When building large software-intensive systems, engineers need to express and reason about at least two different types of choices. One type concerns 
uncertainty – choosing between different design alternatives [13], resolving inconsistencies [3], or resolving conflicting stakeholder requirements [9]. Another type deals with 
variability – supporting different variants of software that serve multiple customers or market segments [1].

Partial modeling has been proposed as a technique for managing uncertainty within a software model [7], [4]. A partial model explicates points of uncertainty and represents the set of possible models that could be obtained by making decisions and resolving the uncertainty. Methods for reasoning about the entire set of possibilities [4], transforming the entire set [5] and uncertainty-reducing refinements [10] have recently been developed.

Software product line engineering approaches propose techniques for managing the variability within sets of related software product variants [6], [8]. Such approaches explicate points of variability (a.k.a. features) and relationships between them in an artifact usually referred to as a feature model. A selection of features from this model guides the derivation of a specific product of a software product line (SPL). Techniques for reasoning about sets of SPL products [12], transforming the entire SPL [11] and supporting their partial configuration [2] have recently been developed.

Partial models and SPL representations are naturally quite similar – both provide ways of encoding and managing sets of artifacts. The techniques for representing, reasoning with and manipulating these sets, naturally, have much in common. Yet, the goals for creating these product sets are quite different, and thus the two techniques lead to distinct methodological considerations. Uncertainty is an aspect of the development process itself; it is transient and must be reduced and eventually eliminated as knowledge is gathered and decisions are made. Thus, the ultimate goal of resolving uncertainty is to produce only one desired artifact. On the other hand, variability is an aspect of the artifacts simultaneously managed through the entire development process; it is to be preserved and carefully engineered to represent the desired range of product variants required. Thus, product lines aim to produce and simultaneously manage multiple artifacts.

In this talk, I will survey approaches to representing, reasoning with and transforming models with uncertainty and variability, separately, as well as discuss current work on trying to combine the two approaches.

REFERENCES