct the product-line SFTA, this allows consistency and traceability between product-line requirements and

* This research was supported by the National Science Foundation under grants 0204139, 0205588 and 0541163.

those requirements that may cause hazards in the product-line SFTAs.

PLFaultCAT provides several automated safety analyses to identify failure points and safety-critical requirements. The *minimum-cut set analysis* identifies the smallest set of events that must occur such that the root node accident will occur. The *probability report* calculates the probability of occurrence of the root node given the probabilities of all other nodes. A *single-point failure analysis* searches the set of SFTAs for single-point failures at a user-specified depth. The *variability failure contribution analysis* finds those variabilities or combination of variabilities that contribute to a high number of SFTA hazards.

The single-point and variability failure contribution analyses, in particular, can aid in identifying latent safety requirements. For example, a single-point failure found in a product-line SFTA may necessitate new safety requirements to remove the single-point failure. Similarly, the variability failure contribution analysis may indicate variabilities that should not be allowed to be present in any product-line member.

3. Application Engineering

The application engineering phase exploits the reuse potential of product-line engineering by constructing a new product-line member from the assets produced in the domain engineering phase and deriving the relevant assets for the new member.

A product-line member is created by selecting its variabilities and defining the values of the variabilities. Verification must then show that the set of variabilities do not violate any of the defined dependencies. DECIMAL allows a user to select the variabilities for a new member and define the variability's values. For scalability, DECIMAL *automatically* verifies that the proposed new member's set of variabilities does not violate the defined dependencies.

The derivation of the set of SFTAs for a new member involves searching through the product-line SFTAs and pruning those hazard nodes that are not possible for the new member. The algorithm for this process is detailed in [1]. With multiple SFTAs and many nodes in each SFTA, this process is not practical in an industrial setting without such tool support.

PLFaultCAT can read in a verified product-line member from DECIMAL and *automatically* derive the member's SFTAs from the product-line SFTAs. To ensure safety, the tool uses a conservative approach to deriving the product line's SFTAs, so some user interaction is required in the derivation process. Further details and an evaluation can be found in [1].

4. Discussion

DECIMAL and PLFaultCAT address scalability issues in product-line member verification and SFTA derivation by automating these processes during the application engineering phase. Further, PLFaultCAT's automated identification of product-line SFTA failure points and safety-critical requirements improves the scalability of the approach over the manual effort needed without such tool support.

DECIMAL aids in the ease of reuse by providing users with the set of variabilities from which to select and then automatically verifying that the selected set does not violate any dependencies. PLFaultCAT promotes the reuse of product-line assets in three ways. First, it links SFTA hazards to the product-line requirements. Second, it uses the DECIMAL-verified product-line members as the driver for deriving the SFTAs for a new member. Finally, PLFaultCAT reuses the product-line SFTAs to derive the members' SFTAs.

DECIMAL addresses feature interactions among variabilities by providing an automated check of each new product-line member to verify that it complies with all required constraints and dependencies [2]. PLFaultCAT further addresses feature interactions by an analysis detailing which combinations of variabilities can contribute to hazards. The linking of DECIMAL product-line requirements and PLFaultCAT's SFTA hazard nodes allows traceability to product-line and product-line member SFTAs.

The tools also provide software engineers the ability to automatically derive valid product-line members' requirements and SFTAs. This allows software engineers to more cost-effectively explore the hazards defined by SFTAs at the requirements level for potential systems. The integrated tools provide a distinct perspective for developers into the safety consequences of proposed new combinations of features and can help identify those systems which are too high-risk to continue development.

5. References

[1] Dehlinger, J. and Lutz, R. R., "PLFaultCAT: A Product-Line Software Fault Tree Analysis Tool," *Automated Software Engineering Journal*, 13(1):169-193, 2006.

[2] Padmanabhan, P. and Lutz, R. R., "Tool-Supported Verification of Product-Line Requirements", *Automated Software Engineering Journal*, 12(4):447-485, 2005.

[3] Weiss, D. M. and Lai, C. T. R., *Software Product-Line Engineering*, Addison-Wesley, Reading, MA, 1999.