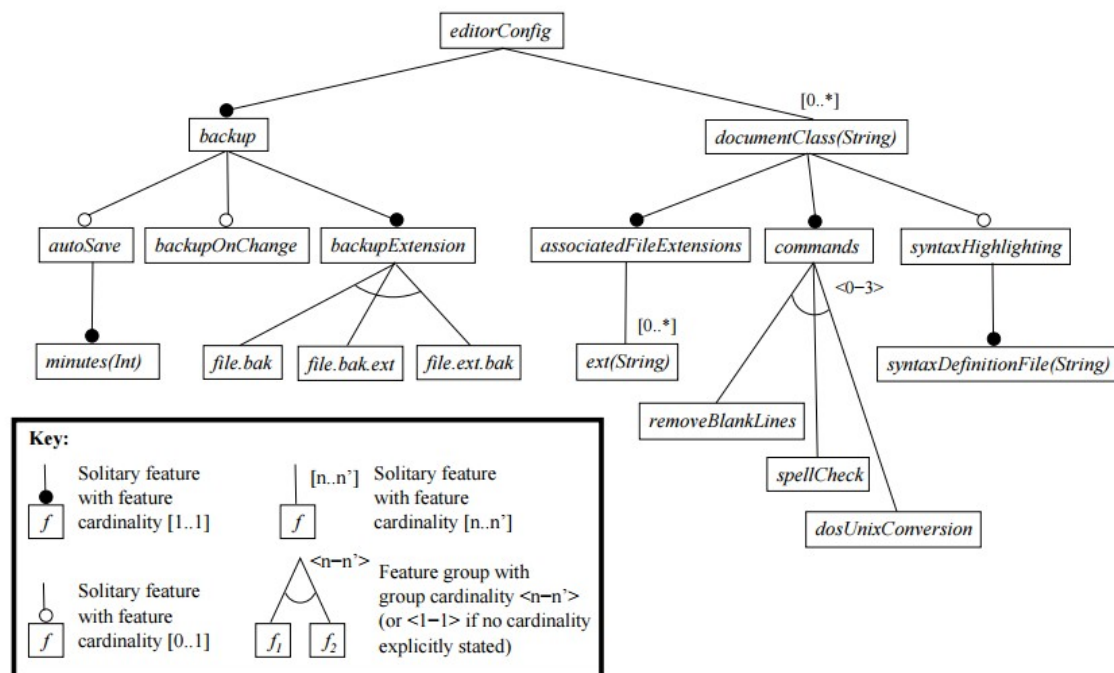


Les Features Models à base de cardinalités

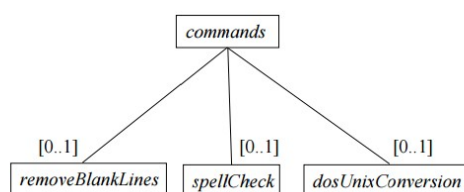
L'article définit les « *Features Models* » tel que des arbres de fonctionnalités dont la racine est un concept (par exemple un logiciel) dans lequel. On peut ajouter à ces fonctionnalités des informations telles qu'une description, un nombre ou une durée (les fonctionnalités peuvent avoir un attribut).



Voici un exemple de feature model.

Chaque branche de l'arbre possède une cardinalité, si elle n'est pas marquée elle est alors sous-entendue. Cette cardinalité permet de définir le nombre d'occurrences de la fonctionnalité pourront exister à la fois. De plus une fonctionnalité peut posséder plus d'un intervalle (ex. : une fonctionnalité qui possède les cardinalités [0..2][6..6] peut être présente 0, 1, 2 ou 6 fois).

L'ajout de cardinalités à un *Feature Model* engendre des représentations redondantes, principalement pour les représentations de groupes.



Ceci est une autre représentation possible de la *Feature* « commands » de l'arbre précédent.

Configuration vs Specialization

La configuration est le choix des features d'un Feature Model . La relation entre un Feature model et sa configuration est comparable à celle entre une classe et ses instances dans la programmation orientée objet.

La spécialisation est un processus de transformation qui part d'un premier diagramme pour en donner un autre, tel que le diagramme engendré possède un ensemble de configuration possible qui est un sous-ensemble du diagramme précédent. Un *feature model* totalement spécialisé ne possède qu'une seule configuration possible.

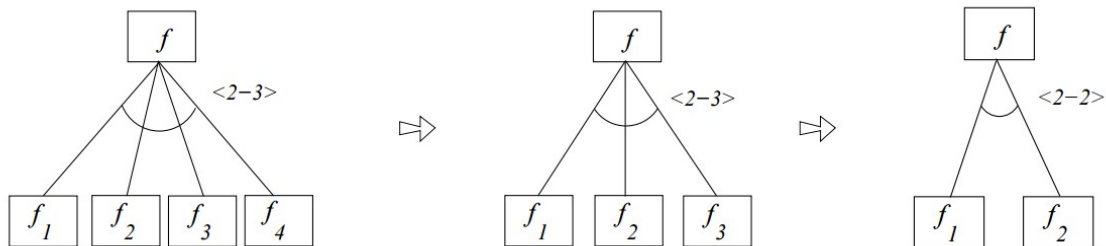
Les étapes de la spécialisation :

Il existe 6 étapes possibles pour la spécialisation.

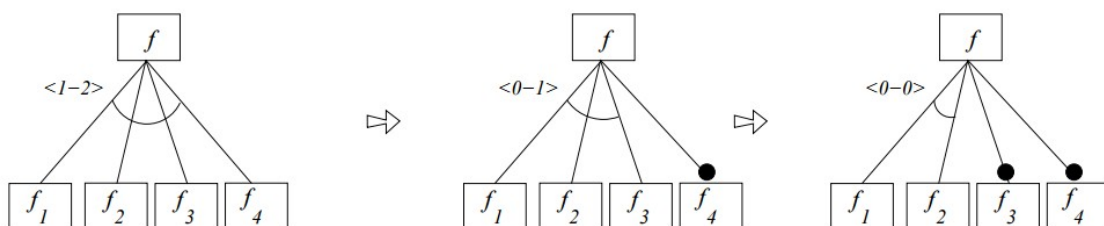
1/L'affinement des cardinalités : cela peut être fait en augmentant le minimum de l'intervalle ou réduire le maximum voir supprimer l'intervalle pour le fixer.

2/L'affinement des cardinalités de groupes de *features* : c'est une étape relativement identique à l'étape précédente, mais qui est effectué pour les groupes de fonctionnalités.

3/Retirer une ou plusieurs sous-fonctionnalités d'un groupe :



4/Sélectionner des fonctionnalités obligatoires d'un groupe :



5/Assigner une valeur à un attribut : cette étape correspond à donner une valeur aux attributs d'une fonctionnalité non initialisée. Sa valeur devra être évidemment du type de l'attribut.

6/Cloner une sous-fonctionnalité unique (qui n'appartient pas à un groupe) :

Cette opération permet de cloner une sous-fonctionnalité unique. Le clone pourra avoir une cardinalité arbitraire, mais fixe, tant que cela n'agrandit pas l'intervalle de la cardinalité de la sous-fonctionnalité d'origine.

Exemple :



Ici le clonage de la sous-fonctionnalité est possible, car les cardinalités $[3..3]$ et $[1..*]$ obligent qu'il y en ait au moins 4. C'est donc bien une spécialisation, car il n'est plus possible d'avoir seulement 2 fois f_0 .

En suivant ces étapes, on peut donc spécialiser un *feature model*. Voici un exemple de spécialisation du *feature model* présenté plus haut :

