

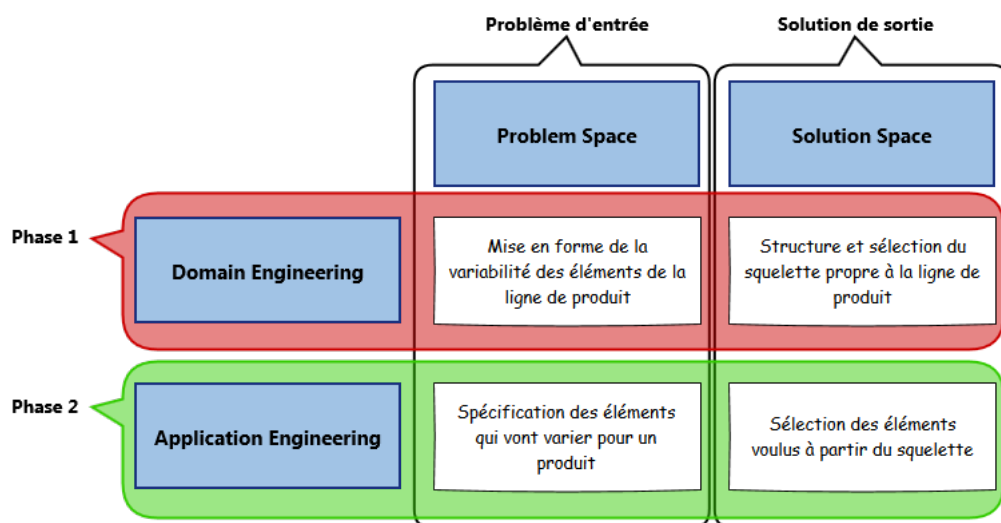
Résumé de l'article 3 : Software Product Line Engineering with Feature Models

L'article explique ce qu'est une ligne de produit et comment on peut l'appliquer aux différents projets de programmation. Ils utilisent un exemple concret ; Entrée et affichage de donnée météorologique sur un pc.

Il est expliqué dans cet article que le développement de logiciels "similaires" (En ligne de produit) prend le dessus sur les logiciels "uniques", puisque l'internationalisation fait que les logiciels doivent être adaptés à tout type de client, et de ce fait, développer un logiciel de zéro à un trop gros coût (budgétaire et temporel).

La SPLE (Software Product Line Engineering) sert à modéliser ce qui est commun et ce qui diffère entre les produits réalisés. Dans la SPLE, il y a une séparation logique entre le coeur du développement (la plateforme) et l'application. Les lignes de produits évoluent constamment, même après livraison d'un produit fini.

Ce schéma résume bien le principe de la SPLE. Dans la phase 1 (*Domain engineering*), on va prendre en entrée un domaine de problème (ici, celui des données météorologiques) et on va définir toutes les variabilités et points communs de ce domaine (les mesures, les sources des données, etc ...). En sortie, on aura un squelette représentant chaque élément du domaine, explicitant ceux qui varient et ceux qui sont communs. Ce squelette sera le même pour tous nos futurs produits. La phase 2 (*Application engineering*) s'applique à un produit spécifique, on va définir les éléments qui varient ou non (*Problem Space*), puis on va créer le squelette représentatif du produit souhaité. Cette deuxième phase est guidée par la première, on ne peut pas ajouter un élément qui ne se trouve pas dans le squelette générique. Le *Problem Space* peut être défini via les Features Models ou via des langages dédiés. Leur définition de Features est la suivante : Une Feature, dans le sens de notre exemple, est une caractéristique ayant un intérêt pour le client. Selon les souhaits que le client a vis à vis de la feature, cette dernière peut être une exigence, une fonction technique (ou groupe de fonction) ou une caractéristique "non-fonctionnel" (en terme de qualité).



Il est difficile de savoir quel feature représenter dans le problem space. L'exemple donné dans l'article est celui où il existe deux (ou plus) appareils pouvant nous donner des mesures. Ce problème peut se résoudre lors de la Solution Space, mais si l'appareil à un intérêt particulier auprès du client, il faudra alors inclure la possibilité de choisir l'appareil dans le feature model.

Pour utiliser une architecture variable, ils montrent l'utilisation d'un pug-in eclipse (Variant Management), c'est une idée à prendre en compte, mais au lieu de résumer l'exemple, je pense que si on regarde des tutoriels, ce sera mieux.