Paper summary of
"**Variability Modeling in the Real: A Perspective from the Operating System Domains**"
by Jason Christian

Variability Modeling is the first step of a generative software development, in which it produces features that meet the requirements as well as the dependencies between them. Variability modeling software becomes significant because it automates variability modeling such as consistency, detecting dead features, and useful graphical configuration tool.

This paper compares 2 variability modeling softwares: Kconfig, which is used to design the variabilities of Linux kernel, and CDL for embedded devices. Both have something in common in the usability of their softwares, which can be of importance for tool designers:

1. Both softwares can control the visibility of the features defined.
2. Both softwares support basic variables, such as boolean, integer, string, and other derived data types.
3. Both softwares make use of *Domain Specific Languages* (DSL) to model variability
4. *Reconfiguration*, a process of reconfigure the hierarchy of features based on the constraints given in a config file, is supported by both softwares.
5. Constraints which are used along with the reconfigurator

The key points that this paper has presented are:

1. The importance of variability modeling softwares
2. The outlines of similar features between 2 variability modeling softwares, which are used for designing features in operating systems. The similarities between those 2 large-scale industrial software will be a great contribution in designing better, more general variability modeling software
3. Overview of how variability modeling tools work as seen from the graphical figures and domain-specific codes.

The weaknesses of the paper are:

1. Comparison between 2 different variability modeling softwares for 2 different operating systems. Validation will be better if the author uses softwares which target or build similar systems
2. There are little explanations or definitions about the features of the modeling tools, such as constraints, configurators, feature groupings, and more.

Discussion Question: Is it possible to build a general, all-purpose variability modeling software using the important features outlined in the paper?