HAL open science

Approche collaborative pour la configuration des lignes de produits

Sabrine Edded

To cite this version:

Sabrine Edded. Approche collaborative pour la configuration des lignes de produits. Environnements Informatiques pour l'Apprentissage Humain. Université Panthéon-Sorbonne - Paris I; Université de la Manouba (Tunisie), 2021. Français. NNT : 2021PA01E034 . tel-03608153

HAL Id: tel-03608153

https://theses.hal.science/tel-03608153v1

Submitted on 14 Mar 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Thèse de doctorat

pour obtenir le grade de docteur en informatique délivré par

École Nationale des Sciences de l'Informatique
Et
Université Paris 1 Panthéon Sorbonne

Présentée et soutenue publiquement par

Sabrine Edded

le 29/juin/2021

Approche Collaborative pour la Configuration des Lignes de Produits

Devant le jury composé de :

Pr. Henda BEN GHEZALA,	Directeur,	ENSI, Université de la Manouba, Tunisie
Pr. Camille SALINESI,	Directeur,	Université Paris 1 Panthéon-Sorbonne, France
Dr. Sihem BEN SASSI,	Co-directeur,	ENSI, Université de la Manouba, Tunisie
Pr. Raùl MAZO,	Co-directeur,	ENSTA Bretagne, France
Pr. Naoufel KRAIEM,	Président,	ISI, Université de Tunis El Manar, Tunisie
Pr. Agnès FRONT,	Rapporteur,	Université Grenoble Alpes, France
Pr. Youcef BAGHDADI,	Rapporteur,	Université Sultan Qaboos, Oman

Résumé

Dans le contexte de l'ingénierie des lignes de produits, la configuration collaborative est un processus décisionnel durant lequel plusieurs parties prenantes décident la spécification d'un produit unique. Ce processus devient compliqué quand les décisions de configuration des parties prenantes se contredisent et/ou ne respectent pas les contraintes du domaine de la ligne des produits ce qui peut conduire à des situations conflictuelles.

Garantir un compromis entre les décisions de configuration de toutes les parties prenantes pour résoudre ces situations de conflit constitue un enjeu crucial. De plus, la résolution de ces situations ne peut pas être considérée comme efficace si elle ne prend pas en compte les décisions de l'ensemble des parties prenantes.
Par conséquent, il est crucial d'opter pour une stratégie de résolution qui tient compte des préférences des parties prenantes.

Cette thèse présente une revue de la littérature sur les approches de configuration collaborative des lignes de produits. Elle montre l'importance de considération des préférences dans le processus de résolution des conflits. L'approche Colla-Config proposée dans cette thèse, permet la configuration collaborative des lignes de produits de manière flexible. Elle propose une méthode de résolution des conflits basée sur les préférences des parties prenante. L'idée clé est de permettre aux parties prenantes d'exprimer leurs préférences sous forme d'un ensemble de règles de substitution et d'utiliser ces dernières pour déterminer l'ensemble minimal de choix à éliminer pour résoudre les conflits. Ces ensembles sont identifiés en utilisant l'algorithme de calcul des sous-ensembles de correction minimaux (Minimal Correction Subsets (MCS)).

Mots-clés : lignes de produits, configuration collaborative, résolution des conflits, préférences, sous-ensemble de correction minimale.

Abstract

In the context of software product line engineering, collaborative configuration is a decision-making process where multiple stakeholders contribute in building a single product specification. This process gets complicated when the configuration decisions of involved stakeholders are contradictory, which may lead to conflicting situations.

Ensuring a compromise between all stakeholders configuration decisions during conflict resolution is a crucial issue. Moreover, conflict resolution cannot be considered effective if it does not take into account the decisions of all stakeholders. Therefore, it is important to follow a resolution strategy that takes into account stakeholders' preferences and fairly considers their decisions.

This thesis presents a literature review on product lines collaborative configuration approaches. It shows how important to consider preferences in the conflict resolution process.

The thesis proposes a new product lines collaborative configuration approach called Colla-Config. This approach, allows the collaborative configuration of product lines in a flexible way. It proposes a conflict resolution method based on stakeholders' preferences. The key idea is to allow stakeholders to express their preferences within a set of substitution rules. Therefore, these rules are used to identify the minimal set of configuration choices to be eliminated to resolve conflicts. These sets are computed using the Minimal Correction Subsets (MCS) algorithm.

Keywords: product lines, collaborative configuration, conflict resolution, preferences, minimal correction subsets.

Remerciements

Je tiens à remercier mes directeurs de thèse, Madame Henda BEN GHEZALA, Professeur à l'École Nationale des Sciences de l'Informatique et Monsieur Camille SALINESI, Professeur à l'Université Paris 1 Panthéon-Sorbonne, qui m'ont encadré tout au long de cette thèse. Je leur suis reconnaissante de m'avoir fait bénéficier de leurs compétences et de leurs pertinents conseils. Je les remercie également pour leurs qualités humaines d'écoute et compréhension tout au long de ce travail de recherche.

Je remercie Madame Sihem BEN SASSI, Maître assistante à l'Institut Supérieur des Études Technologiques en Communications de Tunis et Monsieur Raùl MAZO, Professeur à ENSTA Bretagne, pour leur encadrement, leur disponibilité et leur aide précieuse de tous les jours.

Je tiens à exprimer ma gratitude aux membres du jury de l'honneur qu'ils m'ont fait en acceptant de juger mon travail. Je remercie tout d'abord Monsieur Naoufel KRAIEM, Professeur à l'Institut Supérieur d'Informatique, d'avoir accepté de présider ce jury de thèse. J'adresse tous mes remerciements à Madame Agnès FRONT, Professeur à l'Université Grenoble Alpes et Monsieur Youcef BAGHDADI, Professeur à l'Université Sultan Qaboos, d'avoir accepté de rapporter ce manuscrit.

J'adresse également mes remerciements à tous les membres du Centre de Recherche en Informatique (CRI) et en particulier à l'ensemble des thésards pour leur amitié et leur aide.

J'exprime ma gratitude à mes chers parents Abdelkader et Saadia pour leur soutien lors des moments difficiles durant ces années. Merci pour avoir fait de moi celle que je suis aujourd'hui. Merci à ma Sœur Torkia et mes frères Mohamed, Imed et Ismail pour leur amour et leurs encouragements.

Table des matières

I Introduction **8**
I.1 Contexte 9
I.2 Problématique 10
I.3 Questions de recherche 11
I.4 Méthodologie de recherche 12
I.5 Contributions 14
I.6 Organisation de la thèse 15

II La configuration des lignes de produits **17**
II.1 Introduction 18
II.2 Concepts de base 18
II.2.1 L'ingénierie des lignes de produits 18
II.2.2 Définitions 20
II.3 Types de configuration 21
II.3.1 Configuration totale 22
II.3.1.1 Approche générative 22
II.3.1.2 Approche a priori 22
II.3.1.3 Approche de complétude 22
II.3.2 Configuration progressive 23
II.3.2.1 Configuration partielle 23
II.3.2.2 Configuration par étapes 23
II.4 Les approches de configuration 24
II.4.1 La configuration non guidée 24
II.4.2 La configuration dirigée par les modèles 25
II.4.3 La configuration dirigée par la logique 25
II.4.3.1 La configuration dirigée par la logique propositionnelle 25
II.4.3.2 La configuration dirigée par la logique de description 26

II.4.3.3 La configuration dirigée par la programmation par contraintes . . . 26
II.4.4 La configuration basée sur les algorithmes ad-hoc . . . 27
II.5 Conclusion . . . 27

III La configuration collaborative des lignes de produits 29
III.1 Introduction . . . 30
III.2 Processus d'élaboration du cadre de référence proposé . . . 30
III.2.1 Recherche manuelle . . . 31
III.2.2 Identification des termes et chaines de recherche . . . 32
III.2.3 Recherche automatisée . . . 33
III.2.4 Sélection des études . . . 34
III.2.5 Processus d'extraction de mots-clés . . . 36
III.2.6 Construction et validation du cadre de référence . . . 37
III.3 Présentation du cadre de référence PL2C . . . 37
III.3.1 La dimension But . . . 39
III.3.1.1 Principe . . . 39
III.3.1.2 Scénario de collaboration . . . 39
III.3.1.3 Satisfaction des parties prenantes . . . 40
III.3.2 La dimension Collaboration . . . 41
III.3.2.1 Niveau de collaboration . . . 41
III.3.2.2 Mode de collaboration . . . 41
III.3.2.3 Interaction . . . 42
III.3.3 La dimension Processus de configuration collaborative . . . 42
III.3.3.1 Types de configuration . . . 42
III.3.3.2 Niveau de configuration . . . 43
III.3.3.3 Ordre de sélection . . . 43
III.3.3.4 Guidage . . . 44
III.3.3.5 Attribution des rôles . . . 44
III.3.3.6 Gestion des conflits . . . 44
III.3.3.7 Aide à la décision . . . 46
III.3.4 La dimension Outil . . . 46
III.4 Synthèse . . . 47
III.5 Les défis de la configuration collaborative . . . 48
III.6 Conclusion . . . 50

IV Aperçu de l'approche Colla-Config 51
IV.1 Introduction . . . 52
IV.2 Principe de la configuration collaborative . . . 52

IV.2.1 Cadre méthodologique de la configuration collaborative 52
IV.2.2 Mise en situation 53
IV.3 Vue globale de l'approche Colla-Config 54
IV.3.1 La configuration 55
IV.3.2 La vérification de la configuration 56
IV.3.2.1 La notion de conflit 56
IV.3.2.2 Le principe de vérification 58
IV.3.3 La résolution des conflits 59
IV.3.3.1 Le sous-ensemble de correction minimal (MCS) . . . 59
IV.3.3.2 Le principe de résolution 59
IV.4 Exemple d'application illustratif 60
IV.4.1 La configuration de la ligne des portails web 60
IV.4.2 La vérification de la configuration 62
IV.4.3 La résolution des conflits 63
IV.5 Caractéristiques de l'approche Colla-Config 64
IV.5.1 Flexibilité du processus de configuration 64
IV.5.2 Résolution des inconsistances 65
IV.5.3 Considération de satisfaction des parties prenantes 65
IV.6 Conclusion 66

V Processus de configuration de Colla-Config 67
V.1 Introduction 68
V.2 Préambule sur le formalisme CSP 68
V.2.1 Définition 68
V.2.2 Problème de satisfiabilité 68
V.3 Algorithme de calcul des MCSs 69
V.4 Processus de configuration Colla-Config 70
V.4.1 Formalisation de la configuration 71
V.4.2 Identification et résolution des conflits 73
V.4.3 Les règles de substitution 73
V.4.4 Processus de résolution des conflits 74
V.4.4.1 Suppression de la partie commune des choix de configuration 74
V.4.4.2 Calcul de liste des MCSs 76
V.4.4.3 Application des règles de substitution sur liste des MCSs 79
V.5 Conclusion 82

VI Evaluation **84**
VI.1 Introduction 85
VI.2 Présentation de l'outil Colla-Config 85
VI.3 Principe d'expérimentation 86
VI.4 Question de recherche 87
VI.5 Protocole expérimental pour l'évaluation de l'utilisabilité 87
VI.5.1 Processus d'évaluation 88
VI.5.2 Les métriques d'utilisabilité 89
VI.6 Expérimentation 90
VI.6.1 Expérimentation partie 1 91
VI.6.2 Expérimentation partie 2 95
VI.7 Résultats 97
VI.7.1 Résultats de performance 97
VI.7.2 Résultats de satisfaction 99
VI.8 Les menaces à la validité 100
VI.8.1 La validité de la conclusion 101
VI.8.2 La validité interne 101
VI.8.3 La validité conceptuelle 102
VI.8.4 La validité externe 102
VI.9 Conclusion 103

VII Conclusion **104**
VII.1 Contributions 105
VII.2 Perspectives 106
VII.2.1 Outillage 106
VII.2.2 Extension de l'approche Colla-Config 106
VII.2.3 Évaluation 107
VII.2.4 Intégration aux approches existantes 107
VII.2.5 Intégration à des outils existants 108

Bibliographie **109**
.1 Liste des études sélectionnées 2005-2019 118

Table des figures

I.1 Modèle du processus de recherche conceptuelle (Johannesson and Perjons, 2014) appliqué à notre recherche. 13

II.1 L'ingénierie des lignes de produits (Pohl et al., 2005) 18
II.2 Arborescence des types et approches de configuration des LPs 28

III.1 Processus d'élaboration du cadre PL2C 31
III.2 Résultat de regroupement des termes à haute fréquence 33
III.3 Les étapes du processus de sélection des documents. 35
III.4 Les quatre vues du PL2C 37

IV.1 Processus de configuration collaborative d'une LP. 53
IV.2 Illustration de la configuration collaborative d'une LP. 54
IV.3 L'approche Colla-Config 55
IV.4 Le modèle de caractéristiques des portails web (Mendonca et al., 2008). 61

V.1 Processus de configuration Colla-Config 71
V.2 Représentation d'une configuration 71
V.3 Modèle réduit de la ligne des portails web(Mendonca et al., 2008) . . 72
V.4 Exemple de représentation d'une configuration 72
V.5 Processus de résolution des conflits 75
V.6 Structure d'une configuration initiale 75
V.7 Schéma d'application des règles sur la liste des MCSs 80

VI.1 Architecture globale de Colla-Config. 85
VI.2 Le processus d'expérimentation. 86
VI.3 Processus de test d'utilisabilité. 88
VI.4 Interface d'enregistrement. 92
VI.5 Interface de configuration. 92
VI.6 Interface de création du modèle des portails web. 93

VI.7 Interface de fusion des choix des participants. 93
VI.8 Interface de synthèse des choix des participants. 94
VI.9 Interface de résolution des conflits. 94
VI.10Interface de résultat de résolution. 94
VI.11Interface de configuration finale. 95
VI.12Décomposition du modèle des portails web (Mendonca et al., 2008). . 96
VI.13Plan de configuration(Mendonca et al., 2008). 96
VI.14Temps moyen pour accomplir chaque tâche par les participants. . . . 99
VI.15Résultats des questions de satisfaction. 100

Liste des tableaux

III.1 Groupe des termes. 33
III.2 Chaînes de recherche dérivées. 34
III.3 Résultats de la recherche automatisée. 34
III.4 Liste des critères d'inclusion et d'exclusion. 35
III.5 Critères d'évaluation de la qualité. 35

IV.1 Types d'inconsistance 57
IV.2 La sélection des règles 62
IV.3 Les choix de configuration 63
IV.4 La liste des conflits 63
IV.5 La résolution des conflits 64

V.1 Liste des MUSs pour la formule φ 70
V.2 Liste des règles de substitution et le MCS correspondant 74
V.3 Exemple de configuration 75
V.4 Liste des conflits identifiés 78
V.5 Liste initiale des MCSs calculés 78
V.6 Liste finale des MCSs calculés 79
V.7 Résultat d'application des règles 82

VI.1 Résultats de performance des participants 97
VI.2 Durée des tâches en minutes pour le chef de produit 97
VI.3 Durée des tâches en minutes pour les participants 98
VI.4 Résumé de résultats de performance des 10 participants 99
VI.5 Scores moyens de satisfaction des participants 100

I

Introduction

Sommaire

I.1 Contexte **9**

I.2 Problématique **10**

I.3 Questions de recherche **11**

I.4 Méthodologie de recherche **12**

I.5 Contributions **14**

I.6 Organisation de la thèse **15**

I.1 Contexte

De nos jours, le domaine du génie logiciel a connu une évolution exponentielle afin de couvrir les nouvelles problématiques introduites par le croisement de l'informatique avec l'industrie. Ce croisement donne naissance à des systèmes informatiques industriels hautement variés et complexes, ce qui entraine de sérieux défis en terme de production tels que la variété de choix, les contraintes budgétaires et des échéanciers de réalisation de plus en plus serrés. Face à ces défis, il est indispensable d'envisager une stratégie de production qui permet de fournir des produits personnalisés tout en gardant des faibles coûts de production. L'ingénierie des Lignes de Produits (LP) (Pohl et al., 2005) cherche à répondre à cette exigence. Il s'agit d'un paradigme qui prône une vision de modélisation et de développement dans laquelle l'objectif envisagé n'est plus l'obtention d'un système, mais d'un ensemble de systèmes possédant des caractéristiques communes (commonalités) et se distinguent les uns des autres par des caractéristiques dites optionnelles (variabilité).
Le concept de variabilité représente le pilier de l'ingénierie des lignes de produits qui repose principalement sur deux niveaux : l'ingénierie du domaine et l'ingénierie d'application. L'ingénierie du domaine consiste à identifier les fonctionnalités présentes dans la ligne de produits et spécifier les artéfacts réutilisables en définissant les éléments communs et variables de la LP. L'ingénierie d'application consiste à construire un produit bien particulier en utilisant les artéfacts développés dans l'ingénierie du domaine tout en précisant les éléments de variabilité et en respectant les relations d'interdépendances entre les éléments de la LP. Ces éléments sont communément représentés par un modèle de caractéristiques (Kang et al., 1990).
L'ensemble des activités qui consistent à spécifier un produit valide à partir d'un modèle de LP conformément aux exigences des utilisateurs constitue le processus de configuration (Deelstra et al., 2005). La configuration consiste à choisir, à partir d'un modèle de LP, un ensemble de caractéristiques qui doivent être à la fois conformes aux contraintes du modèle de la LP et aux exigences de l'utilisateur (Clements and Northrop, 2001).
Lorsque le modèle de la LP est grand, le nombre de configurations possibles peut être important également. La configuration de ce type de modèles peut être une activité difficile pour les praticiens et nécessite souvent l'intervention de plusieurs utilisateurs. Dans ce cas, le processus de configuration peut être qualifié de configuration collaborative des LPs (Mendonca et al., 2008).

I.2 Problématique

À l'échelle industrielle, on ne peut pas s'attendre que les activités de configuration soient gérées par un seul utilisateur responsable de toutes les décisions de configuration (Mendonca et al., 2008). Cela est non seulement dû au nombre de décisions à prendre, mais aussi à la nature multidisciplinaire des modèles de LP (ingénierie, marketing, etc.). Dans ce cas, le partage des décisions de configuration entre les parties prenantes est indispensable pour faire face à ces problèmes de configuration.

Les modèles de lignes de produits réels peuvent contenir jusqu'à 10 000 caractéristiques (Batory et al., 2006). Dans un tel cas, la configuration devient une tâche ardue et source d' erreurs. Garantir un compromis entre les décisions de configuration de toutes les parties prenantes lors de la résolution d'un conflit constitue un enjeu crucial. Potentiellement, plus il y a de contraintes, plus l'effort est important pour résoudre les conflits. De plus, la résolution de ces situations ne peut pas être considérée comme efficace si elle ne prend pas en compte les décisions de configuration de l'ensemble des parties prenantes. Pour faire face aux problèmes de résolution de conflits, plusieurs propositions telles que celles de (Czarnecki et al., 2005), (Mendonca et al., 2007), (Mendonca et al., 2008), (Rabiser et al., 2009) et (Hubaux. et al., 2010) ont adopté un principe de configuration progressive où les décisions de configuration sont raffinées en plusieurs étapes par différentes parties prenantes et organisées par un workflow. L'approche basée sur le workflow présente un inconvénient important : le processus de configuration est rigide et manque alors de flexibilité. Cette méthode pose plusieurs obstacles à la négociation interactive des différentes décisions et, dans de nombreux cas, les choix faits lors des étapes précédentes se superposent à ceux des phases ultérieures. En outre, les activités de certaines parties prenantes auraient des effets secondaires sur celles d'autres parties prenantes, parfois par une propagation qui peut être difficile à anticiper et à résoudre au préalable. Le manque de flexibilité qui en résulte risque d'être une limitation dans la résolution des conflits car la stratégie de résolution des conflits adoptée ne garantit pas la satisfaction de toutes les parties prenantes et ne tient pas compte de leurs préférences.

Le grand défi consiste donc à définir des approches qui offrent suffisamment de flexibilité durant le processus de configuration collaborative tout en garantissant une résolution des conflits qui tient compte des préférences des parties prenantes.

I.3 Questions de recherche

La contribution principale de cette thèse réside en une technique de résolution qui combine les préférences et les exigences des acteurs pour garantir leur satisfaction en cas de conflit au sein d'une approche collaborative permettant à plusieurs acteurs de configurer un produit de manière flexible sans aucun ordre prédéfini ni restriction sur les parties du modèle à configurer. Dans le contexte de cette thèse, un acteur est toute personne désirant configurer un produit à partir d'une LP, pouvant être un client désirant acheter un produit sur un site commerçant ou bien des experts/ingénieurs du domaine ou tout acteur de l'entreprise à laquelle appartient la LP en question. La principale question de recherche est :

QR_Principale *Comment permettre à plusieurs acteurs de configurer une ligne de produits de manière collaborative ?*

Afin d'apporter les éléments de réponse nécessaires à cette question, il convient de traiter les deux questions de recherche suivantes.

QR_1 *Quelles sont les différentes situations de conflit qui peuvent être rencontrées durant la configuration collaborative et comment peut-on les définir ?*

Dans le contexte de configuration collaborative des LPs, la gestion des conflits est d'une importance primordiale. D'après (Mendonca et al., 2008), les conflits peuvent être détectés quand deux ou plusieurs caractéristiques ont des dépendances explicites ou implicites entre eux. (Osman et al., 2009) souligne qu'un conflit se produit lorsque deux ou plusieurs décisions de configuration attribuées à différents acteurs de configuration ne peuvent pas être vraies en même temps.

En pratique, la configuration collaborative pose de sérieux problèmes. Notamment il faut coordonner les activités de configuration et assurer un compromis entre les décisions hétérogènes des différents acteurs.

Le problème de la coordination présente un défi encore plus grand pour la configuration des modèles de grandes tailles et de réseaux complexes de contraintes. Plus le nombre d'éléments à configurer est élevé, plus l'effort requis pour mettre en place des stratégies efficaces de résolution des conflits peut s'avérer important. En d'autres termes, la question de la coordination a un impact sur la nature des conflits ainsi que sur leur gestion. La coordination peut être envisagée de deux manières différentes :

1. En suivant un processus de configuration prédéfini où le temps d'intervention de chaque acteur est précisé à priori. Une telle méthode rend la résolution des conflits rigide et ne considère pas les choix de tous les acteurs car souvent les choix faits au cours des premières étapes se superposent à ceux des phases ultérieures.

2. En autorisant un processus de configuration libre où les acteurs expriment leurs choix séparément et sans aucun ordre de passage. Ici la gestion des conflits

devient plus compliquée car la question sera d'identifier les choix à privilégier et selon quel(s) critère(s).

QR_2 *Comment détecter et résoudre les différentes situations de conflit tout en garantissant la satisfaction des acteurs impliqués ?*

Une stratégie de résolution de conflits ne peut pas être considérée efficace si elle ne tient pas compte des préférences des différents acteurs de configuration et ne cherche pas à créer un compromis entre elles. Différentes stratégies de résolution de conflits ont été proposées dans la littérature, certaines sont adaptées à toute situation de travail collaboratif comme la médiation, l'arbitrage et la négociation (Khalil et al., 2014). Quant à la configuration collaborative des lignes de produits, l'étude systématique de la littérature (Edded et al., 2019) constate que la plupart des approches de configuration existantes ne fournissent aucun guidage explicite sur la gestion des conflits. Certaines proposent des stratégies de résolution rigides et inadaptées, à l'exception de (Stein et al., 2014) et (Ochoa et al., 2015) qui proposent des stratégies de résolution basées sur les préférences des acteurs.

À l'instar de ces deux approches, nous proposons dans cette thèse une stratégie de résolution qui combine la technique de calcul des sous-ensembles de correction minimaux (Minimal Correction Subsets (MCS)) (Liffiton et al., 2008) avec les préférences des acteurs pour garantir une résolution satisfaisante qui considère équitablement les différentes décisions et justifie le choix de résolution effectué.

I.4 Méthodologie de recherche

Le projet de recherche présenté dans ce manuscrit a été mené en utilisant la Méthodologie de Recherche Conceptuelle (MRC). La MRC est définie comme " l'étude scientifique et la création d'artefacts au fur et à mesure qu'ils sont développés et utilisés par des personnes dans le but de résoudre des problèmes pratiques d'intérêt général " (Johannesson and Perjons, 2014).

La Figure I.1 présente le modèle de processus de recherche conceptuelle proposé par (Johannesson and Perjons, 2014), et appliqué à ce projet de recherche. La méthode de recherche choisie vise à valider les questions de recherche soulevées au moyen d'évaluation de la solution proposée. Le processus de MRC a été mis en œuvre :

1. L'état de l'art sur la configuration des LPs et ses principales caractéristiques est présenté en Chapitres 2 et 3.
2. La problématique de recherche a été explorée à travers une revue systématique de la littérature (Chapitre 1).

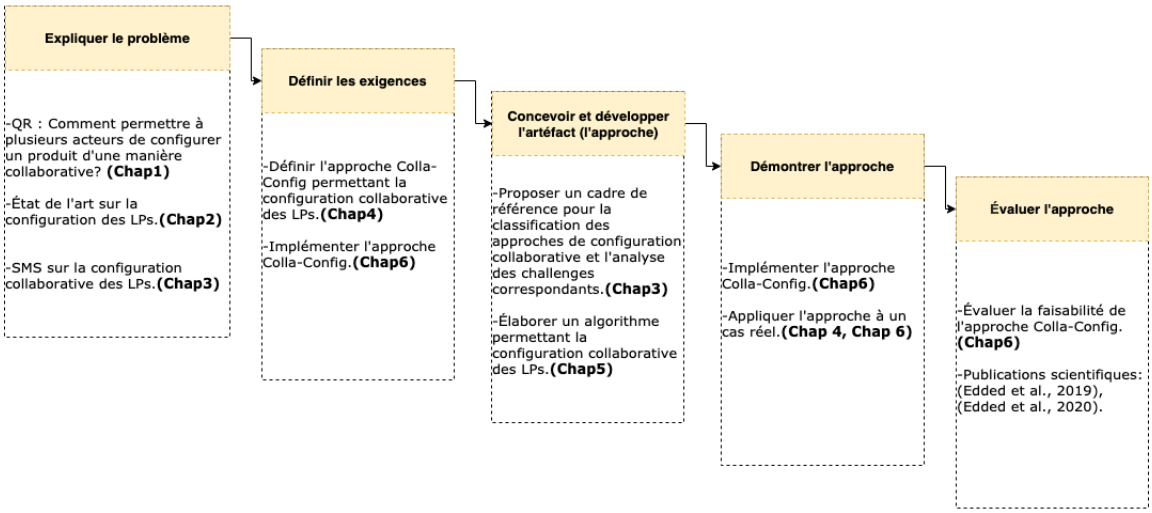


FIGURE I.1 – Modèle du processus de recherche conceptuelle (Johannesson and Perjons, 2014) appliqué à notre recherche.

3. Une analyse critique des travaux relatifs à la configuration collaborative des LPs a été réalisée à travers l'élaboration d'un cadre de référence pour les approches de configuration collaborative (Chapitre 3).
4. Une stratégie de résolution combinant la technique de calcul des sous-ensembles de correction minimaux (MCS) avec les préférences des acteurs a été proposée (Chapitres 4 et 5).
5. Un algorithme exploitant un modèle de variabilité d'une lignes de produits a été conçu pour l'approche de configuration collaborative proposée, et a été implémenté pour servir de prototype de la méthode(Chapitre 5).
6. Une évaluation de l'approche de configuration collaborative a été menée en effectuant une expérimentation contrôlée (Chapitre 6).

I.5 Contributions

Cette thèse présente 5 contributions principales :

1. **Une revue systématique de la littérature concernant les approches de configuration collaborative des LPs :**
   L'étude systématique menée (Edded et al., 2019) a montré qu'aucune solution satisfaisante n'a été proposée jusqu'à présent pour comparer et comprendre les approches de configuration collaborative des LPs. Cela nous a donc incité à explorer davantage le domaine en menant une revue systématique de la littérature. Celle-ci montre que les approches de configuration collaborative manquent de flexibilité tout en considérant équitablement les décisions des acteurs dans les situations conflictuelles et que la prise en compte de satisfaction des acteurs est nécessaire pour réussir le processus de configuration collaborative. Bien que la littérature offre une panoplie d'approches pour la configuration collaborative des LPs, aucune ne propose une stratégie de résolution qui garantisse la satisfaction des acteurs impliqués dans toutes les situations de divergence des décisions.

2. **Un cadre de référence pour l'analyse des approches de configuration collaborative des LPs :**
   Le résultat de la revue de la littérature a été utilisé pour conceptualiser un cadre de référence caractérisant les approches de configuration collaborative des LPs. Ce cadre est composé d'un ensemble de dimensions et facettes qui capturent les principaux aspects des approches collaborative de configuration et permet de :
   - Identifier les caractéristiques des différentes approches existantes.
   - Obtenir une vue d'ensemble de la configuration collaborative des LPs et comprendre comment la configuration collaborative peut être effectuée.
   - Aider à identifier les problèmes actuels et les défis soulevés dans la littérature.

3. **Une approche de résolution des conflits pour la configuration collaborative des LPs :**
   L'approche proposée permet la résolution des conflits en tenant compte des décisions de configuration des acteurs ainsi que de leurs préférences exprimées comme des règles de substitution (Edded et al., 2020). L'approche consiste à partager la totalité du modèle de caractéristiques entre les acteurs de la configuration. Ceux-ci vont dans un premier temps choisir parmi une liste de règles de substitution qui sert comme alternative d'aide à la résolution en cas de divergence de leurs décisions. La configuration finale est générée après fusion

et vérification des configurations partielles effectuées par les différents acteurs. La résolution des conflits est basée sur un algorithme qui calcule les différents ensembles minimaux de décisions à supprimer et considère la combinaison des règles de substitution choisies par les différents acteurs pour supprimer l'ensemble de décisions le plus adéquat. La méthode proposée est indépendante de tout langage de modélisation des LPs et elle pourra donc s'appliquer à tous types de modèles de LPs.

4. **Un outillage qui supporte l'approche :**
   L'outillage consiste au développement d'un prototype pour démontrer la faisabilité de l'approche. Ce prototype a été implémenté en utilisant le framework Vue et permet la configuration collaborative des LPs en offrant les fonctionnalités suivantes :
   (a) Expression de préférences à travers la sélection des règles de substitutions souhaitées.
   (b) Expression de choix de configuration à travers la sélection des caractéristiques désirées et non désirées.
   (c) Détection et résolution des conflits selon les préférences exprimées.
   (d) Validation et retour de la configuration finale à tous les utilisateurs.
5. **Une évaluation de l'approche :**
   L'approche a été évaluée de manière qualitative au moyen d'une expérimentation contrôlée. Cette expérimentation a été menée avec un groupe de 11 doctorants et consiste à évaluer la faisabilité de l'approche ainsi que l'utilisabilité du prototype associé en termes d'efficacité, efficience et satisfaction en appliquant le protocole expérimental de ISO/IEC Common Industry Format (CIF) (ISO/IEC, 2006).

I.6 Organisation de la thèse

La suite de ce manuscrit est structurée comme suit :

Le chapitre 2 présente les fondamentaux de la configuration des LPs.

Le chapitre 3 présente une revue systématique de la littérature sur la configuration collaborative des LPs. Ce chapitre rapporte les avancements, les limites et les challenges des approches identifiées, et les classifie au moyen d'un cadre de référence.

Le chapitre 4 donne un aperçu de la nouvelle approche de configuration collaborative proposée dans cette thèse. Un exemple d'application est développé pour illustrer, ensuite, une première présentation générale de l'approche proposée.

Le chapitre 5 présente une description détaillée de l'approche proposée.

Le chapitre 6 présente le processus d'expérimentation effectuée pour évaluer l'approche. Ce chapitre fournit une description détaillée des étapes de l'expérimentation menée ainsi qu'une synthèse des résultats obtenus après l'évaluation de la solution proposée.
Enfin nous concluons le manuscrit par une présentation des apports, des limitations et des perspectives de ce travail de recherche.

II

La configuration des lignes de produits

Sommaire

II.1 Introduction **18**

II.2 Concepts de base **18**

II.3 Types de configuration **21**

II.4 Les approches de configuration **24**

II.5 Conclusion **27**

II.1 Introduction

Ce chapitre qui présente un background de la configuration dans l'ingénierie des lignes de produits est organisé en cinq sections. La section 2 fournit les concepts de base nécessaires à la compréhension du processus de configuration. La section 3 introduit une typologie de la configuration des LPs. Ensuite, les différentes approches de configuration sont présentées dans la section 4. Section 5 clôturera le présent chapitre.

II.2 Concepts de base

II.2.1 L'ingénierie des lignes de produits

Une ligne de produits est définie comme un ensemble de systèmes partageant un ensemble de propriétés communes et satisfaisant des besoins spécifiques pour un domaine particulier (Pohl et al., 2005). Le paradigme de lignes de produits ne vise pas le développement d'un produit unique mais plutôt la conception et le développement d'une famille de systèmes. Ces derniers partagent des caractéristique en commun et se distinguent les uns des autres par des caractéristiques dites optionnelles.
Comme schématisé par la Figure II.1, l'ingénierie des LPs couvre deux niveaux complémentaires : l'ingénierie du domaine et l'ingénierie d'application.

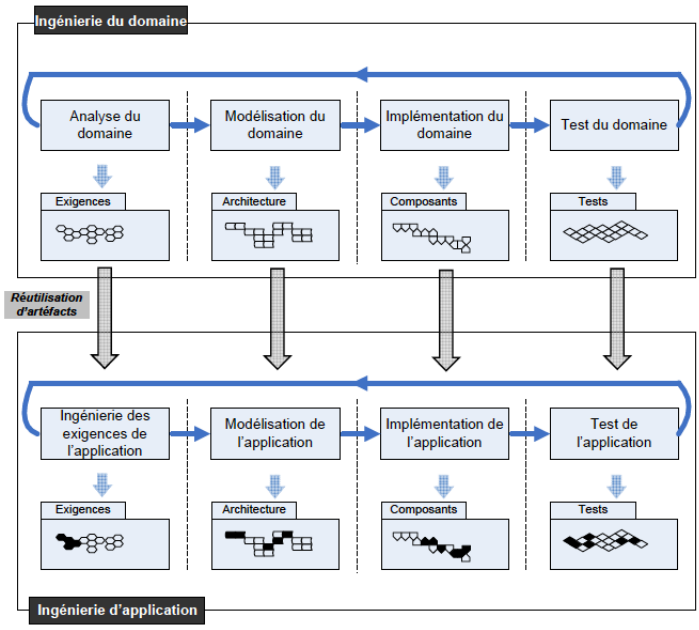


FIGURE II.1 – L'ingénierie des lignes de produits (Pohl et al., 2005)

L'ingénierie du domaine

L'ingénierie du domaine consiste à développer et construire des artéfacts qui seront réutilisés pour dériver différents produits. Un artéfact est un élément qui permet de développer un logiciel, par exemple un document de spécification, code,..etc. . Il s'agit d'un développement pour la réutilisation. Dans ce premier niveau, le domaine de la LP est analysé en identifiant les commonalités et les variabilitiés entre les produits.

La variabilité a été définie par (Weiss and Lai, 1999) comme "une hypothèse sur la façon dont les membres d'une famille peuvent se différencier entre eux". Dans le contexte de LP, la variabilité fait référence à une variété de produits développés par réutilisation d'un ensemble d'artéfacts prédéfinis et ajustables. Elle est modélisée lors de la phase d'ingénierie du domaine sous forme de points de variation et de variantes. Le point de variation représente le sujet de la variabilité et définit un endroit pour lequel le choix entre plusieurs alternatives doit être pris. Les variantes représentent l'objet de variabilité et définissent l'ensemble de valeurs possibles que peut prendre un point de variation (Pohl et al., 2005).

Il existe plusieurs méthodes pour l'analyse de domaine, la plus connue est FODA (Feature Oriented Domain Analysis) (Kang et al., 1990). Le domaine dans FODA est décrit sous forme d'un modèle de caractéristiques. Un modèle de caractéristiques d'une LP formalise les informations relatives aux produits de la LP sous forme de caractéristiques et de relations entre caractéristiques.

Une caractéristique est définie par (Kang et al., 1990) comme : "tout aspect important et distinctif ou caractéristique visible par les diverses parties prenantes ". Plusieurs autres notations de modélisation de caractéristiques ont succédé FODA et ont permis son extension telles que :

- FORM (Kang et al., 1998).
- FOPLE (Kang et al., 2003).
- FeatuRSEB (Griss et al., 1998)
- FORE (Riebisch et al., 2002).

L'ingénierie d'application

L'ingénierie d'application consiste à utiliser les artéfacts développés dans l'ingénierie du domaine pour dériver à chaque fois un produit particulier. La stratégie adoptée à ce niveau est basée sur le développement par réutilisation. En effet, le but est de maximiser le taux de réutilisation des artéfacts définis et développés pour construire un produit bien spécifique de la LP. Les éléments communs et les éléments variables doivent ainsi être exploités pour la dérivation des produits.

La dérivation des produits consiste à sélectionner les artéfacts qui seront présents dans le produit désiré. Ceci constitue le processus de configuration.
Le processus de configuration consiste à dériver une configuration à partir d'un modèle de la LP conformément aux exigences des parties prenantes en respectant les contraintes spécifiées au niveau du modèle (Hubaux and P.Heymans, 2009). Étant donnée la nature multidisciplinaire du modèle de la LP, le processus de configuration nécessite généralement l'intervention de plusieurs parties prenantes. Par conséquent, le processus de configuration est généralement effectué de manière collaborative.

Un ensemble de concepts relatifs au processus de configuration d'une LP sont définis dans la section suivante.

II.2.2 Définitions

Les concepts clés nécessaires à la compréhension de la contribution présentée dans cette thèse, sont : configuration, exigence de configuration, décision de configuration, acteur de configuration, rôle, conflit et priorité.

1. ***Configuration :*** Une configuration est définie par un sous-ensemble de caractéristiques sélectionnées à partir du modèle de la LP (Deelstra et al., 2005) qui : (a) respecte les contraintes de la LP et (b) est conforme aux exigences des parties prenantes.
   Les éléments d'une configuration correspondent donc aux exigences exprimées, et doivent inclure tous les éléments obligatoires et certains des éléments variables. Ces derniers doivent respecter les règles d'interdépendance, comme par exemple, le choix d'exactement un élément parmi un groupe d'éléments alternatifs (Djebbi, 2011).
   Une configuration peut être soit totale soit partielle. Elle est totale quand la variabilité est totalement résolue (une valeur de variante a été assignée a tous les points de variabilité). Elle est partielle quand certains des points de variabilité ne sont pas encore valués (aucune valeur ne leur a encore été assignée).
2. ***Exigence de configuration :*** Les exigences de configuration représentent les choix que les parties prenantes souhaitent retrouver dans le produit souhaité. Elles sont généralement exprimées dans des cahiers de charges indépendamment des modèles de la LP. C'est aux ingénieurs de développement de faire le rapprochement et la correspondance entre les exigences des clients et les fonctionnalités de la LP (Djebbi, 2011).
3. ***Décision de configuration :*** Lors de la configuration d'un produit, les parties prenantes expriment leurs exigences concernant le produit souhaité en spécifiant l'ensemble de caractéristiques désirées et non désirée. Par conséquent, une décision représente un choix lorsqu'une caractéristique est volontairement

sélectionnée ou non sélectionnée pour répondre aux différentes exigences exprimées (Mendonca et al., 2007).

4. ***Acteur de configuration :*** C'est toute personne directement ou indirectement impliquée dans le processus de configuration et qui peut être : un responsable de produit, un client, un ingénieur ou des experts techniques. Dans la littérature, différentes dénominations ont été attribuées aux acteurs de configuration telles que décideurs, utilisateurs et parties prenantes.
5. ***Rôle :*** C'est le moyen d'attribuer les décisions de configuration aux différents acteurs de configuration impliqués dans le processus de configuration du produit. Un acteur ayant rôle particulier est responsable de la configuration de toutes les parties du modèle qui lui ont été attribuées (Mendonca et al., 2007).
6. ***Conflit :*** Un conflit est une situation d'inconsistance où les exigences des différents acteurs se contredisent ou ne respectent pas les contraintes du modèle de la LP. (Osman et al., 2009) ont précisé qu'un conflit se produit lorsque deux ou plusieurs décisions de configuration attribuées aux différents acteurs de configuration ne peuvent pas être vraies en même temps.
   Un conflit peut être explicite ou implicite :
   - *Explicite :* Un conflit est explicite lorsqu'une caractéristique est exigée par un acteur et refusée par un autre.
   - *Implicite :* Un conflit implicite est identifié à travers les dépendances du modèle de la ligne, lorsque les décisions des acteurs violent les contraintes.

   Mis à part les conflits, deux autres situations d'inconsistance sont identifiées :
   (a) *Incohérence :* Elle est explicite et représente le cas où un acteur se contredit à lui-même il veut et ne veut pas la même caractéristique à la fois.
   (b) *Non conformité :* Elle est implicite et représente le cas où les décisions d'un acteur ne respectent pas les contraintes du modèle de la LP.
7. ***Priorité :*** Si deux décisions sont en conflit, la résolution consiste à donner la priorité à l'une des deux. Dans cette thèse, donner la priorité à une décision désigne le fait de conserver son état (sélectionnée/non sélectionnée) tel que exprimé par l'acteur de configuration.

II.3 Types de configuration

La configuration des LPs peut être faite de différentes manières selon la stratégie de production adoptée par l'entreprise. Dans cette section, différents types de configuration sont présentés.

II.3.1 Configuration totale

La configuration totale génère des configurations en une seule étape, sans configurations intermédiaires. Elle est par conséquent non-itérative : les configurations sont générées de façon automatique (Triki et al., 2014). Dans la littérature, on distingue trois approches de configuration totale qui sont : l'approche générative, l'approche a priori et l'approche de complétude.

II.3.1.1 Approche générative

Dans l'approche générative, seul les caractéristiques souhaitées par l'utilisateur sont sélectionnées. À la fin du processus de génération, plusieurs combinaisons (configurations) contenant les caractéristiques sélectionnées sont générées. Toutefois, il ne sera pas possible de les récupérer toutes. Une stratégie naïve de génération de produits est possible en utilisant un tableau de vérité. Les colonnes de la table de vérité représentent les caractéristiques de la LP. Il suffit ensuite de remplir la table de vérité avec des valeurs de type « Vrai » /« Faux » ou « 1» /« 0 » en générant exhaustivement toutes les combinaisons (Salinesi et al., 2010). Le problème avec l'approche générative est qu'il est difficile de sélectionner la configuration voulue parmi le grand nombre de combinaisons. Le passage à l'échelle devient donc une limite évidente.

II.3.1.2 Approche a priori

L'approche à priori permet à l'utilisateur de définir librement ses choix de configuration et tester leur consistance. Dans cette approche, l'utilisateur doit définir une nouvelle configuration chaque fois que son choix précédent viole les contraintes de la LP. Par exemple, l'utilisateur exprime le choix de configuration suivant : C=(F1, F2, F3, F4, F5) et souhaite savoir si elle est valide. La configuration C est rejetée par le configurateur, car il existe une contrainte qui spécifie que F2 exige F6 alors que F6 ne fait pas partie de la configuration exprimée. L'utilisateur doit alors réviser sa configuration et la tester encore une fois, et ce, jusqu'à l'obtention d'une combinaison valide.
La limite de cette approche est liée au risque de proposer des configurations non valides à chaque fois : c'est le problème de probabilités (Triki et al., 2014).

II.3.1.3 Approche de complétude

Dans l'approche de complétude, l'utilisateur peut exprimer partiellement ses choix (c.f Section 3.2.1 configuration partielle) et l'outil de configuration se charge de compléter la configuration partielle donnée pour en avoir une complète et qui

respecte les contraintes du modèle de la LP. L'avantage avec cette approche est que l'espace de solutions est réduit comparé à celui de l'approche générative. Cependant, il est possible qu'aucune solution contenant la configuration partielle donnée existe.(Triki et al., 2014).

II.3.2 Configuration progressive

La configuration progressive est obtenue progressivement par une série de spécialisations successives du modèle de la LP durant laquelle certains choix de configuration sont éliminés (Czarnecki et al., 2005). Le caractère progressif de la configuration permet soit de compléter une telle configuration ou de revenir en arrière pour réviser certains choix ce qui n'est pas possible avec la configuration totale. La configuration progressive initialement proposée par (Czarnecki et al., 2005) a été améliorée par (Mendonca et al., 2007) qui ont proposé une approche qui consiste à organiser les différentes spécialisations de la configuration progressive par un workflow durant lequel plusieurs acteurs interviennent. Cette approche fait référence à la configuration collaborative. En effet, dans les projets réels, les modèles de LPs peuvent contenir jusqu'à 10 000 caractéristiques (Batory et al., 2006), la configuration de ces modèles est généralement une tâche ardue et de longue haleine nécessitant l'intervention de plusieurs parties prenantes à des moments différents. Celles-ci partagent les activités de configuration afin de décider l'ensemble de caractéristiques du produit souhaité. On distingue deux types de configuration progressive : la configuration partielle et la configuration par étape.

II.3.2.1 Configuration partielle

Une configuration partielle représente la sélection de certaines caractéristiques du modèle de LP, tandis que certains autres choix ne sont pas encore entrepris. La configuration partielle représente ainsi le résultat intermédiaire entre deux spécialisations.
Une spécialisation est un processus de transformation qui prend en entrée un modèle de LP et produit un autre modèle où, seulement certains choix de configuration sont éliminés. L'ensemble des configurations représentées par le modèle spécialisé est un sous-ensemble des configurations représentées par le premier modèle (Djebbi, 2011).

II.3.2.2 Configuration par étapes

La configuration par étape consiste à faire la transition d'une configuration de départ à une configuration cible à travers une série de configurations intermédiaires.

Ces dernières ne représentent pas des configurations partielles mais des configurations complètes fournies à la fin de chaque étape de la configuration par étape. En effet, certaines entreprises se trouvent parfois dans des situations où il faut nécessairement fractionner la configuration sur plusieurs étapes afin d'adhérer à un changement de contraintes, tels que le budget maximum de développement par année. La configuration par étapes vient pallier à ce genre de contraintes relatif à cette problématique (White et al., 2009).

II.4 Les approches de configuration

Quatre catégories d'approches de configuration peuvent être distinguées : la configuration non guidée, la configuration dirigée par les modèles, la configuration dirigée par la logique et la configuration basée sur des algorithmes ad-hoc (Djebbi, 2011). Le principe de chacune est détaillé dans les sous-sections suivantes.

II.4.1 La configuration non guidée

Dans cette approche de configuration, le modèle de la ligne est configuré sans suivre un ordre bien défini. L'idée est que l'utilisateur exprime la configuration du produit souhaité avec une simple sélection des caractéristiques d'une manière descendante à partir du modèle de la LP.
Il s'agit de parcourir de façon manuelle le modèle et sélectionner/ omettre les caractéristiques au fur et à mesure que des décisions de configuration sont prises au niveau des points de variation.
Bien qu'elle permette d'aboutir à la dérivation d'un produit valide, la configuration non guidée présente certaines limites tels que :

- La taille du modèle, quand il contient un grand nombre de caractéristiques et de contraintes, ceci rend la configuration manuelle pénible et risque l'erreur.
- Proportionnellement à la taille du modèle, la prise en compte manuelle des interdépendances entre les caractéristiques de la ligne de produits est encore difficile.
- La difficulté du retour sur les décisions déjà prises vu que ça nécessite la mémorisation des choix exprimés au niveau de différents points de variation ainsi que la gestion de l'impact de leur propagation.

Pour pallier à ces limites, plusieurs approches d'automatisation des activités de configuration ont été proposées. Le principe de ces approches consiste à formaliser des langages de modélisation de la variabilité afin de pouvoir les analyser et les configurer de manière automatique selon les choix de l'utilisateur. Ces approches

peuvent être classées en deux catégories : les approches dirigées par les modèles et celles dirigées par la logique (Djebbi, 2011).

II.4.2 La configuration dirigée par les modèles

La configuration dirigée par les modèles s'appuie sur un modèle permettant de prendre des décisions sur les caractéristiques souhaitées de la LP. Les modèles sont utilisés pour définir et communiquer les connaissances relatives à la variabilité pour la dérivation de nouveaux produits.
Dans cette catégorie d'approches, le langage UML est utilisé pour résoudre la variabilité par la transformation des modèles (Ziadi et al., 2003). la transformation consiste à transformer le modèle de la LP en un modèle de produit modélisé par un diagramme de classes UML (Ziadi, 2011). La variabilité d'une LP est aussi modélisée par un modèle de décision en utilisant le DoplerVML (Decision-oriented Variability Modelling Language) qui modélise les options de personnalisation comme un ensemble de décisions(Dhungana et al., 2011). Ensuite, selon les valeurs de décisions fixées par un utilisateur, les éléments nécessaires à la composition du produit sont automatiquement déterminés et des configurations de produit peuvent être générées.
D'autres exemples d'approches de configuration dirigée par les modèles ont également été proposées telles que (Atkinson et al., 2002), (Flege, 2000) et (Cerón et al., 2004). Ces approches ont été inspirées des travaux de (Jézéquel, 1998) et (Jézéquel, 1999) qui ont proposé des solutions qui supportent la dérivation des modèles de produits à la conception.
De manière générale, les approches de configuration dirigées par les modèles s'appuient sur un langage de modélisation de la variabilité de la LP afin de générer des modèles de produits valides comme instances du modèle de la LP(Djebbi, 2011).

II.4.3 La configuration dirigée par la logique

Dans cette catégorie, les approches de configuration peuvent être classifiées comme suit selon la logique utilisée : les approches basées sur la logique propositionnelle, celles basées sur la logique de description et celles basées sur la programmation par contraintes. Les différentes sous-catégories sont détaillées dans les sous-sections suivantes.

II.4.3.1 La configuration dirigée par la logique propositionnelle

L'approche de configuration dirigée par la logique propositionnelle consiste à traduire le modèle de la LP en formules propositionnelles ainsi que le modèle de la

configuration dérivée. Il s'agit ensuite d'analyser ces formules afin de déterminer la consistance de la configuration et la validité du modèle de la LP. Différents travaux dans la littérature ont proposé l'utilisation des formules propositionnelles comme moyen pour l'analyse automatique des modèles de lignes de produits tels que (Batory et al., 2006) et (White et al., 2009).
Une formule propositionnelle est un ensemble de variables booléennes connectées par des opérateurs logiques ($\neg$, $\vee$, $\wedge$, $\leftrightarrow$, $\rightarrow$).
La consistance de ces formules est analysée par des solveurs de type SAT. Un solveur SAT est un logiciel sur étagère qui résout la SATisfiabilité des propositions données en entrée. Il détermine si la proposition est satisfaisable ou non, en vérifiant l'existence d'une affectation des variables qui la rend vraie.

II.4.3.2 La configuration dirigée par la logique de description

La logique de description est une famille de langages qui permettent la représentation de la connaissance terminologique d'un domaine d'application d'une manière formelle et structurée. Elle permettent également le raisonnement sur les connaissance en utilisant des systèmes de raisonnement logique spécifiques (Baader et al., 2007).
Un système de raisonnement de logique de description prend en entrée un problème décrit par une logique de description et retourne des résultats de vérification de la cohérence. (Wang et al., 2005) ont proposé la première approche basée sur la logique de description pour analyser les modèles des LPs. L'approche proposée consiste à utiliser le langage de description OWL-DL (Web Ontology Description Language) pour transformer le modèle de la LP en une ontologie. Un moteur de raisonnement de logique de description tel que RACER est employé pour analyser de manière automatique la représentation OWL du modèle de la LP.
Généralement, les approches de cette catégorie proposent la traduction des modèles de caractéristiques en logique de description et l'utilisation des moteurs de raisonnement pour effectuer les analyses de véracité nécessaires. Les différents travaux ont été présentés en détail dans la thèse de (Djebbi, 2011).

II.4.3.3 La configuration dirigée par la programmation par contraintes

Ce type d'approches de configuration consiste à traduire le modèle de LP ainsi que le modèle de produit en un problème de satisfaction de contraintes (appelé aussi CSP pour Constraint Satisfaction Problem). Un CSP (Ghédira, 2012) est défini comme un ensemble de variables et un ensemble de contraintes qui restreignent les valeurs que peuvent prendre ces variables.
L'analyse de modèle sous forme de CSP permet de déterminer si les modèles de la

LP et de produit sont valides. L'analyse se fait grâce à l'utilisation de solveurs de contraintes sur étagère.
L'analyse du modèle en tant que CSP permet de générer des modèles de produits et vérifier leur consistance. La résolution d'un problème dCSP consiste à attribuer des valeurs aux variables de manière à satisfaire toutes les contraintes en utilisant des solveurs.
Différents travaux de recherche ont proposé l'utilisation de la programmation par contraintes pour l'analyse automatique des modèles tels que (Batory et al., 2006), (Salinesi et al., 2010) et (Mazo et al., 2011).

II.4.4 La configuration basée sur les algorithmes ad-hoc

Cette catégorie de configuration, inclut toute approche de configuration qu'on ne peut pas classifier dans les catégories précédentes. Ces approches sont soit des études dont la logique conceptuelle utilisée est non clairement exposée ou des études utilisant des algorithmes ou des outils d'analyse ad-hoc (Djebbi, 2011).
Comme exemple de cette catégorie d'approche, on cite l'approche de (Bachmeyer and Delugach, 2007) qui se base sur les graphes conceptuels (Sowa, 1984) pour la modélisation des modèles de caractéristiques. Cette approche permet de vérifier la validité des modèles et de générer des configurations valides mais de façon manuelle.
Une panoplie d' approches ad-hoc ont été abordée dans la thèse de Djebbi (Djebbi, 2011).

II.5 Conclusion

Ce chapitre a été consacré à la présentation des fondamentaux de la configuration des LPs. D'abord, les principales définitions nécessaires à la compréhension de la configuration des LPs et plus particulièrement les travaux de cette thèse ont été présentées. Ensuite, nous avons identifié les différents types de la configuration. Puis, nous nous sommes focalisées sur les approches de configuration des LPs.
Une arborescence synthétisant les différents types et approches de configuration présentés dans ce chapitre est illustrée par la Figure II.2. Le chapitre suivant présentera l'état de l'art de la configuration collaborative des LPs.

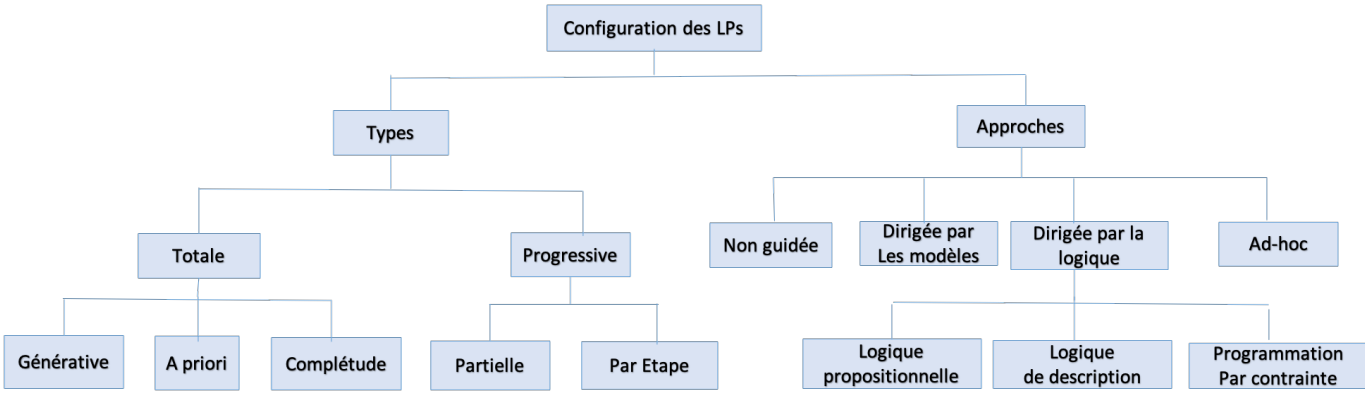


FIGURE II.2 – Arborescence des types et approches de configuration des LPs

III

La configuration collaborative des lignes de produits

Sommaire

III.1 Introduction **30**

III.2 Processus d'élaboration du cadre de référence proposé . **30**

III.3 Présentation du cadre de référence PL2C **37**

III.4 Synthèse **47**

III.5 Les défis de la configuration collaborative **48**

III.6 Conclusion **50**

III.1 Introduction

Ce chapitre présente la configuration collaborative des lignes de produits. Nous avons identifié à travers une étude systématique de la littérature les différentes approches existante de configuration collaborative des lignes de produits. Nous procédons dans ce chapitre à la caractérisation de ces différents travaux à travers l'élaboration d'un cadre de référence. Le cadre est composé d'un ensemble de dimensions permettant de positionner et comparer les travaux identifiés dans la littérature. Le chapitre conclut par une synthèse de la classification et une analyse des principaux résultats.

III.2 Processus d'élaboration du cadre de référence proposé

Le cadre de référence employé dans cette thèse s'inspire des cadres proposés dans un premier temps pour l'ingénierie des systèmes et qui ont prouvé leur efficacité comme un moyen de bien comprendre diverses disciplines d'ingénierie comme l'ingénierie des systèmes d'information (Jarke et al., 1992), l'ingénierie des besoins (Jarke et al., 1993), l'ingénierie des processus (Rolland, 1998) et l'ingénierie des exigences (Rolland et al., 1998). De ce fait, et après une étude systématique de la littérature, nous proposons le cadre de référence, baptisé PL2C Framework (*Product Line Collaborative Configuration Framework*). Ce cadre a été élaboré pour les raisons suivantes :

- Comprendre le processus de configuration collaborative des LPs.
- Guider la classification et la comparaison des approches de configuration collaborative.
- Identifier les problèmes et les défis qu'on peut rencontrer durant le processus de configuration collaborative.

Pour élaborer ce cadre (Edded et al., 2019), une étude systématique de la littérature (Systematic Mapping Study) a été menée en identifiant les trois questions de recherche suivantes :

1. Quelles sont les principales caractéristiques des approches dans le domaine de la configuration collaborative des LPs ?
2. Quelles sont les points forts et faibles de chacune des approches de configuration collaborative existantes ?

3. Quels sont les défis majeurs de la configuration collaborative dans le domaine de l'ingénierie des LPs ?

Le but d'une étude systématique de la littérature est d'étudier la littérature dans un domaine particulier afin de déterminer la nature, la portée et le nombre d'études primaires publiées (Petersen et al., 2008). Elle facilite l'obtention d'une vision plus large des différents domaines de recherche qui sont généralement vastes et souvent mal définis (Petersen et al., 2008).

Comme illustré dans la Figure III.1, le processus d'élaboration du cadre comprend sept sous-processus qui sont détaillés dans les sous-sections qui suivent : recherche manuelle, recherche des mots et des termes, recherche automatique, sélection des études, identification des mots-clés, construction du cadre et finalement la validation du cadre. Le processus d'élaboration a été mené en suivant le protocole décrit par (Petersen et al., 2015).

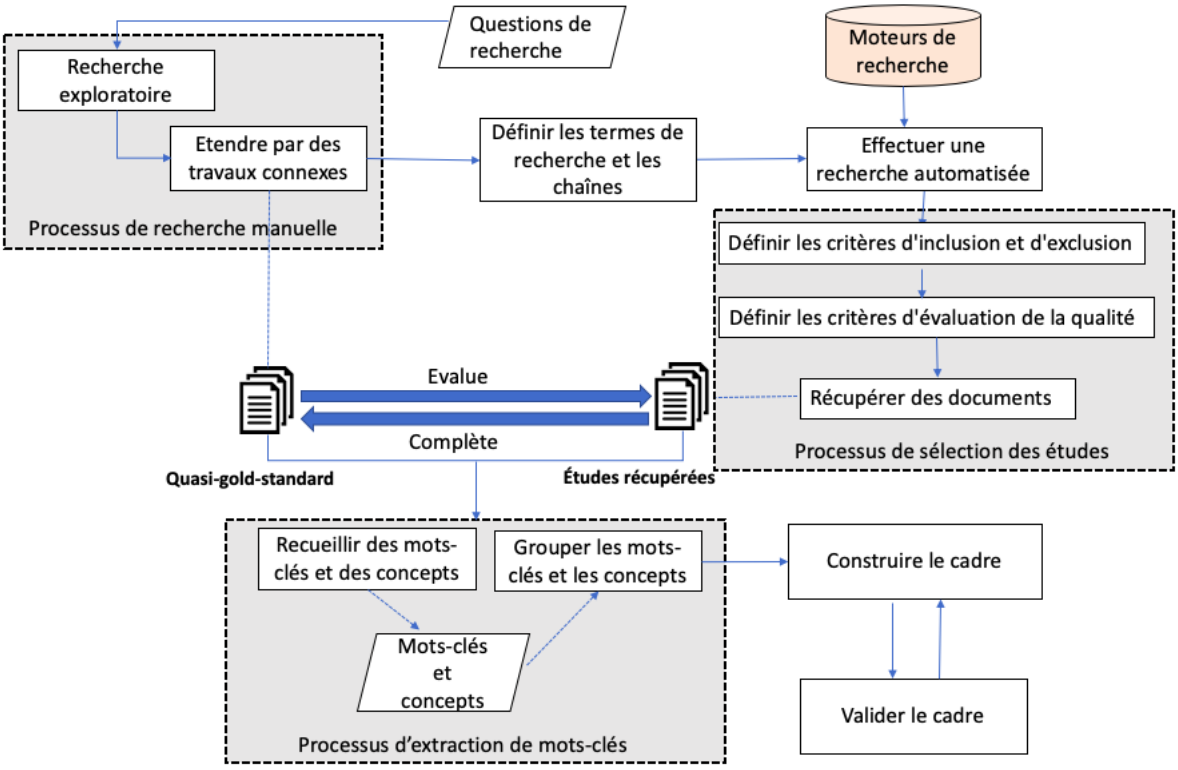


FIGURE III.1 – Processus d'élaboration du cadre PL2C

III.2.1 Recherche manuelle

Le sous-processus de recherche manuelle consiste à appliquer un processus de "snowballing" à un ensemble d'articles décrivant les premières approches proposées

par certains auteurs connus dans le domaine, à savoir [S1, S3, S4, S11] [1].
La recherche manuelle a été menée en deux étapes : (i) recherche exploratoire dans laquelle un processus de "forward snowballing" a été appliqué. Au cours de cette étape, une collection initiale de 14 publications a été obtenue après avoir suivi les références des publications citées dans l'ensemble ci-dessus. (ii) extension par des travaux connexes où un processus de "backward snowballing" a été appliqué sur la collection initiale de publications en vérifiant leurs références et en sélectionnant les publications pertinentes pour cette étude. Cela a donné lieu à 8 publications supplémentaires [S9, S10, S20, S21, S22, S23, S25, S31]. Afin d'éviter toute redondance, les publications non examinées telles que les rapports techniques, les livres et les descriptions d'ateliers, n'ont pas été incluses dans la série d'études retenues.
Les résultats de la recherche manuelle ont été utilisés pour établir un Quasi-Gold Standard (QGS), qui est un ensemble d'études connues représentant les principales contributions identifiées dans le domaine de recherche (Zhang and Babar, 2009). Le QGS a été ensuite utilisé pour obtenir les chaînes de recherche pour la recherche automatisée.

III.2.2 Identification des termes et chaines de recherche

Afin de s'assurer que tous les travaux liés à la configuration collaborative des LPs ont été explorés, une recherche automatisée a été effectuée en se basant sur les termes et les chaînes objectivement dégagés de l'ensemble de QGS. Dans la littérature, les termes de recherche constituent une étape cruciale dans le processus d'étude systématique de la littérature, qui consiste à obtenir des termes relatifs au domaine pour récupérer des études pertinentes (Zhang and Babar, 2009).
L'identification des termes de recherche est effectuée à l'aide de l'exploration de texte dans chaque papier du QGS afin d'analyser statistiquement les mots les plus fréquents. Le titre et le résumé de chaque article ont été importés sous QDA Miner et WordStat [2] qui sont des outils d'analyse permettant de déterminer les termes les plus fréquents et identifier la relation entre eux. La Figure III.2 présente des groupes de termes dérivés avec une fréquence élevée. Les couleurs représentent l'ensemble des termes fortement liés les uns aux autres.
En général, ces termes peuvent être divisés en plusieurs groupes qui ont un sujet précis (cf. Tableau III.1). Certains mots ,malgré leur faible fréquence, sont étroitement liés à la configuration collaborative des LPs, telles que la coordination, la résolution et la détection. Afin d'élargir la couverture des résultats de recherche, ces termes ont été ajoutés à la série de termes obtenus. Les synonymes des termes obtenus, par

1. La liste complète des études identifiées est disponible dans l'annexe.
2. `http://provalisresearch.com/products/`

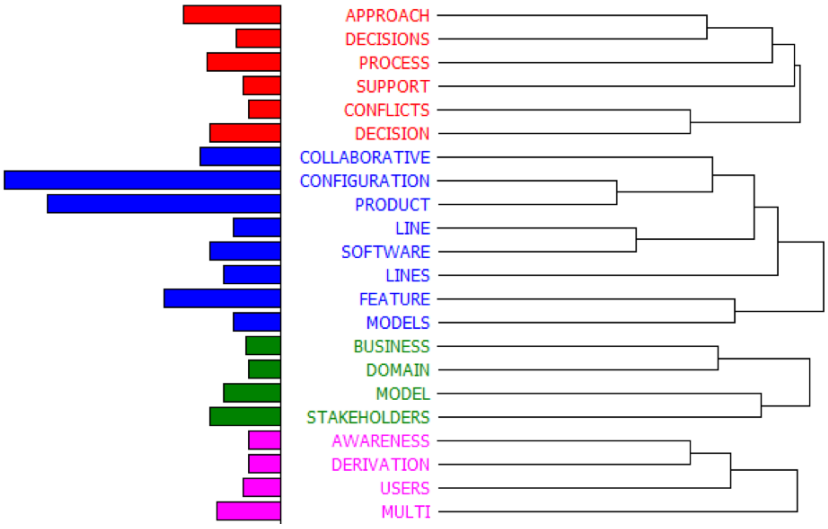


FIGURE III.2 – Résultat de regroupement des termes à haute fréquence

exemple parties prenantes et utilisateurs, décision ou décisions, modèle ou modèles, ont été pris en compte dans les chaînes de recherche finales. Enfin, chaque groupe de termes définissant les différents sujets a été utilisé pour dériver une ou plusieurs des chaînes de recherche présentées dans le Tableau III.2. En effet, le groupe de termes présentant le domaine d'application a été pris en compte dans toutes les chaînes de recherche dérivées ; le deuxième groupe de termes a été utilisé pour formuler la première chaîne de recherche (S1) ; le troisième groupe de termes pour formuler la deuxième chaîne de recherche (S2), et le dernier a permis de formuler la troisième et la dernière chaîne de recherche (S3, S4).

TABLE III.1 – Groupe des termes.

Group	Term	Subject
1	Product line(s)/software product line(s), multi product line(s)	Application domain
2	Feature model(s), collaborative, configuration, derivation, business, process	Specific process
3	Stakeholders/users, multi	Collaboration context
4	Approach, awareness, decision(s), support, conflicts	Conflict resolution mechanism

III.2.3 Recherche automatisée

Conformément aux directives précisées par (Petersen et al., 2015), un ensemble de bases de données scientifiques a été sélectionné pour effectuer la recherche automatisée, notamment : IEEE Xplore, ACM DL, Science Direct, Springer, Scopus et Web of science. Le résultat de recherche basée sur les chaînes dérivées (cf. Tableau III.2) est indiqué dans le Tableau III.3. La deuxième colonne indique le nombre d'études retournées par chaque chaîne de recherche. La troisième colonne indique

Table III.2 – Chaînes de recherche dérivées.

No	Configuration process	Search String
S1	Process nature	((software product line **OR** multiple product line) **AND** (configuration **OR** derivation)) **AND** (collaborative process **OR** collaboration process **OR** process collaboration).
S2	Multi user involvement	((software product line **OR** multiple product line) **AND** (configuration **OR** derivation)) **AND** (stakeholders **OR** users **OR** multiple stakeholders **OR** multiple users).
S3	Conflict resolution	((software product line **OR** multiple product line) **AND** (configuration **OR** derivation)) **AND**(conflict resolution **OR** conflict detection **OR** resolution of conflicts **OR** detection of conflicts).
S4	Process coordination	((software product line **OR** multiple product line) **AND** (configuration **OR** derivation)) **AND** (coordination of decisions **OR** decision coordination **OR** supporting awareness **OR** awareness support **OR** support of awareness).

Table III.3 – Résultats de la recherche automatisée.

Moteur de recherche	Nombre de documents trouvés				Total	Premier tour de sélection des articles
	S1	S2	S3	S4		
IEEE Xplore	29	10	0	1	40	14
ACM DL	41	15	23	9	88	14
Science Direct	5	5	0	3	13	3
Springer	15	10	4	8	37	5
Scopus	9	14	5	14	42	19
Web of Science	6	8	0	4	18	11

le nombre total d'articles retournés par moteur de recherche. La quatrième colonne représente le nombre d'études retenues après élimination des doublons par moteur de recherche.

III.2.4 Sélection des études

Le sous-processus de sélection des études consiste à identifier les études les plus pertinentes au domaine de configuration collaborative des LPs. Ce processus comprend quatre étapes comme indiqué par la Figure III.3. 238 documents ont été initialement recueillis. Au cours de la deuxième étape, les documents dupliqués par moteur de recherche et les études qui n'appartiennent pas au domaine de recherche ont été rejetés. En conséquence, 66 des 238 articles ont été retenus. Au cours de la troisième étape, les études ont été filtrées selon les critères d'inclusion et d'exclusion présentés dans le Tableau III.4. Deux critères d'inclusion et trois critères d'exclusion ont été identifiés : Le premier critère d'inclusion (CI1) consiste à inclure les études qui ont été publiées entre 2005 (la première approche a été proposée par Czarnecki en 2005) et 2018. Le deuxième critère (CI2) permet d'inclure seulement les études qui portent sur la configuration collaborative des LPs. Ceci est déterminé en fonction du titre, du résumé et des mots clés. Pour les critères d'exclusion, le premier critère (CE1) consiste à exclure les articles qui ne sont pas rédigés en anglais. Le deuxième critère (CE2) exclut les copies multiples du même article. Le dernier critère (CE3) exclut les articles qui ne sont pas examinés par des comités scientifiques.

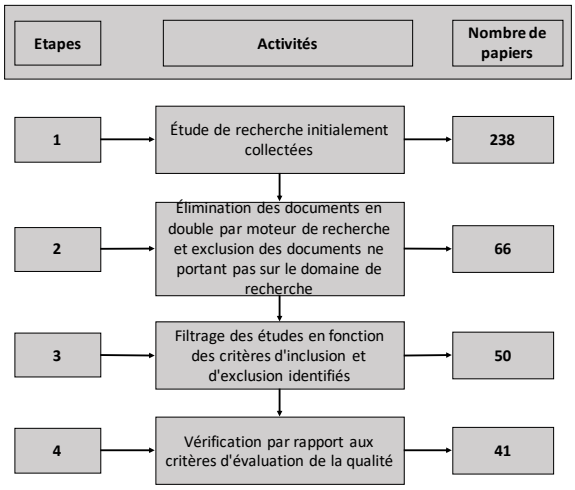


FIGURE III.3 – Les étapes du processus de sélection des documents.

TABLE III.4 – Liste des critères d'inclusion et d'exclusion.

Critères d'inclusion	Critères d'exclusion
CI1. Documents de recherche de 2005 à 2018	**CE1.** Toutes les études de recherche qui ne sont pas rédigées en anglais
CI2. Tous les travaux de recherche qui portent sur la configuration collaborative des lignes de produits en se basant sur le titre, les mots clés et le résumé des articles	**CE2.** Duplicata d'articles provenant de différentes bases de données scientifiques : l'exclusion des copies multiples de la même étude
	CE3. Tous les documents qui ne sont pas examinés par une comité de lecture (résumé, poster, proposition, rapport technique, thèse)

À la dernière étape, l'ensemble des documents récupérés de l'étape précédente (50 documents) a été vérifié par rapport aux critères d'évaluation de la qualité présentés dans le Tableau III.5. Chaque question de qualité peut avoir une réponse parmi

TABLE III.5 – Critères d'évaluation de la qualité.

Critère	Question
Q1	L'étude se concentre-t-elle sur la configuration collaborative des LPs ?
Q2	L'étude inclut-elle les réponses potentielles aux questions de recherche ?
Q3	L'étude fait-elle référence à des pratiques concrètes de configuration collaborative et ne se contente-t-elle pas de les traiter de manière générale ?
Q4	L'objectif de l'étude est-il clair ?
Q5	L'étude illustre-t-elle certaines caractéristiques principales de la configuration collaborative ?
Q6	L'étude présente-t-elle des résultats originaux qui n'ont pas été publiés ailleurs ?

les trois réponses suivantes : "oui", "partiellement", et "non". Chaque étude a été évaluée selon ces critères. Si une étude a pour réponse "oui", alors 1 lui est attribué, si elle a "partiellement", 0,5 lui est attribué, et un 0 est donné à une étude qui a reçu "non"comme réponse.

Le critère Q1 garantit que le document traite tout sujet lié à la configuration collaborative des LPs. Le critère Q2 garantit que le document répond au moins à

une question de recherche (par exemple, les documents présentant des aperçus rapides, des approches d'analyse comparative). Le critère Q3 exige que des pratiques concrètes de configuration collaborative sont totalement ou partiellement décrites. La combinaison des critères Q2 et Q3 garantit que les travaux proposent effectivement des approches de configuration collaborative ou traitent une de ses étapes. Le critère Q6, permet de retenir les études étendues si elles présentent une nouveauté par rapport aux études originales.
L'évaluation de la qualité a été effectuée en étudiant le titre, le résumé et le contenu de chaque article. Afin de garantir la fiabilité des résultats de l'évaluation de la qualité, seules les études ayant obtenu un score supérieur ou égal à 3 ont été incluses, soit la moitié du total des points correspondant aux 6 critères de qualité. En conséquence, 41 [3] sur 238 études de recherche ont été sélectionnées pour représenter le domaine de recherche.

III.2.5 Processus d'extraction de mots-clés

Pour identifier les concepts relatifs à la configuration collaborative des LPs, un processus d'extraction de mots-clés (keywording en anglais) systématique proposé par (Petersen et al., 2008) a été appliqué. Ce processus permet de définir un cadre de référence en considérant tous les concepts identifiés dans les études récupérées. Il a été adopté dans plusieurs travaux proposant des cadres de référence tel que (Franzago et al., 2017). Comme illustré dans la Figure III.1, le processus d'extraction de mots-clés comporte deux étapes :

- *Extraire des mots-clés et des concepts :* les mots-clés et les concepts ont été identifiés en lisant chaque étude notamment les sections qui contiennent des termes identifiés durant le processus d'identification des termes et chaines de recherche. Après avoir analysé toutes les études collectées, tous les mots-clés et tous les concepts ont été combinés pour identifier clairement le contexte, la nature et la contribution de chaque étude.
- *Grouper les mots-clés et les concepts :* les mots-clés et les concepts identifiés ont été ensuite regroupés en des ensembles représentatifs selon quatre dimensions : collaboration, but, processus et outil. Ces dimensions ont été choisies car elles couvrent l'ensemble des informations nécessaires pour caractériser la configuration collaborative des LPs. Chaque dimension est caractérisée par des facettes facilitant la compréhension et la classification des différents aspects de la configuration collaborative. La classification par facette a été initialement

3. Le résultat du score de qualité pour chaque étude sélectionnée est disponible sur le lien suivant `https://bit.ly/31TfgE2` (Fiche 2)

proposée par(Prieto-Diaz and Freeman, 1987), et est basée sur la définition des classes d'attributs instanciées avec des valeurs différentes.

III.2.6 Construction et validation du cadre de référence

Le cadre de référence a été construit en structurant les concepts précédemment identifiés sous forme de dimensions et facettes. Ces dernières représentent les différents aspect spécifiques de la configuration collaborative des LPs.
Afin de fournir un cadre aussi bien défini et complet, plusieurs itérations de vérification ont été appliquées sur des versions intermédiaires où seuls les aspects et les caractéristiques suffisamment discriminants ont été conservés. Au cours de chaque itération, le cadre a été examiné et chaque désaccord a été discuté et résolu.
La validation du cadre a été effectuée en deux étapes :

1. Réalisation d'une étude systématique pour collecter les informations caractérisant la configuration collaborative des LPs. L'ensemble des mots clés et des concepts extraits des études ont été utilisés pour construire le cadre et prouver son exhaustivité.
2. Classification de toutes les études collectées en utilisant le cadre. Au cours de cette étape, le texte complet de chaque article a été lu et ses données ont été extraites selon les différentes dimensions. Une analyse complète de tous les travaux identifié est reportée dans la section 4.

III.3 Présentation du cadre de référence PL2C

Comme illustré dans la Figure III.4, le cadre de référence est composé de quatre dimensions : but, collaboration, processus et outils. Ces dimensions permettent de capturer un aspect pertinent des approches de configuration collaborative des LPs. Chaque dimension est composée par un ensemble de facettes qui facilitent la ca-

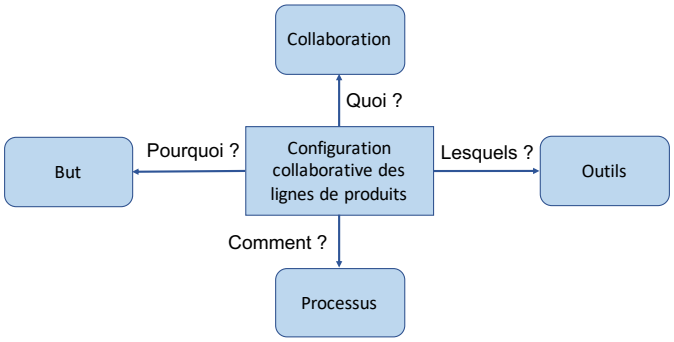


FIGURE III.4 – Les quatre vues du PL2C

ractérisation de la configuration collaborative. Les facettes sont définies à l'aide d'attributs décrits par un ensemble de valeurs permettant de mesurer et de positionner l'aspect observé (Abri et al., 2016).
Un attribut est qualifié par un ensemble de valeurs définies dans un domaine. Ces valeurs peuvent être : (a) de type prédéfini (entier, booléen, etc.), (b) de type énuméré (ENUM{x, y}), (c) ou type structuré (ENSEMBLE ou TUPLE) et servent à mesurer et positionner les caractéristiques des approches de configuration collaborative.
L'approche multi-dimensions et multi-facettes adoptée permet de capturer l'ensemble des informations nécessaires pour caractériser la configuration collaborative des LPs avec :

— Des dimensions qui permettent de montrer la variété et la diversité des aspects pouvant caractériser une approche de configuration collaborative.

— Des facettes qui fournissent une description en profondeur de ces aspects.

— Des relations entre les facettes qui facilitent la compréhension de l'indépendance entre les dimensions.

Les approches de configuration collaborative des LPs identifiées dans la littérature sont étudiées à travers le cadre afin de fournir des réponses aux questions de recherche suivantes :

- QR1 : Quelles sont les principales caractéristiques des approches dans le domaine de la configuration collaborative des LPs ?
  Le cadre est composé de quatre dimensions, comme illustré dans la Figure III.4. Les caractéristiques extraites des études récupérées ont été regroupées en fonction de ces dimensions :
  — La dimension *But* permet d'expliquer les objectifs des approches de configuration collaborative.
  — La dimension *Collaboration* définit ce qu'est la collaboration dans la configuration des LPs et de quoi il s'agit (par exemple, mode de collaboration, interaction et niveau de collaboration).
  — La dimension *Processus* explique comment se déroule le processus de configuration collaborative des LPs.
  — La dimension *Outils* représente la mise en œuvre et plus particulièrement l'ensemble d'outils développés pour supporter une ou plusieurs tâches de la configuration collaborative.
- QR2 : Quelles sont les points forts et faibles des approches de configuration collaborative existantes ?

Étant donné que les dimensions sont définies avec de multiples facettes, les caractéristiques de chaque approche, y compris les forces et les faiblesses, peuvent être identifiées par le positionnement des approches existantes dans le cadre.

- QR3 : Quels sont les principaux défis de la configuration collaborative dans le domaine de l'ingénierie des LPs ?
  De même, les défis peuvent être identifiés grâce à l'analyse de classification de toutes les approches existantes.

III.3.1 La dimension But

Cette dimension est composée de trois facettes qui permettent de caractériser l'objectif de l'approche de configuration collaborative. Ces facettes sont : "Principe", "Scénario de collaboration" et "Satisfaction des parties prenantes".

III.3.1.1 Principe

Les approches étudiées montrent que la prise de décision lors de la configuration collaborative peut être : (a) partagée entre deux ou plusieurs parties prenantes (Mendonca et al., 2008) et (b) répartie entre les différentes parties prenantes où chaque partie prenante décide par rapport le module de configuration qui lui a été attribué (Mendonca et al., 2008), (Mendonca et al., 2007), (Czarnecki et al., 2005), (Hubaux. et al., 2010).
La facette *Principe* est décrite par l'attribut prise de décision qualifié par l'ensemble des valeurs suivantes :

- *Prise de décision : ENUM {Décision partagée, Décision individuelle, Les deux}.*

III.3.1.2 Scénario de collaboration

La configuration collaborative peut être réalisée dans différents scénarios selon la nature de l'équipe impliquée (utilisateurs finaux et / ou experts de domaine). Une approche de configuration collaborative peut être conçue pour permettre à un groupe d'utilisateurs finaux d'exprimer de manière collaborative leurs choix par rapport le produit souhaité (par exemple, un groupe d'amis souhaitant acheter une tablette (Stein et al., 2014)). Une approche de configuration collaborative peut aussi être conçue pour l'adopter dans un projet industriel réel (par exemple, la construction automobile). Les différents scénarios de collaboration sont présentés comme suit :

— *Groupe d'utilisateurs finaux* : L'approche de configuration collaborative est conçue pour aider les utilisateurs finaux à exprimer leurs exigences envers le produit désiré (Dou et al., 2016).

— *Chef de produit et experts de domaine* : La configuration collaborative peut être partagée entre le chef du produit qui décide des fonctionnalités du haut niveau et les experts du domaine qui décident des fonctionnalités au niveau technique, comme est le cas dans le contexte industriel (Mendonca et al., 2008).

— *Groupe d'experts du domaine* : Dans ce scénario de collaboration, il n'y a pas de hiérarchie de rôles, les activités de configuration étant partagées équitablement entre les experts du domaine en fonction du niveau d'expertise (Czarnecki et al., 2005).

Dans certains cas, il est possible de trouver des approches qui sont conçues pour supporter à la fois un groupe d'utilisateurs et un groupe d'experts du domaine où les utilisateurs finaux expriment leurs exigences et les experts du domaine considèrent ces choix dans la configuration du produit souhaité (par exemple (Bingliang et al., 2010)).

Les différents scénarios de collaboration présentés ci-dessus sont capturés par les trois attributs booléens suivants :

- *Groupe d'utilisateurs finaux : Booléen.*
- *Chef de produit et experts du domaine : Booléen.*
- *Groupe d'experts du domaine : Booléen.*

III.3.1.3 Satisfaction des parties prenantes

La satisfaction des parties prenantes est d'une importance majeure dans la configuration collaborative d'un produit. Elle peut être ,entre autres, garantie en tenant compte des préférences des parties prenantes dans la configuration du produit souhaité. Pour cette raison les préférences ont été prises en compte différemment dans les approches de configuration collaborative existantes. Par exemple, les préférences peuvent être exprimées sous forme d'exigences strictes et flexibles, comme proposé par (Bagheri et al., 2010), ou sous forme d'exigences fonctionnelles et non fonctionnelles, comme proposé par (Soltani et al., 2012). La satisfaction peut également être assurée lors de la résolution des conflits en trouvant un bon compromis entre les différentes exigences des parties prenantes, comme proposent (Stein et al., 2014), (Ochoa et al., 2015).

La mesure de la satisfaction est capturée par les deux attributs suivants :

- *Mesure de satisfaction : Booléen.*
- *Méthode de mesure : ENUM {Disponible, Indisponible}.*

III.3.2 La dimension Collaboration

Cette dimension permet de définir la collaboration pour la configuration des LPs en termes de mode et niveau de collaboration. Trois facettes constituent la dimension collaboration : niveau de collaboration, mode de collaboration, et interaction.

III.3.2.1 Niveau de collaboration

Selon les approches étudiées, la collaboration peut être présente à deux moments différents : (1) pendant la configuration, où les parties prenantes partagent les activités de configuration (Mendonca et al., 2008), et (2) pendant l'aide à la décision, où des techniques de recommandation basées sur la collaboration (par exemple, le filtrage collaboratif (Schafer et al., 2007)) sont utilisées pour guider les parties prenantes à trouver les produits qui les satisfont en recommandant des caractéristiques (Pereira, 2017). L'aide à la décision est aussi possible durant la résolution des conflits où les parties prenantes peuvent raisonner ensemble sur les alternatives possibles pour la résolution. Il est également possible qu'une approche de configuration englobe les deux niveaux de collaboration.
La facette *niveau de collaboration* est caractérisée avec un attribut type comme suit :

- *Type : ENUM {Pendant la configuration, Pendant l'aide à la décision, Mixte}.*

III.3.2.2 Mode de collaboration

Il existe deux formes de collaboration en terme d'espace : (1) co-localisé où toutes les activités sont exécutées au même endroit comme est le cas pour les approches proposées par (Mendonca et al., 2007), (Stein et al., 2014) et (2) distribué où les activités sont exécutées de manière distribuée dans des endroits différents. Plusieurs approches ont proposé des méthodes différentes pour assurer la cohérence globale d'une configuration géographiquement distribuée, telles que : (Junior et al., 2011), (Holl et al., 2012).
Il existe également deux formes de collaboration en terme de temps. Les activités peuvent être réalisées en temps réel (synchrone) lorsque les parties prenantes configurent le produit en même temps (par exemple (Bingliang et al., 2010)) ou à des moments différents (asynchrone) telle que dans (Hubaux. et al., 2010)) où les états de configuration sont partagés pour assurer la cohérence globale (Abri et al., 2016), (Camarinha-Matos and Afsarmanesh, 2008).
La facette *Mode de collaboration* est caractérisée par les deux attributs temps et espace :

- *Temps : ENUM {Synchrone, Asynchrone.}.*
- *Espace : ENUM {Distribué, Localisé}.*

III.3.2.3 Interaction

L'interaction est nécessaire entre les parties prenantes durant le processus de configuration pour partager certaines informations et assurer la cohérence globale de la configuration. L'amélioration de la cohérence globale est d'un interêt majeur dans le contexte de la configuration des lignes de produits multiples où certaines approches telles que (Rabiser et al., 2010b), (Holl et al., 2012) ont proposé un référentiel partagé qui permet aux parties prenantes de partager leurs décisions afin d'être au courant des décisions des autres parties prenantes. Dans certains cas, les parties prenantes doivent également communiquer lorsqu'un conflit survient (Mendonca et al., 2008). L'interaction peut être assurée avec différents moyens et représentée par l'attribut mode comme suit :

- *Mode : SET (ENUM {Tableau de bord, Communication directe, Partage d'état }).*

III.3.3 La dimension Processus de configuration collaborative

Cette dimension décrit la manière dont le processus de configuration est réalisé. Elle est caractérisée par sept facettes : type de configuration, niveau de configuration, flexibilité, guidage, attribution des rôles, gestion des conflits et l'aide à la décision.

III.3.3.1 Types de configuration

Dans les approches étudiées, différents types de configuration ont été identifiés, notamment : la configuration partielle, la configuration totale et la configuration par étapes. Dans la configuration partielle, chaque partie prenante configure une partie du produit comme dans l'approche (Mendonca et al., 2008). Dans la configuration totale, chaque partie prenante configure le produit entier, les différentes configurations sont ensuite vérifiées et fusionnées en tenant compte des exigences des parties prenantes comme proposée par l'approche (Stein et al., 2014). Dans la configuration par étapes, comme souligné par Czarnecki et al. (Czarnecki et al., 2005), la configuration est raffinée d'une étape à une autre via un ensemble de spécialisations appliquées progressivement sur le modèle de caractéristiques de la ligne.
Le type de configuration est caractérisé par un attribut *type* ayant les valeurs suivantes :

- *Type : ENUM {Partielle, Totale, Par étapes}.*

III.3.3.2 Niveau de configuration

Le processus de configuration peut être réalisé à deux niveaux différents : le niveau intentionnel et opérationnel. Au niveau intentionnel, les parties prenantes expriment leurs exigences concernant le produit souhaité sans être contraintes à une structure de modèle (par exemple, un modèle de caractéristiques). La proposition de (Djebbi and Salinesi, 2007) est parmi les premiers travaux qui proposent une approche de configuration qui n'est pas basée sur un modèle de LP. Dans cette approche, les exigences des parties prenantes sont liées aux exigences de la LP pour dériver un produit consistant tout en respectant les priorités des parties prenantes et les contraintes de la compagnie. Différentes autres approches de configuration collaborative ont également adopté une configuration non basée sur un modèle telles que (Chen et al., 2009),(Dou et al., 2016).
Quant au niveau opérationnel, les exigences sont projetées sur le modèle de la LP pour capturer les caractéristiques qui les satisfont. Dans ce cas, les exigences sont traduites en décisions de sélection/dé-sélection des caractéristiques à partir du modèle de la LP. La plupart des propositions identifiées dans la littérature telles que (Stein et al., 2014), (Mendonca et al., 2008), (Hubaux. et al., 2010),(Rabiser et al., 2009),(Junior et al., 2011),(Pereira, 2017),(Holl et al., 2012) ont proposé des approches de configuration au niveau opérationnel (i.e. basées sur un modèle).
La facette niveau de configuration est caractérisée par un attribut *type*, comme suit :

- *Type : ENUM {Non basée sur un modèle, Basée sur un modèle}.*

III.3.3.3 Ordre de sélection

La flexibilité est reconnue comme une des dimensions les plus importantes d'une stratégie de fabrication réussie. (de Groote, 1994) souligne que "une technologie particulière est dite plus flexible qu'une autre si une augmentation de la diversité de l'environnement engendre un changement de performance plus souhaitable (c'est-à-dire une augmentation plus importante ou une diminution plus faible) que le changement qu'on obtiendrait avec l'autre technologie dans les mêmes conditions".
Dans un processus de configuration flexible tel que dans les approches (Junior et al., 2011) et (Stein et al., 2014), les parties prenantes peuvent exprimer librement leurs exigences sans être contraintes à des décisions antérieures. Quand la configuration repose sur un processus à ordre prédéfini tel que dans les approches (Mendonca et al., 2007) et (Mendonca et al., 2008), les décisions prises par certaines parties prenantes limitent les décisions prises ultérieurement par d'autres. Dans un tel cas, il est parfois nécessaire de revenir en arrière pour réviser certaines décisions ceci rend l'obtention d'une configuration qui satisfait toutes les parties prenantes difficile (Stein et al., 2014).

La facette est représentée par un attribut *flexibilité* ayant les valeurs suivantes :

- *Flexibilité : ENUM {Sans ordre, Ordre prédéfini}.*

III.3.3.4 Guidage

Le processus de configuration peut être guidé différemment selon le modèle de la LP et les besoins des parties prenantes. Généralement la configuration est guidée par le modèle de la LP qui est divisé en plusieurs modules (également appelé vues). Dans ce cas, chaque partie prenante configure un module tout en respectant les contraintes du domaine (Hubaux. et al., 2010). Dans certains cas, la configuration peut être guidée par un modèle de processus décrivant le processus de configuration en termes d'attribution de rôle, ordre de configuration et schéma de priorité tel qu'il est le cas dans l'approche (Mendonca et al., 2007). De plus, le processus de configuration peut être guidé par les exigences des parties prenantes indépendamment de la hiérarchie du modèle de la LP en planifiant les différentes activités de configuration en fonction de l'importance des exigences.
La facette guidage est caractérisée par l'attribut suivant :

- *Guidage : SET (ENUM {Modèle de caractéristiques, Modèle de processus, Exigences des utilisateurs}).*

III.3.3.5 Attribution des rôles

Le processus d'attribution des rôles prend en compte deux critères : l'autorité et la connaissance du domaine (Mendonca et al., 2008), (Mendonca et al., 2007), (Czarnecki et al., 2005). Ainsi, le chef du produit configure le produit en haut niveau, puis les autres rôles sont attribués en fonction de l'expertise des parties prenantes (Mendonca et al., 2007). Une attribution aléatoire est également possible dans certains cas, en fonction de la nature des personnes impliquées (par exemple, un prototype académique testé par un groupe d'étudiants ou un outil de configuration en ligne commercial utilisé par un groupe de clients). La facette attribution des rôles est caractérisée par l'attribut *rôle* comme suit :

- *Rôle : ENUM {Basé sur l'expertise, Basé sur l'autorité, Aléatoire}.*

III.3.3.6 Gestion des conflits

Dans la configuration collaborative, les causes de conflit vont des exigences hétérogènes à des points de vue divergents sur la manière dont le produit souhaité devrait être. Ces conflits sont gérés par plusieurs *méthodes de résolution* et selon différentes *politiques*. La facette gestion des conflits permet de comprendre comment

la résolution peut être réalisée. Elle est caractérisée par les attributs suivants : *résolution basée sur les préférences, décidabilité, méthode de résolution* et *politiques de résolution.*

Une stratégie de résolution des conflits ne peut pas être considérée comme efficace si elle ne tient pas compte des *préférences* des différentes parties prenantes et ne tente pas de trouver un compromis entre entre elles (Stein et al., 2014). En outre, la stratégie de résolution doit être également *décidable* dans le sens où elle doit garantir un état fini où les conflits sont toujours résolus (Ochoa et al., 2015). La nature de la stratégie de résolution est caractérisée par les attributs suivants :

- *Résolution basée sur les préférences : Booléen.*
- *Décidabilité : Booléen.*

Différentes méthodes de résolution ont été proposées pour la configuration collaborative des LPs. Certains travaux tels que (Chen et al., 2009) résolvent les conflits en invitant les parties prenantes à *négocier.* Autres travaux tels que (Mendonca et al., 2008) proposent une méthode de résolution des conflits par *session de fusion.* Durant cette session, les experts du domaine responsables des décisions conflictuelles raisonnent sur les solutions du conflit. La fusion est nécessaire si deux ou plusieurs sessions interdépendantes sont configurées en parallèle et contiennent des décisions qui, ensemble, violent certaines contraintes. Dans le cas contraire, les conflits sont résolus en donnant la priorité au choix de configuration exprimé en premier dans le workflow de configuration. D'autres travaux tels que (Mendonca et al., 2007), proposent un *schéma de priorité prédéfini* où les conflits sont résolus selon l'importance du rôle des parties prenantes ou l'importance de la décision. Autres méthodes de résolution ont été proposées où un ensemble de valeurs de correction est soit directement suggéré aux parties prenantes (Rabiser et al., 2009), (Xiong et al., 2012), soit recommandé en utilisant une *métaphore d'assistant personnel* (Junior et al., 2011). D'autres approches s'interessent à la satisfaction des parties prenantes proposent des méthodes de résolution des conflits basées sur les préférences. Dans ces méthodes, les préférences sont considérées de différentes manières : (1) exprimées à travers des contraintes flexibles et strictes (Stein et al., 2014), pendant la résolution, les contraintes strictes sont atténuées en leur attribuant des degrés de préférence et en gardant la décision avec la préférence supérieure, (2) en utilisant *critères basés sur la langue* comme proposé dans (Ochoa et al., 2015) pour considérer les préférences des parties prenantes exprimées en termes de propriétés non fonctionnelles.

Les différentes méthodes de résolution collaborative présentées ci-dessus sont représentées par un ensemble de valeurs caractérisant la *méthode de résolution* comme suit :

- *Méthode de résolution* : SET (ENUM {Négociation, Schémas de priorité pré-

définis, Session de fusion, Valeurs de correction, Métaphore de l'assistant personnel, Atténuation de la contrainte stricte, Résolution de critères basés sur la langue}).

Ces méthodes de résolution peuvent être exécutées selon différentes politiques : (1) les conflits sont *manuellement* résolus lorsque les parties prenantes sont impliquées dans le processus de résolution soit en révisant leurs choix de configuration, soit en exprimant leurs préférences par rapport les exigences conflictuelles, (2) les conflits sont *automatiquement* résolus en tenant compte d'un ensemble de critères tels que l'importance du rôle des parties prenantes et l'importance des exigences. Dans certains cas, il est possible de combiner les deux *politiques*, où les parties prenantes expriment leurs préférences par rapport aux exigences conflictuelles, ensuite un processus de résolution est automatiquement lancé.
Les politiques de résolution sont caractérisées par l'attribut suivant :

- *Politiques de résolution : ENUM {Automatique, Manuelle, Mixte}.*

III.3.3.7 Aide à la décision

Les techniques de guidage et de recommandation sont souvent utiles pour aider les parties prenantes dans la prise des décisions à différents niveaux. Le guidage peut être assuré par différentes méthodes. (Junior et al., 2011) et (Afonso et al., 2020) proposent une méthode d'assistance pour les parties prenantes fournie par des agents logiciels (agent mobile et chatbots) pour les guider durant le processus de configuration. En outre, les techniques de recommandation peuvent être également un moyen de guidage efficace : soit *durant la configuration*, où des caractéristiques peuvent être recommandées pour aider les parties prenantes à satisfaire leurs exigences (Pereira, 2017), soit *durant un conflit*, où des scénarios alternatifs de sélection de caractéristiques peuvent être recommandés aux parties prenantes pour les guider dans la résolution des conflits (Xiong et al., 2012).
La facette aide à la décision est caractérisée par les attributs suivants :

- *Recommandation : Booléen.*
- *Guidage : Booléen.*
- *Niveau de support : ENUM {Durant la configuration, Durant un conflit, Mixte, Indisponible}.*

III.3.4 La dimension Outil

La dernière dimension du cadre PL2C est la dimension outil. Un outil, ou une combinaison d'outils, est un moyen nécessaire pour exécuter les différentes tâches

de l'approche concernée.
Selon les approches étudiées, certaines ne sont pas supportées par un outil de configuration telles que les approches de (Czarnecki et al., 2005), (Mendonca et al., 2007). D'autres proposent un outil qui supporte une étape de la configuration collaborative : par exemple un outil de décomposition du modèle de la LP et d'attribution de rôles comme (Mendonca et al., 2008). De manière générale, la plupart des outils développés supportent principalement l'étape de configuration (Holl et al., 2012), (Soltani et al., 2012). Certains outils sont encore au niveau *expérimental* (Junior et al., 2011), (Stein et al., 2014), d'autres sont validés dans un cadre d'étude de cas *industriel* et déployés dans des entreprises telles que les approches de (Hubaux. et al., 2010), (Rabiser et al., 2010b).
Cette dimension est caractérisée par une facette outil ayant les trois attributs suivants :

- *Supportée par outil : Booléen.*
- *Couverture : SET (ENUM {Attribution des rôles, Configuration, Résolution des conflits, Toutes les étapes, Indisponible}).*
- *Version : ENUM {Expérimentale, Industrielle}.*

III.4 Synthèse

Le positionnement des différentes approches dans le cadre PL2C permet de les comprendre et facilite leur comparaison. Selon le positionnement des 41 études collectées [4], les travaux peuvent être classés comme suit :

- Études qui proposent des approches collaboratives basées sur un processus prédéfini. Durant ce processus, les tâches de configuration sont coordonnées et assignées à des groupes d'experts où chacun configure un module spécifique (appelé view dans certains articles) du modèle de la LP, comme [S1, S2, S3, S4, S5, S6, S8, S11, S14, S20, S25, S40]. Certaines de ces approches [S4, S5, S8] permettent de partager la configuration lorsqu'un module est configuré par une équipe au lieu d'un seul expert.
- Études qui proposent des approches collaboratives basées sur un processus flexible où aucun ordre de configuration n'est imposé sur les parties prenantes [S17, S26, S27, S32, S33, S34, S35, S37].
- Études proposant des approches collaboratives qui ne fournissent pas des informations explicites sur les aspects de la collaboration et l'ordre de configuration [S7, S9, S10, S12, S13, S15, S16, S18, S19, S21, S22, S23, S24, S28, S29, S30,

4. L'analyse complète est disponible via le lien suivant : `https://bit.ly/31TfgE2` (Feuille 3)

S31, S36]. Chaque groupe de ces études concentre sur une des principales caractéristiques de la configuration collaborative. Par exemple, [S12, S16, S19, S21, S22, S31] proposent des méthodes pour assurer la cohérence globale durant la configuration des LPs multiples. Les approches proposées dans les études [S7, S9, S10, S13, S15, S18, S23, S24, S28, S29, S30, S36], offrent différentes solutions qui permettent d'obtenir une configuration valide en prenant compte des exigences fonctionnelles et non fonctionnelles des parties prenantes.

- Études qui considèrent la collaboration à un niveau d'aide à la décision [S38, S39, S41] où des techniques basées sur la collaboration sont utilisées pour guider les utilisateurs dans le choix des caractéristiques pendant le processus de configuration. Dans ces approches, l'aspect collaboratif consiste à étudier les caractéristiques pertinentes des configurations précédentes et s'en servir pour guider les futurs utilisateurs dans leur choix.

Selon l'analyse rapportée, 14 études sur 41 ont traité la satisfaction des parties prenantes dans les solutions qu'elles proposent [S9, S10, S13, S15, S18, S23, S24, S27, S28, S29, S32, S34, S36, S37]. Ces études proposent l'amélioration de la satisfaction des parties prenantes de deux manières différentes : (1) en considérant les exigences fonctionnelles et non fonctionnelles des parties prenantes dans la configuration finale dérivée [S9, S10, S13, S15, S18, S23, S24, S28, S29, S34, S36].(2) en augmentant la satisfaction des parties prenantes grâce à l'adoption des stratégies de résolution des conflits basées sur les préférences [S27, S32, S37].
Cependant, 17 études parmi les 27 autres études ne fournissent pas d'informations sur la gestion des conflits, à l'exception de [S2, S3, S4, S5, S7, S8, S11, S17, S22, S26] qui proposent des stratégies de résolution qui ne tiennent pas compte des préférences des parties prenantes.
De manière générale, la plupart des approches s'intéressent soit à l'analyse des exigences des parties prenantes pour dériver une configuration cohérente, ou à l'assurance de coordination durant le processus de configuration. La collaboration a été généralement traitée dans les deux premières étapes du processus de configuration collaborative (attribution des rôles, configuration) que dans la gestion des conflits (méthode ou politique de résolution).

III.5 Les défis de la configuration collaborative

Trois défis ont été identifiés de l'analyse complète des 41 études (Edded et al., 2019) :

1. *Coordination du processus de configuration :* la coordination présente un des principaux défis de la configuration collaborative car elle est responsable de

la cohérence de l'état global de la configuration. La coordination peut être assurée avec deux méthodes : *(1)* en suivant un processus de configuration prédéfini qui précisent le rôle et le moment de configuration de chaque partie prenante (Mendonca et al., 2008). Toutefois, cette méthode limite les décisions de certaines parties prenantes par manque de flexibilité (Stein et al., 2014). (2) en permettant aux parties prenantes de configurer un produit de manière libre et mettre en place des moyens de coordination pour partager les états entre les parties prenante et assurer la cohérence globale (Holl et al., 2012).

2. *Gestion des conflits :* consiste à trouver des stratégies efficaces de résolution des conflits de configuration. La gestion de conflit est une phase cruciale dans la configuration collaborative car les choix des différentes parties prenantes ont souvent tendance à diverger. Différentes stratégies peuvent être adoptées pour résoudre les conflits.(1)*résolution systématique* consiste à privilégier chaque fois les choix faits à un stade précoce dans le processus de configuration (Mendonca et al., 2007), (2)*résolution automatique* consiste à analyser les choix de toutes les parties prenantes et retourner une configuration cohérente indépendamment des préférences exprimées (Dou et al., 2016), (3)*résolution manuelle* assurée par la négociation des alternatives de résolution possibles (Mendonca et al., 2008), (4)*résolution basée sur les préférences* consiste à considérer les exigences strictes/souples et fonctionnelles/non fonctionnelles des parties prenantes dans le processus de résolution (Ochoa et al., 2015).
   Cependant, la résolution des conflits reste un défi majeur quelque soit la stratégie adoptée, car elle dépend de la manière dont le compromis mis en place respecte les différents choix contradictoires des parties prenantes.

3. *Satisfaction des parties prenantes :* la satisfaction des parties prenantes est un des critères clés qui mesure l'efficacité d'un processus de configuration collaborative (Stein et al., 2014). Car le produit dérivé doit être à la fois valide et respecte la plupart des choix des parties prenantes. Pour les choix non retenus, ils doivent être décidés par un processus de résolution qui tient compte des préférences des parties prenantes pour pouvoir assurer leur satisfaction vis à vis la stratégie de résolution. Ceci rend la satisfaction un défi encore difficile à assurer dans certaines situations. Notamment dans le cas de la contradiction totale de tous les choix des parties prenantes ou lorsque ces dernières partagent la totalité de l'espace de configuration sans ordre de passage prédéfini.

III.6 Conclusion

L'étude systématique présentée dans ce chapitre a permis d'explorer davantage le domaine et de montrer qu'aucune étude complète n'a été proposée jusqu'à présent pour comparer et comprendre les approches de la configuration collaborative existantes. Le résultat de cette étude a été utilisé pour conceptualiser un cadre de référence pour l'analyse des travaux dans le domaine de la configuration collaborative des LPs. Le cadre de référence nommé "PL2C" permet de : (1) saisir l'état actuel de la configuration collaborative grâce à la notion de dimensions et de facettes et faciliter l'identification des caractéristiques des différentes approches existantes, (2) obtenir une vue d'ensemble de la configuration collaborative des LPs et comprendre comment la configuration collaborative peut être effectuée et (3) aider à identifier les défis soulevés dans la littérature.

IV

Aperçu de l'approche Colla-Config

Sommaire

IV.1 Introduction **52**

IV.2 Principe de la configuration collaborative **52**

IV.3 Vue globale de l'approche Colla-Config **54**

IV.4 Exemple d'application illustratif **60**

IV.5 Caractéristiques de l'approche Colla-Config **64**

IV.6 Conclusion **66**

IV.1 Introduction

L'analyse de l'état de l'art (cf. Chapitre 3) a fait ressortir les problèmes énoncés dans cette thèse, à savoir : (1) le pouvoir d'expression des exigences de configuration est limité, (2) le processus de configuration est basé sur un ordre prédéfini et (3) les stratégies de résolution des conflits sont systématiques et ne considèrent pas les préférences des parties prenantes.
Ce chapitre présente un aperçu de l'approche de configuration Colla-Config (Collaborative Configuration) proposée dans cette thèse pour la configuration collaborative des lignes de produits et qui a pour objectif de résoudre ces problèmes.

IV.2 Principe de la configuration collaborative

Cette section explique le principe de la configuration collaboratives des LPs à travers la présentation de son cadre méthodologie et l'illustration d'une mise en situation.

IV.2.1 Cadre méthodologique de la configuration collaborative

L'approche Colla-Config s'inscrit dans la famille des approches de configuration collaborative des lignes de produits (Mendonca et al., 2008). Dans ce type d'approches, le processus de configuration est partagé entre plusieurs parties prenantes (chef de produit, ingénieur, logistique, client, etc.) qui décident la spécification du produit désiré en exprimant leurs exigences de configuration. Comme défini par(Djebbi, 2011), une exigence de configuration est une exigence qui a pour conséquence de réduire le nombre de combinaisons possibles définissant la LPs afin d'aboutir à une seule combinaison satisfaisante. Par conséquent, l'expression des exigences peut être aperçue comme étant un processus de spécification des *choix de configuration* en précisant à travers le modèle de la LP les caractéristiques souhaitées et non souhaitées.
Comme schématisé par la Figure IV.1, le processus de configuration collaborative nécessite deux entrées et retourne une sortie comme suit :
- *Les entrées du processus* : ce sont l'ensemble des choix de configuration exprimés par les différentes parties prenantes ainsi que la spécification du modèle de la ligne de produits à configurer qui sera considérée comme l'ensemble des éléments potentiels du produit final.

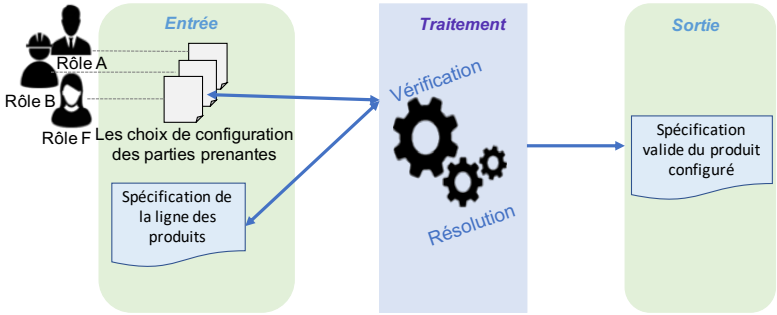


FIGURE IV.1 – Processus de configuration collaborative d'une LP.

- *La sortie du processus* : c'est la spécification valide du produit désiré. La construction de ce produit doit prendre en compte les choix de configuration des différentes parties prenantes représentant leurs attentes vis-à-vis du produit final.
Les choix de configuration des différents parties prenantes passent par un processus de traitement qui consiste principalement en deux étapes qui sont la vérification et la résolution :

- *Vérification* : durant cette étape, les choix des configuration sont fusionnés et analysés pour assurer que la configuration obtenue respecte bien les contraintes des modèles du domaine de la LP.
- *Résolution* : pendant cette étape et après avoir vérifié sa consistance, la configuration est retournée comme spécification finale du produit désiré. Dans le cas d'inconsistance, les choix de configuration sont révisés pour résoudre le(s) conflit(s) détecté(s) et corriger la configuration.
  Afin d'aboutir à une configuration valide, l'étape de résolution consiste à trouver un compromis entre les choix de configuration sources de(s) conflit(s). D'où, l'étape de résolution doit se baser sur une stratégie qui maximise le nombre de choix respectés et justifie la suppression de ceux non pris en considération.

L'approche Colla-Config permet la configuration collaborative d'un produit à partir d'un modèle de LP et d'un ensemble de choix de configuration exprimés par un groupe de parties prenantes, tout en considérant leurs préférences dans l'étape de résolution des conflits.

IV.2.2 Mise en situation

Durant la configuration collaborative, chacune des parties prenantes exprime ses choix de configuration en interagissant avec le modèle de la LP à travers les caractéristiques qui le constituent. Étant donnée la nature collaborative du processus, des situations de désaccord peuvent survenir. Ces situations sont dues au fait que les choix de configuration d'une ou plusieurs parties prenantes ne respectent pas les

contraintes du modèle de la LP.
Comme illustré dans le schéma de la Figure IV.2, ces situations peuvent survenir dans deux moments différents : - *(1) :* Quand la partie prenante exprime ses choix

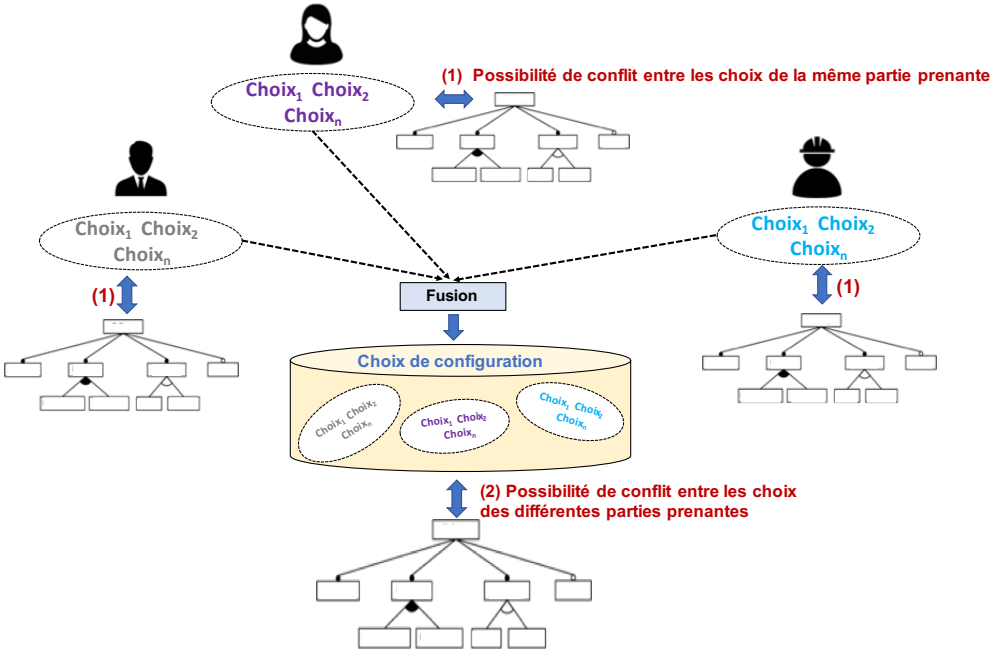


FIGURE IV.2 – Illustration de la configuration collaborative d'une LP.

de configuration contradictoires : par exemple, la partie prenante choisit la même caractéristique à la fois dans la liste des caractéristiques désirées et la liste de celles non désirées. Ces choix peuvent également violer les contraintes du modèle/ domaine de la LP.
- *(2) :* Comme schématisé par la Figure IV.2, au moment de la fusion de tous les choix de configuration des différentes parties prenantes, les situations de désaccord peuvent survenir à cause de : (1) la contradiction de deux ou plusieurs de ces choix, ou (2) la violation des contraintes du modèle de la LP.
Dans le contexte de la configuration collaborative des LPs, ces situations de désaccord sont communément appelées conflits. La notion et la typologie des conflits sont présentées dans la Section 3.2.1 de ce chapitre.

IV.3 Vue globale de l'approche Colla-Config

Colla-Config est une approche qui permet une configuration flexible où les parties prenantes peuvent librement exprimer leurs choix de configuration sans leur imposer un ordre de passage défini. Elle permet une résolution de conflits dynamique basée sur les préférences des parties prenantes. Ces préférences sont exprimées à travers

un ensemble prédéfini de scénarios de configuration alternatif appelés règles de substitution.
La Figure IV.3 donne une vue globale de l'approche Colla-Config. Elle consiste en trois sous- processus : (1) configuration, (2) vérification, (3) et la résolution qui sont détaillés dans les sections suivantes.

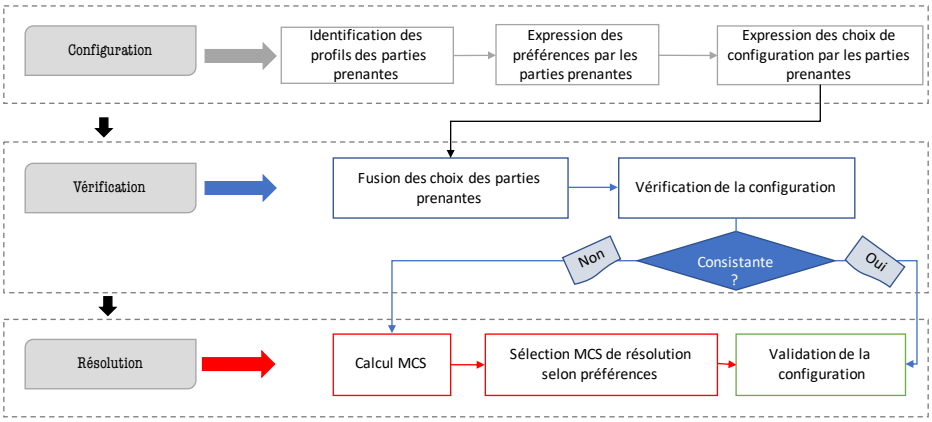


FIGURE IV.3 – L'approche Colla-Config

IV.3.1 La configuration

Configurer un produit à partir d'un modèle de LP d'une manière collaborative nécessite l'identification du groupe des parties prenantes participantes au processus de configuration.
Les parties prenantes sont identifiées par un ensemble d'informations représentées par des profils. (Gauch et al., 2003) a définit le profil comme « toutes les variations qui caractérisent un utilisateur ou un groupe d'utilisateurs, peuvent se regrouper sous le terme de profil utilisateur ». Plusieurs autres définitions ont été identifiées dans la littérature qui permettent de définir le profil selon le contexte d'utilisation telles que (Tamine-Lechani and Bahsoun, 2006), (Kostadinov, 2007) , (Daoud et al., 2008).
Dans le contexte de Colla-Config, un profil est un ensemble de données caractérisant chaque partie prenante. Ces données représentent les informations personnelles et les préférences de différentes parties prenantes qui serviront principalement à la résolution des conflits détectés.
Le profil d'une partie prenante contient les informations suivantes :

- Informations personnelles : notamment le nom et le prénom.
- Le statut : représente le niveau d'expertise de la partie prenante (expert ou novice) dans la configuration des LPs.

- Le rôle : représente le rôle qu'occupe la partie prenante durant le processus de configuration (chef de produit, ingénieur, client, utilisateur final).
- Préférences : présentent des scénarios de configuration alternatifs que la partie prenante souhaite avoir si ses choix de configuration initiaux ne peuvent pas être totalement/ partiellement retenus à cause d'une situation de conflit.

Après avoir identifié les profils à travers différentes informations (nom, prénom, statut et rôle), les parties prenantes expriment en une deuxième étape leurs préférences par rapport le produit souhaité (2ème étape du processus de configuration Figure IV.3). Ces préférences sont représentées sous forme d'un ensemble de règles de substitution. Les règles de substitution représentent des scénarios de configuration alternatives à considérer si les choix de configuration initiaux des parties prenantes ne peuvent pas être retenus.
Par exemple, une partie prenante peut exprimer qu'elle préfère avoir *le produit le plus complet*. Un produit complet est un produit qui inclut le maximum possible de l'ensemble total des caractéristiques sélectionnées par les différentes parties prenantes. Un autre exemple de règle est qu'une partie prenante peut exprimer qu'elle préfère *le produit le plus simple*. Un produit simple est un produit qui inclut le minimum possible de l'ensemble total des caractéristiques sélectionnées par les différentes parties prenantes. La liste totale des règles de substitution est présentée en détail dans le chapitre suivant.
Après avoir exprimé leurs préférences, les parties prenantes expriment les choix de configuration sur le modèle de la LP.

IV.3.2 La vérification de la configuration

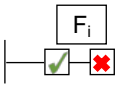
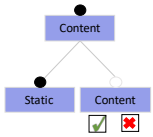
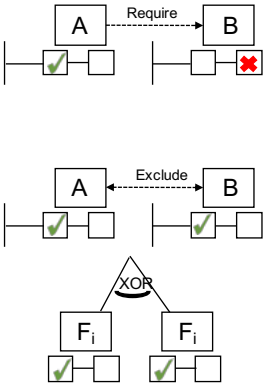
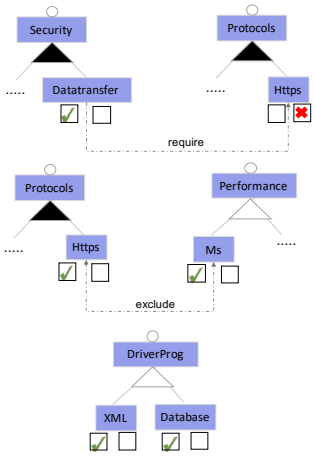
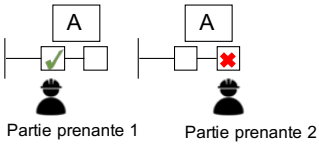
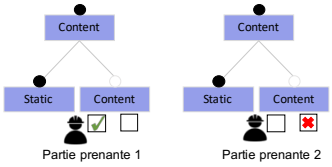
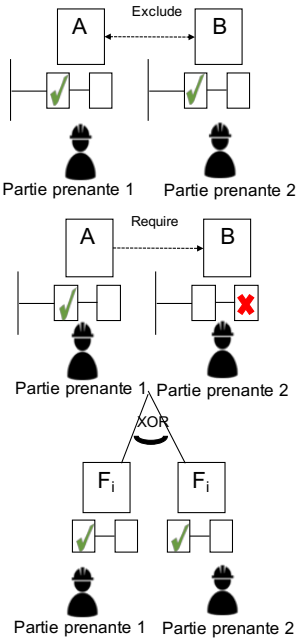
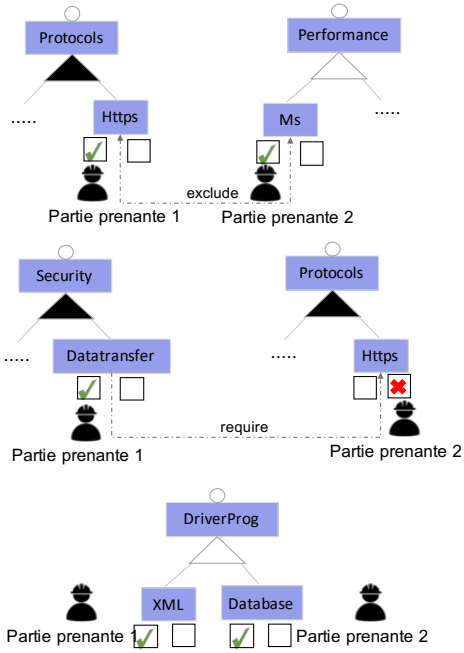
Les choix de configuration exprimés par les différentes parties prenantes passent par un processus de vérification afin de détecter les conflits.

IV.3.2.1 La notion de conflit

Les conflits sont des situations de désaccord où les choix exprimés ne peuvent pas être retenus ensemble dans la configuration finale. De manière générale, il est possible de catégoriser les conflits sous quatre types différents d'inconsistance (Edded et al., 2020) comme illustré dans le Tableau IV.1. Pour chaque type d'inconsistance, un exemple concret de situation est illustré.
Les deux premiers types d'inconsistance concernent les choix de configuration de la même partie prenante :

- ***Incohérence :*** elle est *explicite* et représente le cas où une partie prenante se contredit à elle-même.

Table IV.1 – Types d'inconsistance

Type d'inconsistance		Illustration	Exemple
Incohérence			
Non conformité			
Conflits	Explicite		
	Implicite		

- ***Non conformité :*** elle est *implicite* et représente le cas où les choix d'une même partie prenante ne respectent pas les contraintes du modèle/domaine de la LP.

Les deux autres types d'inconsistance concernent les choix de configuration des dif-

férentes parties prenantes :

- ***Conflit explicite :*** représente le cas où les choix des parties prenantes par rapport la même caractéristique sont contradictoires (une caractéristique désirée par une partie prenante et non désirée par une autre).
- ***Conflit implicite :*** représente le cas où les choix des parties prenantes ne respectent pas les contraintes du modèle de la LP alors que les choix de chaque partie sont consistants individuellement.

Dans le cas des modèles de LP étendus (Mazo, 2014) c'est à dire les modèles avec attributs (les attributs permettent d'associer davantage d'informations aux caractéristiques) ou modèles avec cardinalité, il peut y avoir de désaccord sur les attributs ou les cardinalités. Ces désaccords peuvent être également représentés par un des types de conflits présentés ci-dessus :

— *Désaccord sur la présence d'un attribut :* Il s'agit d'un conflit explicite car ça représente le cas où un attribut est désiré par une partie prenante et non désiré par une autre.

— *Désaccord sur la valeur d'un attribut :* Il s'agit d'un conflit implicite car ça représente le cas où les choix des parties prenantes ne respectent pas les contraintes du modèle. La contrainte c'est le Ou exclusif (XOR dans le TableauIV.1) entre les valeurs que peut prendre l'attribut en question.

— *Désaccord sur la cardinalité d'une caractéristique :* Il s'agit d'un conflit explicite car ça porte sur le nombre d'une même caractéristique (e.g. le nombre de portes dans une voiture).

IV.3.2.2 Le principe de vérification

La fusion consiste à regrouper les choix de configuration des différentes parties prenantes sous forme d'une seule configuration tout en éliminant les choix de configuration dupliqués : si deux ou plusieurs parties prenantes expriment le même choix de configuration ceci sera considéré une seule fois.
Comme schématisé dans la Figure IV.3, la deuxième étape de la vérification consiste à vérifier la consistance de la configuration obtenue pour assurer que les différents choix de configuration sont cohérents et respectent les contraintes du modèle et du domaine de la LP. Si la configuration est consistante, alors elle sera validée et retournée aux parties prenantes comme spécification du produit final. Dans le cas contraire, un processus de résolution des conflits est exécuté.

IV.3.3 La résolution des conflits

La résolution consiste à résoudre les conflits source d'inconsistance de la configuration. La stratégie de résolution consiste à identifier les différentes combinaisons minimales de choix de configuration conflictuels dont la suppression résout tous les conflits détectés.

IV.3.3.1 Le sous-ensemble de correction minimal (MCS)

Le concept de sous-ensemble de correction minimal (Minimal Correction Subset (MCS) en anglais) a été initialement proposé par (Liffiton and Sakallah, 2008) qui l'a défini comme un sous-ensemble irréductible de contraintes dont la suppression rend satisfiable un programme de contraintes.
Le principe de calcul de MCSs a été adopté dans Colla-Config pour calculer le nombre de choix de configuration minimal à supprimer pour résoudre les conflits détectés. L'algorithme de calcul de MCSs est détaillé dans le chapitre suivant.

IV.3.3.2 Le principe de résolution

Le processus de résolution proposé dans Colla-Config consiste à calculer la liste des MCSs pour garantir la correction de la configuration avec un minimum de choix à éliminer. Une fois la liste des MCSs calculée, il faut choisir en une deuxième étape un MCS pour la résolution de(s) conflit(s) détecté(s) selon les préférences des parties prenantes.
Comme précisé précédemment, les préférences sont exprimées à travers un ensemble de règles de substitution proposant des scénarios de configuration alternatifs. Chaque règle correspond à la suppression d'un MCS bien précis. Par exemple, pour une règle qui permet d'avoir le produit le plus complet, le MCS correspondant est celui qui élimine le minimum de caractéristiques sélectionnées par les parties prenantes.
Afin de déterminer quel MCS retenir pour résoudre le conflit, les différentes règles de substitution sélectionnées sont appliquées une par une sur la liste des MCSs calculée. Selon le résultat d'application de ces règles sur la liste des MCSs, deux situations sont initialement possibles :

1. Un MCS commun : dans ce cas, la liste des règles de substitution sélectionnées par les parties prenantes retourne le même MCS comme résultat. Les choix qu'il contient sont simplement retirés de la configuration.
2. Aucun MCS commun : dans ce cas, l'application de la liste des règles de substitution sélectionnées par les parties prenantes donne des MCSs différents. En conséquence, une règle supplémentaire est appliquée. Cette règle permet au chef de produit de choisir une partie prenante comme prioritaire. Finalement,

le MCS qui respecte les choix de la partie prenante priorisée est choisi pour résoudre les conflits.

Une fois le conflit résolu, la configuration sera validée et retournée comme spécification finale aux parties prenantes. Un exemple qui illustre le principe de l'approche Colla-Config est présenté dans la section suivante.

IV.4 Exemple d'application illustratif

L'exemple de la LP choisi pour illustrer l'application de Colla-Config est une version adaptée du modèle de la ligne des portails web proposé par (Mendonca et al., 2008). Le modèle présenté dans la Figure IV.4 est spécifié au moyen de la notation FODA(Feature Oriented Domain Analysis)(Kang et al., 1990) (cf. chapitre 2). Comme illustré dans la Figure IV.4, chaque portail web possède obligatoirement un serveur web « Webserver». Un portail web peut aussi avoir une ou plusieurs caractéristiques optionnelles parmi les suivantes : « AdditionalServices», « Persistance», « Security» et « Performance». Chacune de ces caractéristiques possède des sous-caractéristiques qui sont liées avec des cardinalités de groupe de type [1..1] qui correspond au OU exclusif (XOR), de type [1..N] qui correspond au OU inclusif(OR) ou bien de type [0..N] qui correspond au ET. Par exemple, « XML » et « Database» sont liées avec un XOR, ce qui signifie qu'une seule option parmi les deux peut être sélectionnée dans une configuration. Les caractéristiques « Nntp», « Ftp» et « Https» sont liées avec un OR, ce qui signifie qu'au moins une option doit être sélectionnée. « SiteStatistics», « SiteSearch» et « AdServer» sont liées avec un ET, ce qui signifie qu'aucune ou plusieurs options peuvent être sélectionnées.
Le modèle des portails web contient également un ensemble de contraintes de deux types qui sont : (1) "exclude" tel qu'il est le cas entre les options « Https» et « Ms» ce qui signifie que ces deux options ne peuvent pas être sélectionnées dans la même configuration, et (2) "require" comme entre les options « Datatransfer» et « Https» ce qui signifie que la sélection de Datatransfer exige la sélection de Https.
Pour illustrer la configuration de la ligne des portails web avec l'approche Colla-Config, un scénario de configuration impliquant trois parties prenantes est détaillé dans les sections suivantes.

IV.4.1 La configuration de la ligne des portails web

Après avoir choisi les informations personnelles (nom et prénom, statut et rôle), chaque partie prenante exprime ses préférences en sélectionnant une ou plusieurs règles de substitution. Comme indiqué dans le Tableau IV.2, les trois parties pre-

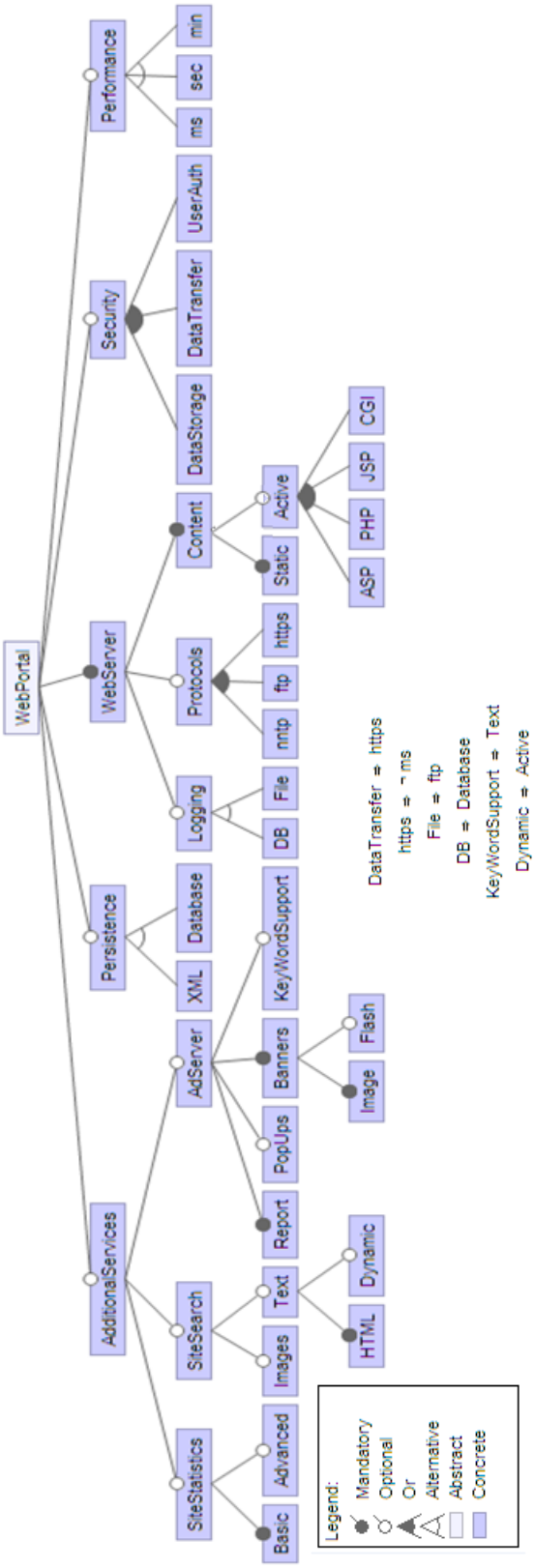


FIGURE IV.4 – Le modèle de caractéristiques des portails web (Mendonca et al., 2008).

nantes (PP1, PP2 et PP3) ont choisi la règle R1 : "*le produit le plus complet*", qui correspond au MCS qui élimine le minimum de caractéristiques du produit configuré. PP2 a également choisi la règle R2 : "*le produit le plus simple*", qui correspond au MCS qui élimine le maximum de caractéristiques du produit configuré.

TABLE IV.2 – La sélection des règles

Partie prenante	**Règle sélectionnée**
PP1	R1 : le produit le plus complet
PP2	R1 : le produit le plus complet R2 : le produit le plus simple
PP3	R1 : le produit le plus complet

À l'étape suivante, chaque partie prenante spécifie ses choix de configuration. Dans l'approche Colla-Config, les caractéristiques non sélectionnées sont aussi considérées comme des décisions indispensables. Par conséquent, chaque partie prenante doit spécifier les caractéristiques désirées et non désirées.
Comme illustré dans le Tableau IV.3, PP1 a sélectionné la caractéristique « KeywordSupport». Cette sélection nécessite la sélection automatique de la caractéristique « Text» vu la contrainte *require* entre ces deux caractéristiques (Figure IV.4). PP1 a aussi sélectionné la caractéristiques «DB» qui induit à la sélection automatique de la caractéristique « Database» vu la contrainte *require* entre ces deux caractéristiques.
PP2 a sélectionné la caractéristique «XML» et déslectionné la caractéristique «Text» (exprimée sous forme ¬Text dans le Tableau IV.3).
Enfin, PP3 a choisi un contenu dynamique en sélectionnant la caractéristique « Active» et sa sous-caractéristique « PHP ».

IV.4.2 La vérification de la configuration

Les différents choix de configuration sont fusionnés et vérifiés par rapport aux contraintes du modèle. Après vérification de la configuration obtenue, 2 conflits sont détectés.
Comme illustré dans le Tableau IV.4, le premier conflit est causé par la contradiction des choix de configuration portant sur la caractéristique «Text» qui est exigée par la caractéristique « KeywordSupport» et exprimée comme non désirée par PP2 (« ¬ Text»). Le deuxième conflit est dû à la violation de la contrainte "*XOR*" vu que PP2 a sélectionné «XML» alors que « Database» a été aussi sélectionnée.

Table IV.3 – Les choix de configuration

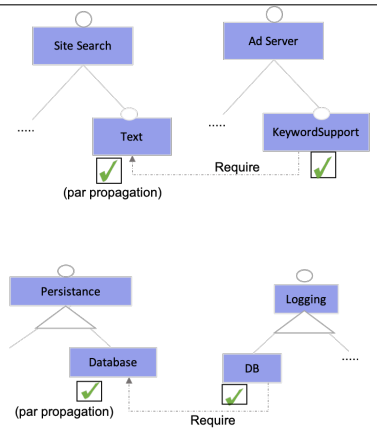
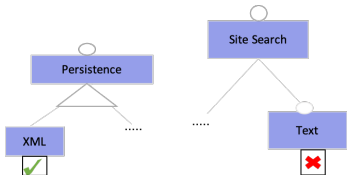
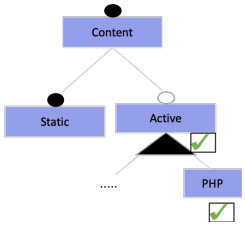
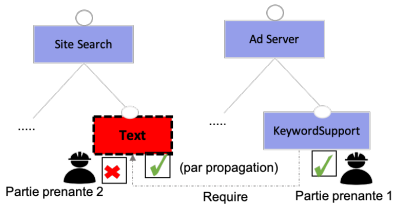
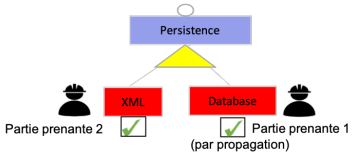
Partie prenante	Choix de configuration	Illustration
PP1	KeywordSupport, Text (par propagation de contrainte), DB, Database (par propagation de contrainte)	
PP2	XML, ¬ Text	
PP3	Active, PHP	

Table IV.4 – La liste des conflits

Conflit	Illustration
(Text VS ¬ Text)	
(Xml VS Database)	

IV.4.3 La résolution des conflits

Pour résoudre ces conflits, la liste de MCSs est d'abord calculée pour déterminer les combinaisons de choix possibles dont la suppression résout ces 2 conflits. Comme présenté dans le Tableau IV.5, la liste calculée comporte six MCSs.

Ensuite, les règles choisies sont appliquées sur cette liste de MCSs. Le résultat

Table IV.5 – La résolution des conflits

Liste des MCSs	Règle choisie	Résultat de règle	MCS de résolution
{Text, XML, Database} {Text, XML, ¬ Database} {Text, ¬ XML, Database} {¬ Text, XML, Database} {¬ Text, XML, ¬ Database} {¬ Text, ¬ XML, Database}	R1	{¬ Text, XML, ¬ Database} ou {¬ Text, ¬ XML, Database}	{¬ Text, XML, ¬ Database}
	R2	{Text, XML, Database}	

d'application des règles est présenté dans la troisième colonne du Tableau IV.5.

La règle R1 représente le produit le plus complet ceci correspond au MCS qui permet de supprimer le minimum de caractéristiques (garder le maximum). Les MCSs correspondants sont : {¬ Text, XML, ¬ Dtabase} et {¬ Text, ¬ XML, Database}.

La règle R2 représente le produit le plus simple ceci correspond au MCS qui permet de supprimer le maximum de caractéristiques. Le MCS correspondant est alors {Text, XML, Dtabase}.

Comme présenté dans le Tableau IV.2, la règle R1 est choisie par les trois parties prenantes. En conséquence, le MCS rétourné par cette règle est bien commun et donc choisi pour résoudre les conflits détectés.

Étant donné qu'une règle peut retourner plusieurs MCSs tel qu'il est le cas pour R1, il suffit juste d'en choisir un de manière aléatoire. En conséquence, le MCS retenu est : {¬ Text, XML, ¬ Database}, il consiste à supprimer XML et garder les caractéristiques Text et Database (en supprimant les choix de leur suppression ¬ Text et ¬ Database).

IV.5 Caractéristiques de l'approche Colla-Config

L'approche Colla-Config est caractérisée par certains points originaux : (1) la flexibilité du processus de configuration, (2) la résolution des inconsistances et (3) la considération de satisfaction des parties prenantes.

IV.5.1 Flexibilité du processus de configuration

La flexibilité concerne la manière dont les parties prenantes collaborent durant le processus de configuration. En effet, l'espace de configuration peut être : (1) soit partagé en totalité entre les différentes parties prenantes qui ont alors la liberté d'exprimer leurs choix, (2) soit divisé en sous-espaces où chacune des parties prenantes se charge de la configuration d'un aspect bien spécifique de la LP. Généralement, la notion de division est intimement liée à l'attribution des rôles aux différentes parties

prenantes.
Notre approche s'inscrit dans le cadre des travaux qui ont étudié l'apport de la flexibilité dans la configuration collaborative des lignes de produits (Junior et al., 2011), (Stein et al., 2014), (Ochoa et al., 2015), (Dou et al., 2016).
Au contraire de l'approche basée sur ordre prédéfini, l'approche flexible permet aux parties prenantes d' exprimer librement leurs choix de configuration sans être contraintes à des choix précédents. Lorsque la configuration s'appuie sur un processus avec un ordre prédéfini (par exemple, (Mendonca et al., 2008) ; (Mendonca et al., 2007)), les décisions prises par certaines parties prenantes limitent les décisions prises ultérieurement par d'autres. Dans ce cas, il est nécessaire de revenir en arrière pour réviser certains choix de configuration décidés précédemment, ce qui rend difficile la dérivation d'une configuration valide acceptée par toutes les parties prenantes.
La contrepartie de l'approche flexible est que la coordination des activités de configuration n'est pas systématique, mais elle est assurée à travers la mise en place des moyens de communication et de partage des différents états de configuration afin d'assurer la cohérence globale du processus.

IV.5.2 Résolution des inconsistances

Généralement, une configuration est inconsistante si la combinaison des caractéristiques qui la constituent ne respecte pas les contraintes spécifiées dans le modèle de la LP. L'approche Colla-Config, propose un processus de résolution qui permet de détecter et résoudre les types d'inconsistance pouvant être détectés durant la configuration collaborative d'une LP (c.f section 3.2.1). Le processus de résolution est expliqué en détail dans le chapitre 5.

IV.5.3 Considération de satisfaction des parties prenantes

La satisfaction des parties prenantes est d'une importance primordiale lors de la configuration collaborative d'un produit. Cela peut être garanti en trouvant le bon compromis entre les choix de différentes parties prenantes pour en finir avec le produit souhaité. L'approche Colla-Config permet d'assurer la satisfaction des parties prenantes à travers la prise en compte de leurs préférences dans le processus de résolution des conflits.

IV.6 Conclusion

Ce chapitre a présenté un aperçu de l'approche Colla-Config et un exemple illustratif de configuration collaborative de la ligne des portails web. Cet exemple sera utilisé plus tard pour démontrer le principe de fonctionnement de l'approche Colla-Config. Le processus de configuration et résolution est décrit de manière détaillé dans le chapitre suivant.

V

Processus de configuration de Colla-Config

Sommaire

V.1 Introduction **68**

V.2 Préambule sur le formalisme CSP **68**

V.3 Algorithme de calcul des MCSs **69**

V.4 Processus de configuration Colla-Config **70**

V.5 Conclusion **82**

V.1 Introduction

Dans Colla-Config, la configuration collaborative d'une ligne de produits est ramenée à la résolution d'un problème de satisfaction des contraintes ou CSP (Constraint Satisfaction Problem). Les choix de configuration des différentes parties prenantes ainsi que les contraintes de la ligne de produits sont formalisés afin d'identifier et de résoudre les conflits. La configuration est dès lors assimilée à la résolution d'un problème de satisfaction de contraintes.
Ce chapitre décrit une démarche guidée pour la configuration collaborative d'une ligne de produits. Cette démarche couvre principalement :
- La fusion des configurations partielles des différents utilisateurs et sa vérification.
- Le calcul des sous-ensembles de correction minimaux ou MCS.
- Le choix du MCS de résolution selon les préférences des utilisateurs.

V.2 Préambule sur le formalisme CSP

Le formalisme CSP est un concept introduit par (Montanari and Rossi, 1988), qui consiste à la recherche des valeurs à affecter à un ensemble de variables soumises à certaines conditions ou contraintes qui restreignent la liberté du choix des valeurs.

V.2.1 Définition

Un CSP peut être défini comme un quadruplet P = (X, D, C, R) où :
- X = $\{X_1, ...X_n\}$ désigne l'ensemble des variables ;
- D = $\{D_1, ...D_n\}$ l'ensemble des domaines, avec D_i est l'ensemble de valeurs possibles de la variable X_i tel que $D_i = \{val_{i1}, val_{i2}, ...val_{ik}\}$;
- C = $\{C_1, ...C_m\}$ l'ensemble des contraintes où toute contrainte C_i concerne un sous-ensemble de variables : $C_i = \{X_{i1}, X_{i2}, ...X_{ik}\}$;
- R=$\{R_1, ...R_m\}$ l'ensemble des relations associées aux contraintes où chacune des relations R_i définit l'ensemble des combinaisons de valeurs permises par C_i.
Résoudre un CSP, veut dire soit trouver une solution ou toutes les solutions du problème, c'est à dire affecter à chacune des variables une valeur de façon à ce que toutes les contraintes soient satisfaites, soit déterminer qu'il n'existe aucune combinaison de valeurs qui satisfait les contraintes(Ghédira, 2012).

V.2.2 Problème de satisfiabilité

Un type particulier de CSP auquel s'intéresse le processus Colla-Config, est le CSP booléen où le domaine de valeurs des variables ne peut prendre que zero et

un (vrai ou faux).Trouver une solution à un CSP booléen consiste à déterminer s'il existe une assignation des variables propositionnelles qui rend la formule vraie.
Ceci est connu comme un problème SAT (Montanari and Rossi, 1988). En effet, SAT est le problème de base pour l'étude des méthodes de résolution par inférence logique en intelligence artificielle. Une instance de SAT est définie par un ensemble de variables propositionnelles (booléennes) X = $\{X_1, ...X_n\}$ et un ensemble C = $\{C_1, ...C_m\}$ de clauses.
Une clause est une disjonction de littéraux, où un littéral est une variable X_i ou sa négation $\bar{X}_i$.
La résolution d'un problème SAT consiste à décider si une formule logique sous forme normale conjonctive ou FNC (conjonction de clauses) est valide ou non.
Une des méthodes de résolution d'un problème SAT consiste à calculer la liste de MCSs pour rendre une FNC satisfiable.

V.3 Algorithme de calcul des MCSs

La non satisfiabilité d'un CSP peut être définie comment étant un ensemble de MUSs (Minimal Unsatisfiable Subsets en anglais). Un MUS est un sous-ensemble de contraintes qui est non satisfiable et minimal dont la suppression de l'un de ses éléments rend satisfiable l'ensemble des contraintes restantes (Liffiton and Sakallah, 2008). Résoudre un CSP consiste donc à trouver l'ensemble des MUSs et les "neutraliser" en supprimant au moins une contrainte de chaque MUS.
Étant donné qu'un MCS est un sous-ensemble irréductible de contraintes dont la suppression rend CSP satisfiable, un MCS doit donc contenir au moins une contrainte de chaque MUS de ce CSP.
Formellement, étant donné un CSP C non satisfiable, ses MUSs et MCSs sont définis comme suit (Liffiton and Sakallah, 2008) :
Définition 1 Un sous-ensemble $U \subseteq C$ est un MUS, si U est non satisfiable et $\forall C_i \in U, U \backslash \{C_i\}$ est satisfiable.
Définition 2 Un sous-ensemble $M \subseteq C$ est un MCS, si C$\backslash M$ est satisfiable et $\forall C_i \in M, C \backslash (M \backslash \{C_i\})$ est non satisfiable.
Comme proposé (Liffiton and Sakallah, 2008), le calcul de l'ensemble de MCSs possibles consiste à parcourir la liste des MUSs et identifier les différentes combinaison formées d'un ensemble de contraintes de chaque MUS (prendre à chaque fois une contrainte de chaque MUS) qui rendent la formule C satisfiable. Ces combinaisons constitue l'ensemble de MCSs.
L'exemple suivant illustre le principe de calcul des MCSs :
pour une formule *FNC* $\varphi = C_1 \wedge C_2 \wedge C_3 \wedge C_4 \wedge C_5 \wedge C_6$, où chaque C_i représente

une contrainte, la liste des MUSs est présentée dans le Tableau VI.10. Pour obtenir

TABLE V.1 – Liste des MUSs pour la formule φ

MUSs (φ)	C_1	C_2	C_3	C_4	C_5	C_6
$\{C_1, C_2\}$	X	X				
$\{C_1, C_3, C_4\}$	X		X	X		
$\{C_1, C_5, C_6\}$	X				X	X

la liste des MCSs, chaque MUS (ligne du Tableau) devient une disjonction ($\vee$) des contraintes C_i qui le constituent (colonnes ayant signe X). Ces disjonctions sont ensuite multipliées et factorisées pour obtenir la liste des MCSs.
La liste des MCSs pour la formule φ est donc calculée comme suit :
$\text{MCSs}(\varphi) = (C_1 \vee C_2) \times (C_1 \vee C_3 \vee C_4) \times (C_1 \vee C_5 \vee C_6)$
$\text{MCSs}(\varphi) = C_1 \vee C_2C_3C_5 \vee C_2C_3C_6 \vee C_2C_4C_5 \vee C_2C_4C_6$
$\text{MCSs}(\varphi) = \{\{C_1\}, \{C_2, C_3, C_5\}, \{C_2, C_3, C_6\}, \{C_2, C_4, C_5\}, \{C_2, C_4, C_6\}\}$
L'identification des MUSs et le calcul des MCSs correspondants sont adoptés dans Colla-Config, où la configuration est formalisée sous forme de système de contraintes, afin de vérifier sa consistance, détecter et résoudre les conflits.
Le processus Colla-Config est détaillé dans la section suivante.

V.4 Processus de configuration Colla-Config

Colla-Config repose sur la fusion et la formalisation des choix des différents parties prenantes sous forme d'une configuration présentée en tant que CSP. Les conflits sont identifiés en calculant la liste des MUSs et MCSs et résolus en choisissant un seul MCS selon les règles de substitution sélectionnées par les parties prenantes.

Comme schématisé dans la Figure V.1, le formalisme MAP (Rolland, 2007) est utilisé afin de modéliser de manière simple et flexible les étapes du processus Colla-Config. Le formalisme MAP permet de décrire un processus par des intentions (but à atteindre) et des stratégies qui permettent de passer d'une intentions à une autre. La modélisation d'une MAP est réalisée à travers un graphe orienté où les intentions sont représentées par des cercles, et les stratégies par des arcs entrants.

La Figure V.1 représente les étapes du processus Colla-Config et les différentes stratégies adoptées. Ce processus consiste en : *(i)* l'expression des préférences qui serviront pour la résolution des conflits ; *(ii)* l'expression des choix de configuration par chaque partie prenante à travers la sélection des caractéristiques à partir du modèle de la LP ; *(iii)* la formulation de tous les choix sous forme d'un système de contraintes ; *(iv)* la vérification de la configuration ; *(v)* la résolution des conflits en se basant sur les préférences des parties prenantes.

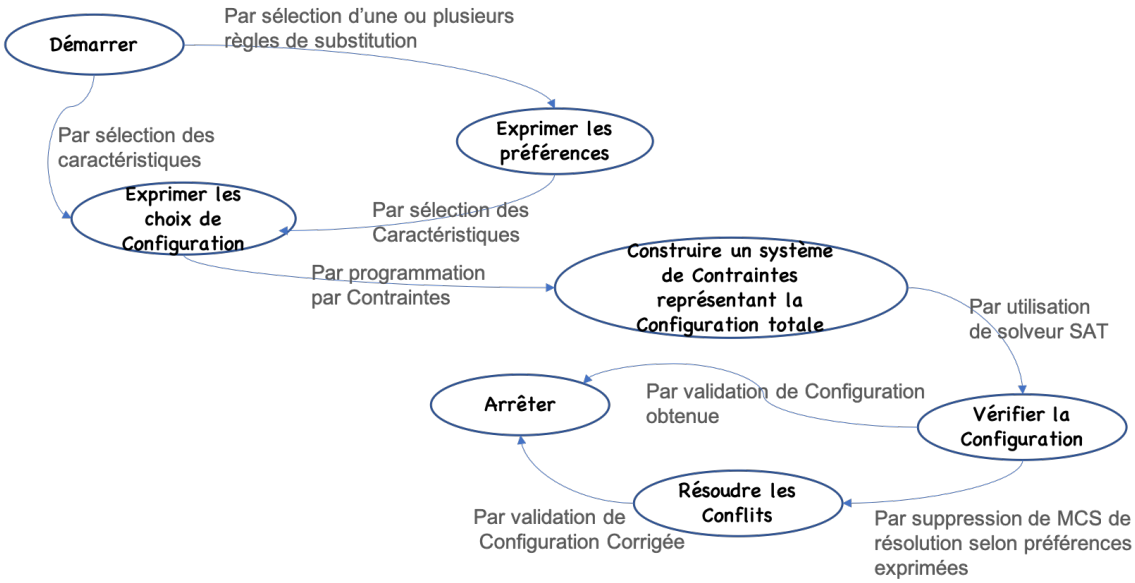


FIGURE V.1 – Processus de configuration Colla-Config

V.4.1 Formalisation de la configuration

La configuration peut être définie comme étant une instanciation du modèle de LP qui représente l'ensemble des caractéristiques sélectionnées tout en respectant les contraintes du modèle de la LP. Dans le contexte collaboratif, la configuration est la fusion des différentes caractéristiques sélectionnées par les différentes parties prenantes. Comme schématisé dans la Figure V.1, chaque partie prenante (PP_i) exprime ses choix en spécifiant les caractéristiques souhaitées (F_i) et celles non souhaitées ($\neg F_i$). La configuration Conf est formée par toutes ces caractéristiques comme illustré dans la Figure V.2. Pour la représentation réduite du modèle de la

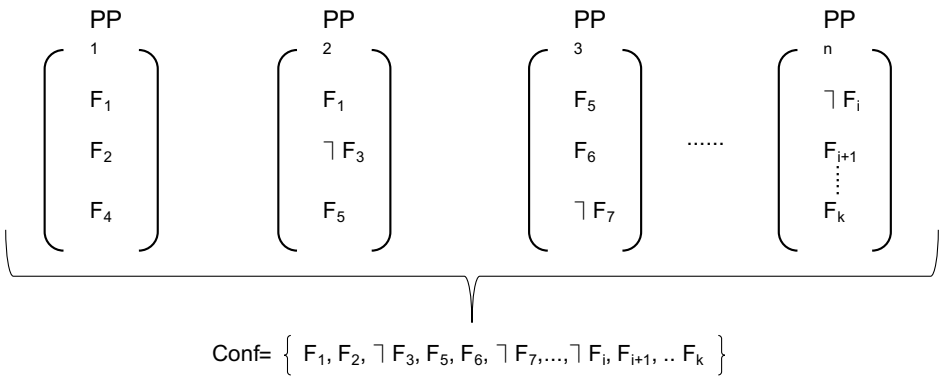


FIGURE V.2 – Représentation d'une configuration

ligne des portails web présenté dans la Figure V.3, un exemple de configuration est représenté dans la Figure V.4 .

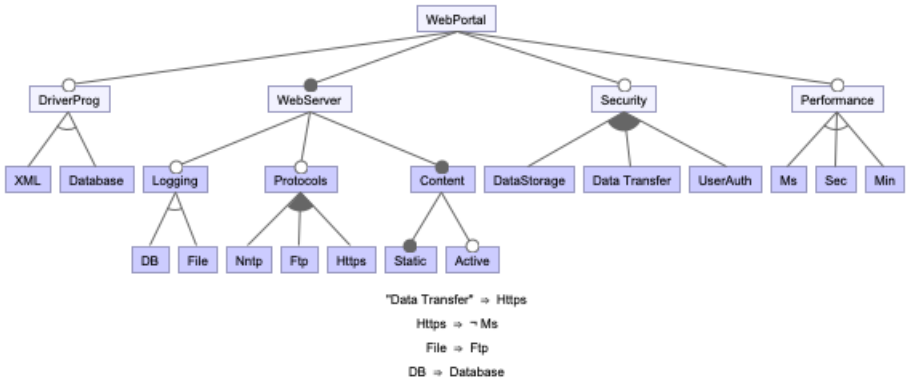


FIGURE V.3 – Modèle réduit de la ligne des portails web(Mendonca et al., 2008)

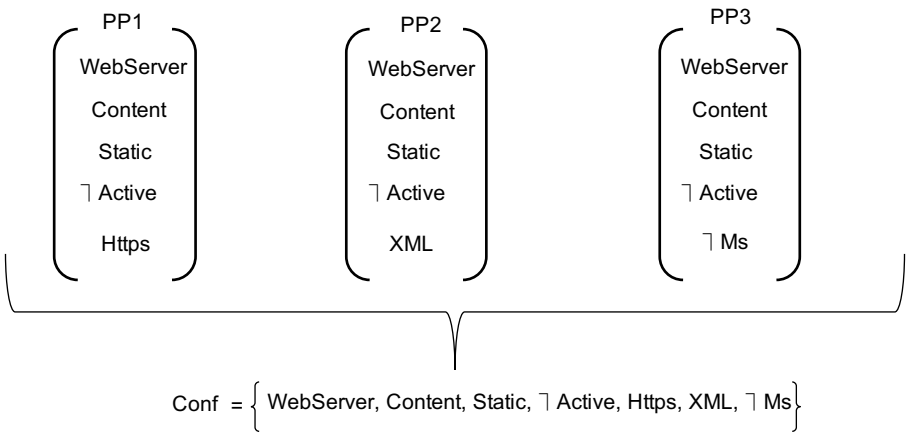


FIGURE V.4 – Exemple de représentation d'une configuration

Formellement, le problème de configuration peut être formalisé sous forme d'un CSP booléen formé d'un triplet (X, D, C) comme suit :
- X = $\{X_1, ...X_n\}$ désigne l'ensemble des caractéristiques ;
- D = {0,1} désigne le domaine de toutes les caractéristiques, 0 si la caractéristique est non sélectionnée et 1 dans le cas contraire.
- C = $\{C_1, ...C_m\}$ l'ensemble des contraintes où toute contrainte C_i concerne un sous-ensemble de caractéristiques : $C_i = \{X_{i1}, X_{i2}, ...X_{ik}\}$;
Chaque instance de ce CSP représente une configuration, qui pour vérifier sa satisfiabilité, il suffit de vérifier à travers un solveur SAT (cinquième intention dans la Figure V.1) si l'affectation des variables (sous forme CNF) rend la formule vraie ou non. Si après vérification le solveur retourne faux, ceci signifie que la présence ou l'absence d'une ou plusieurs caractéristiques rend la configuration non satisfiable.
Dans le contexte de la configuration collaborative, la non satisfiablité de la configuration est dûe au fait que deux ou plusieurs caractéristiques sélectionnées par différentes parties prenantes ne peuvent pas être retenues ensemble dans la configu-

ration.

V.4.2 Identification et résolution des conflits

Définition Formellement un conflit peut être défini comme suit :
Pour une configuration Co, un sous-ensemble $Cflt \subseteq Co$ est un conflit, si Co est non satisfiable et $\forall Ch_i \in Cflt, Co \backslash \{Ch_i\}$ est satisfiable.
Quatre types de conflits pouvant être détectés durant la configuration collaborative (cf. chapitre 4), notamment : (1) l'incohérence, (2) la non conformité, (3) conflit explicite et (4) conflit implicite.
Dans Colla-Config, les deux premiers types de conflits se résolvent à travers la propagation automatique des choix explicitement exprimés par une partie prenante. Par exemple, si une partie prenante sélectionne une caractéristique "A" qui exige la présence d'une caractéristique "B", cette dernière sera automatiquement sélectionnée pour empêcher la violation de la contrainte entre "A" et "B". Pour les deux autres types, la résolution est basée sur les préférences exprimées à travers un ensemble de règles de substitution. Ces règles sont expliquées dans la section suivante.

V.4.3 Les règles de substitution

Comme schématisé dans la Figure V.1, la résolution d'un conflit consiste à supprimer le MCS choisi selon les préférences des parties prenantes. Le but d'une telle résolution est de considérer équitablement tous les choix des parties prenantes et ne pas favoriser certains d'une manière prédéfinie. Les préférences sont exprimées à travers un ensemble de règles de substitution (cf. chapitre 4). La liste des règles de substitution est la suivante :

- ☐ *RS1 : Le produit le plus complet (toutes options).* Un produit complet contient le maximum possible de l'ensemble total des caractéristiques qui ont été sélectionnées par les différentes parties prenantes.
- ☐ *RS2 : Le produit le plus simple (minimum d'options).* Un produit simple contient le minimum possible de l'ensemble total des caractéristiques qui ont été sélectionnées par les différentes parties prenantes.
- ☐ *RS3 : Produit similaire à une configuration passée.* Une configuration est considérée similaire si elle respecte tous les choix de configuration exprimés par la partie prenante qui sélectionne RS3.
- ☐ *RS4 : Produit configuré par un profil similaire.* Des profils sont similaires s'ils ont le même rôle et le même statut. Dans ce cas, la configuration choisie doit respecter les choix de configuration effectués par le profil similaire à celui de la partie prenante ayant choisi la règle RS4.

- ☐ *RS5 : Prioriser mes choix explicites.* Une partie prenante peut exprimer ses choix de configuration en exprimant explicitement si elle veut ou non une telle caractéristique. Les sélections/désélections des caractéristiques qui découlent des choix de la partie prenante par propagation des contraintes sont des choix de configuration implicites. En cas de conflit, cette règle consiste à prioriser les choix explicites de la partie prenante par rapport aux choix des autres.

Chacune de ces règles correspond à un MCS bien précis. La correspondance entre les différentes règles de substitution et le MCS équivalent est présenté dans le Tableau V.2.

Table V.2 – Liste des règles de substitution et le MCS correspondant

Règle de substitution	**MCS correspondant**
RS1 : Le produit le plus complet (toute option)	MCS qui élimine le minimum de caractéristiques de la configuration
RS2 : Le produit le plus simple (minimum d'option)	MCS qui élimine le maximum de caractéristiques de configuration
RS3 : Produit similaire à une configuration passée	MCS qui respecte les choix de configuration d'une configuration précédente similaire
RS4 : Produit configuré par un profil similaire	MCS qui respecte les choix de configuration d'un profil similaire
RS5 : Prioriser mes choix explicites	MCS qui respecte le maximum des choix de configuration exprimés explicitement par la partie prenante

La projection des règles choisies sur l'ensemble de MCSs obtenu présente le cœur du processus de résolution qui couvre plusieurs cas de figure, à titre d'exemple : les règles donnent plusieurs MCSs, ou alors aucun MCS ne correspond aux règles choisies. Les détails du processus de résolution sont présentés dans la section suivante.

V.4.4 Processus de résolution des conflits

Comme schématisé dans la Figure V.5, le processus de résolution consiste en quatre étapes : *(1)* la suppression de la partie commune des choix de configuration, *(2)* le calcul de la liste des MCSs, *(3)* l'application des règles de substitution sur la liste de MCSs obtenue et *(4)* la suppression du MCS de résolution de la configuration initiale pour la valider.

V.4.4.1 Suppression de la partie commune des choix de configuration

Afin de résoudre le(s) conflit(s) identifié(s), il faut tout d'abord supprimer l'ensemble des choix de configuration qui sont communs entre toutes les parties prenantes. Comme illustré par la Figure V.6, une configuration est initialement formée par trois parties : la partie des caractéristiques obligatoires dont la présence est indispensable dans toutes les configurations, la partie des choix de configuration communs entre toutes les parties prenantes et enfin la partie non commune. la partie des caractéristiques obligatoires ainsi que la partie des choix de configuration communs

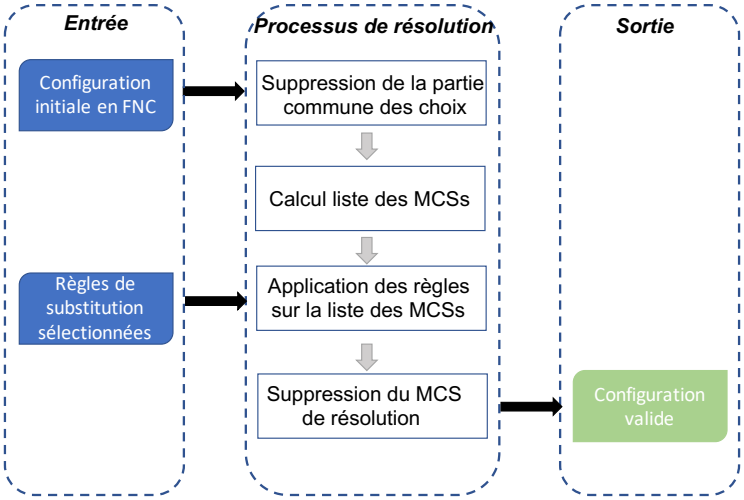


FIGURE V.5 – Processus de résolution des conflits

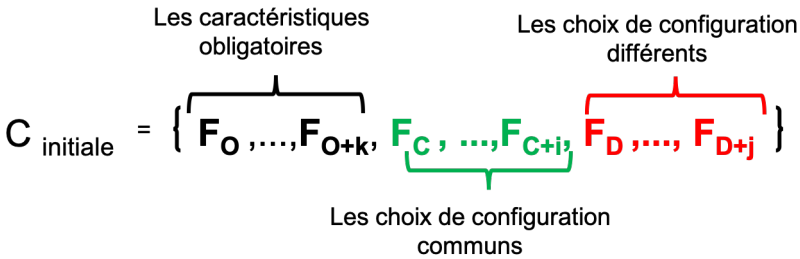


FIGURE V.6 – Structure d'une configuration initiale

ne peuvent pas être un source de conflit. Par conséquent, pour une configuration inconsistante, seuls les choix différents (la partie non commune) sont considérés dans le calcul des MCSs.

Pour le modèle de la ligne des portails web présenté dans la Figure V.3, un exemple de configuration initiale est présenté dans le Tableau V.3. Selon la structure illustrée

TABLE V.3 – Exemple de configuration

Partie prenante	**Choix de configuration**
PP1	WebServer, Content, Static, $\neg Active$, Protocols, Https , $\neg Ms$ (par propagation de contrainte)
PP2	WebServer, Content, Static, $\neg Active$, DriverProg, XML,$\neg Database$ (par propagation de contrainte) Performance, Sec,$\neg Ms$ (par propagation de contrainte) , $\neg Min$ (par propagation de contrainte)
PP3	WebServer, Content, Static, $\neg Active$, DriverProg, Database, $\neg XML$ (par propagation de contrainte) Performance, Ms, $\neg Sec$ (par propagation de contrainte), $\neg Min$, $\neg Https$ (par propagation de contrainte)
Configuration initiale en FNC	**WebServer** $\wedge$ **Content** $\wedge$ **Static** $\wedge\neg$**Active** $\wedge$ Https $\wedge$ $\neg Https$ $\neg Ms\wedge$ DriverProg $\wedge$ XML $\wedge\neg Database\wedge$ Performance $\wedge$ Sec $\wedge\neg Min\wedge$ Database $\wedge\neg XML\wedge$ Ms $\wedge\neg Sec$

par la Figure V.6, la configuration initiale est formée par un ensemble de caractéristiques obligatoires (cf. Figure V.3) qui sont : *WebServer, Content* et *Static,* ainsi que la partie des choix communs entre toutes les parties prenantes : $\neg Active$ (marquée en gras dans le Tableau V.3). Le reste de la configuration est constitué par les choix dif-

férents ainsi que les propagations correspondantes. Ces propagations servent à éviter les situations d'inconsistance de type *incohérence* et *non conformité*. Par exemple, pour empêcher la *non conformité*, dès que la partie prenante (PP1) sélectionne la caractéristique "Https", "Ms" est automatiquement éliminée par la contrainte *exclude* entre "Https" et "Ms".

V.4.4.2 Calcul de liste des MCSs

Plusieurs travaux s'interessent au calcul de MCS tels que (Liffiton and Sakallah, 2008), (Marques-Silva et al., 2013), (Morgado et al., 2013), (Mencía et al., 2015), (Rinconn et al., 2015). L'algorithme initiale proposé par (Liffiton and Sakallah, 2008) est adopté dans cette thèse vu qu'il permet le calcul des MCSs à partir de la liste des MUSs.

Le calcul des MCSs (cf. section 3), consiste initialement à calculer la liste des MUSs, ce qui correspond dans Colla-Config à l'identification des choix conflictuels. Ensuite, partant de cette liste, il faut identifier toutes les combinaisons possibles des choix dont la suppression assure la résolution de tous les conflits détectés.

Pour la méthode Colla-Config, comme décrit par le pseudo code de l'Algorithme 1, le calcul de la liste des MUSs consiste à identifier les contraintes violées du modèle de la LP et déduire le MUS correspondant à partir des clauses de cette contrainte. La ligne 4 de l'Algorithme 1 permet de vérifier si la contrainte violée (conflit) est de type *exclude* ou *XOR*. En effet, ces deux types de contraintes, consistent en une exclusion mutuelle entre les caractéristiques qui les constituent. En conséquence, un conflit aura lieu à chaque fois que deux ou plusieurs parties prenantes choisissent parmi des caractéristiques en exclusion mutuelle. Dans ce cas, comme décrit par la ligne 9 de l'Algorithme 1, le MUS correspond à l'absence et présence à la fois de chacune de ces caractéristiques. Formellement, si C_1 et C_2 sont deux caractéristiques en exclusion, et elles ont été choisies par différentes parties prenantes, elles seront représentées sous la forme $(C_1 \wedge \neg C_2)$ et $(C_2 \wedge \neg C_1)$ et les MUSs correspondants seront : $\{C_1, \neg C_1\}$ et $\{C_2, \neg C_2\}$.

Dans le cas où la contrainte violée est de type *require*, comme décrit par la ligne 15 de l'Algorithme 1, le conflit a lieu si certaines parties prenantes ne veulent pas avoir la (les) caractéristique(s) exigée(s)(CE). En effet, pour deux caractéristiques C_1 et C_2 tel que C_1 *require* C_2, si une partie prenante (PP1) choisit la caractéristique (C_1) ceci propage automatiquement la sélection de la partie exigée (CE) qui est C_2. Dans ce cas, un conflit aura lieu si une autre partie prenante (PP2) ne veut pas cette caractéristique $(\neg C_2)$. Par conséquent, comme indiqué dans la ligne 20 de l'algorithme 1, le MUS correspondant sera $\{C_2, \neg C_2\}$.

Pour le conflit explicite, si la caractéristique n'est pas impliquée dans aucune contrainte,

Algorithme 1 : Algorithme de calcul des MUSs à partir des conflits

```
1  n ← nombre_contraintes ;
2  Liste_MUS ← ∅ ;
3  for (i de 1 à n) do
4      if (Etat_Contrainte_i = Violée)ET(Ty_Contrainte_i ∈ {Exclude, XOR })
        then
5          j ← 1 ;
6          m ← nombre_caractéristique(contrainte_i) ;
7          while (j < m) do
8              if (C_j ∈ Contrainte_i)ET(C_j ∈ Liste_Choix_i) then
9                  Liste_MUS← Liste_MUS ∪ {C_j, ¬ C_j };
10             end
11             j ← j+1 ;
12         end
13     end
14     if (Etat_Contrainte_i = Violée)ET(Ty_Contrainte_i = Require ) then
15         j ← 1 ;
16         m ← nombre_caractéristique(contrainte_i) ;
17         while (j < m) do
18             if (CE_j ∈ partie_exigée_contrainte_i)ET( CE_j ∈ Liste_Choix_i)
                then
19                 Liste_ MUS ← Liste_ MUS ∪ {CE_j , ¬ CE_j } ;
20             end
21             j ← j+1 ;
22         end
23     end
24 end
25 Retourner(Liste_MUS);
```

le MUS correspondant est constitué de deux choix de présence et absence de cette caractéristique comme suit : {C, ¬C }. Par contre, si la caractéristique est impliquée dans une contrainte, le MUS est calculé comme décrit par le pseudo code de l'Algorithme 1.

Pour le scénario de configuration présenté dans le Tableau V.3, les conflits identifiés dans la configuration initiale sont présentés dans le Tableau V.4. Le premier

TABLE V.4 – Liste des conflits identifiés

Conflit identifié	**Type de conflit**	**MUS correspondant**
1) Https VS Ms	Implicite	{Https, $\neg Https$} {Ms, $\neg Ms$}
2) XML VS Database	Implicite	{XML,$\neg XML$ } {Database, $\neg Database$}
3) Ms VS Sec	Implicite	{Ms, $\neg Ms$} {Sec, $\neg Sec$}

conflit est implicite et dû à la violation de la contrainte *exclude* entre les caractéristiques "Https" et "Ms" vu que la première partie prenante (PP1) a choisi "Https" ce qui correspond formellement à (Https, ¬ Ms) et la troisième partie prenante (PP3) a choisi "Ms" ce qui correspond formellement à (MS, ¬ Https). Selon l'algorithme 1 (ligne 9), les MUS identifiés sont {Https, $\neg Https$}, {Ms, $\neg Ms$}.
Pour le deuxième et le troisième conflits, ils sont aussi implicites et dus à la dépendance *XOR* respectivement entre "XML" et "Database" et "Ms" et Sec".
Étant donné que ces caractéristiques sont en exclusion mutuelle, les MUSs identifiés sont : {XML,¬ XML }, {Database, ¬ Database}, {Sec, ¬ Sec} et {Ms, $\neg Ms$}.
En se basant sur l'ensemble des MUSs identifiés, la liste des MCSs est ensuite calculée(cf. Section 3). La liste initiale des MCSs obtenus est présentée dans le Tableau V.5. Initialement, le calcul des MCS suggère plusieurs combinaisons pour corriger les

TABLE V.5 – Liste initiale des MCSs calculés

Liste des MUSs identifiés
{Https, ¬ Https}, {Ms, ¬ Ms}, {XML,¬ XML }, {Database, ¬ Database}, {Sec, ¬ Sec}
Liste initiale des MCSs résultants
(Https, Ms, XML, Database, Sec) ∨ (Https, Ms, XML, Database, ¬ Sec) ∨ (Https, Ms, XML, ¬ Database, Sec) ∨ (Https, Ms, XML, ¬ Database, ¬ Sec) ∨ (Https, Ms, ¬ XML, Database, Sec) ∨ (Https, Ms, ¬ XML, Database, ¬ Sec) ∨ **(Https, Ms, ¬XML, ¬ Database, Sec)** ∨ **(Https, Ms, ¬ XML, ¬ Database, ¬ Sec)** ∨ (Https, ¬ Ms, XML, Database, Sec) ∨ **(Https, ¬ Ms, XML, Database, ¬ Sec)** ∨ (Https,¬ Ms, XML,¬ Database, Sec) ∨ **(Https, ¬ Ms, XML, ¬ Database, ¬ Sec)** ∨ (Https, ¬ Ms, ¬ XML, Database, Sec) ∨ **(Https, ¬ Ms, ¬ XML, Database, ¬ Sec)** ∨ **(Https, ¬ Ms, ¬ XML, ¬ Database, Sec)** ∨ **(Https, ¬ Ms, ¬ XML, ¬ Database, ¬ Sec)** ∨ (¬ Https, Ms, XML, Database, Sec) ∨ (¬ Https, Ms, XML, Database, ¬ Sec) ∨ (¬ Https, Ms, XML, ¬ Database, Sec)∨ (¬ Https, Ms, XML, ¬ Database, ¬ Sec) ∨ (¬ Https, Ms, ¬ XML, Database, Sec) ∨ (¬ Https, Ms, ¬ XML, Database, ¬ Sec)∨ **(¬ Https, Ms, ¬ XML, ¬ Database, Sec)** ∨ **(¬ Https, Ms, ¬ XML, ¬ Database, ¬ Sec)** ∨ **(¬ Https, ¬ Ms, XML, Database, Sec)** ∨ **(¬ Https,¬ Ms, XML, Database, ¬ Sec)** ∨ **(¬ Https, ¬ Ms, XML, ¬ Database, Sec)** ∨ **(¬ Https, ¬ Ms, XML, ¬ Database, ¬ Sec)**∨ **(¬ Https, ¬ Ms, ¬ XML, Database, Sec)** ∨ **(¬ Https, ¬ Ms, ¬ XML, Database, ¬ Sec)** ∨ **(¬ Https, ¬ Ms, ¬ XML, ¬ Database, Sec)**∨ **(¬ Https, ¬ Ms, ¬ XML, ¬ Database, ¬ Sec)**

MUSs identifiés. Parmi les MCSs calculés, certains violent les contraintes du modèle

de la LP. Par exemple, la combinaison **(Https, Ms, ¬XML, ¬ Database, Sec)** ne peut pas être considérée comme un MCS, car elle permet de garder les caractéristiques "XML" et "Database" bien qu'elles sont mutuellement exclusives à cause de la contrainte *exclude*. De même, pour toutes les autres combinaisons marquées en gras dans la liste initiale présentée dans le Tableau V.5.
Par conséquent, la liste initiale des MCSs est filtrée à travers l'algorithme 2. Comme décrit par la ligne 7, toute combinaison identifiée dans la liste initiale qui viole une des contraintes, ne sera pas considérée comme MCS. La liste finale des MCSs est présentée dans le Tableau V.6.

Algorithme 2 : Algorithme de filtrage de la liste initiale de MCSs

```
1  n ← nombre initial de MCSs ;
2  m ← nombre de contraintes ;
3  Liste_initiale ← ensemble initial de MCSs ;
4  Liste_finale ← ∅ ;
5  for (i de 1 à n) do
6  |  for (j de 1 à m) do
7  |  |  if (MCS_i ⊆ ¬ Contrainte_j) then
8  |  |  |  Liste_finale ← Liste_initiale - MCS_i ;
9  |  |  else
10 |  |  |  Liste_finale ← Liste_finale + MCS_i ;
11 |  |  end
12 |  end
13 end
14 Retourner(Liste_finale);
```

Table V.6 – Liste finale des MCSs calculés

Liste finale des MCSs
{Https, Ms, XML, Database, Sec }, {Https, Ms, XML, Database, ¬ Sec }, {Https, Ms, XML, ¬ Database, Sec }
{Https, Ms, XML, ¬ Database, ¬ Sec }, {Https, Ms, ¬ XML, Database, Sec }, {Https, Ms, ¬ XML, Database, ¬ Sec },
{Https, ¬ Ms, XML, Database, Sec }, {Https, ¬ Ms, XML,¬ Database, Sec} {Https, ¬ Ms, ¬ XML, Database, Sec }
{¬ Https, Ms, XML, Database, Sec}, {¬ Https, Ms, XML, Database, ¬ Sec}, {¬ Https, Ms, XML, ¬ Database, Sec }
{¬ Https, Ms, XML, ¬ Database, ¬ Sec }, {¬ Https, Ms, ¬ XML, Database, Sec}, {¬ Https, Ms, ¬ XML, Database, ¬ Sec}

V.4.4.3 Application des règles de substitution sur liste des MCSs

Décider quel MCS choisir pour résoudre le(s) conflit(s) identifié(s) est basé sur les préférences des parties prenantes exprimées à travers les règles de substitution. Ceci consiste à appliquer l'ensemble des règles sélectionnées sur la liste des MCSs obtenue.
Étant donné qu'une partie prenante est libre de choisir ou non une ou plusieurs règles, trois situations d'application sont possibles selon le nombre total des règles sélectionnées. Ces situations sont illustrées dans la Figure V.7 comme suit :

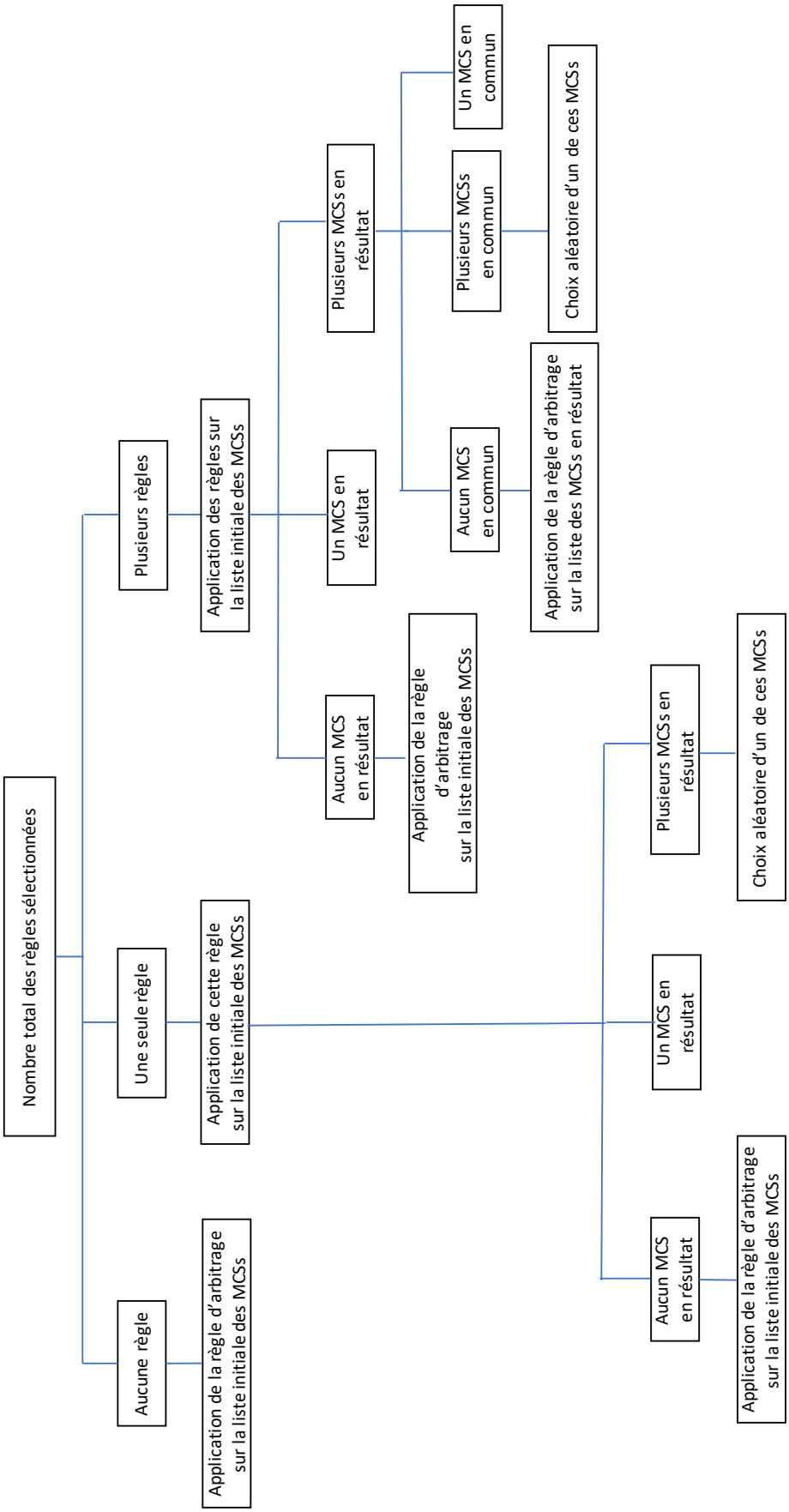


FIGURE V.7 – Schéma d'application des règles sur la liste des MCSs

1. ***Aucune règle n'est choisie*** : si toutes les parties prenante n'ont choisi aucune règle, la sélection du MCS de résolution consiste à appliquer une règle supplémentaire appelée *règle d'arbitrage*. Cette règle permet de privilégier une partie prenante en priorisant ses choix. Ceci est générique et diffère d'un contexte à un autre (compagnie et son organigramme). En conséquence, le MCS qui respecte les choix de la partie prenante priorisée est choisi.
2. ***Une seule règle est choisie*** : si seulement une partie prenante a choisi une règle, cette règle est directement appliquée sur la liste des MCSs selon le Tableau V.2 de correspondance entre les règles et les MCSs. Dans ce cas, trois résultats sont possibles :
   - *Aucun MCS en résultat :* si la règle choisie ne retourne aucun MCS, la règle d'arbitrage est donc appliquée sur la liste initiale des MCSs.
   - *Un MCS en résultat :* si la règle retourne un seul MCS, ce dernier est alors choisi comme MCS de résolution où les choix qu'il contient sont simplement retirés de la configuration.
   - *Plusieurs MCS en résultat :* si la règle retourne une liste de MCSs, une sélection aléatoire est appliquée sur cette liste pour en choisir un.
3. ***Plusieurs règles sont choisies*** : dans ce cas, les différentes règles sont appliquées une par une sur la liste initiale des MCSs. Selon le schéma d'application (cf. Figure V.7), trois résultats sont possibles :
   - *Aucun MCS en résultat :* si l'application des règles choisies ne retourne aucun MCS, la règles d'arbitrage est donc appliquée sur la liste initiale pour choisir le MCS qui respecte les choix de configuration de la partie prenante priorisée.
   - *Un MCS en résultat :* si seulement une seule règle retourne seulement un MCS, ce dernier est choisi comme MCS de résolution.
   - *Plusieurs MCS en résultat :* si les règles choisies retournent plusieurs MCSs, il faut vérifier s'il y a une intersection entre les MCSs obtenus. Comme illustré dans la Figure V.7, trois situations sont également possibles :
     — *Aucun MCS en commun* : dans ce cas, l'application des règles sélectionnées par les parties prenantes donne des MCS différents. Ici, il est difficile de choisir le MCS de résolution. En conséquence, la règle d'arbitrage est appliquée sur la liste des MCSs.
     — *Un MCS en commun* : dans ce cas, la liste des règles sélectionnées par les parties prenantes retourne un MCS commun en résultat, ce dernier est choisi comme MCS de résolution.

— *Plusieurs MCSs en commun* : dans ce cas, le MCS de résolution est choisi en sélectionnant aléatoirement un des MCSs de la liste commune.

En considérant comme exemple la configuration présentée dans le Tableau V.3, en plus des choix de configuration, les parties prenantes ont sélectionné des règles de substitution comme suit : la première partie prenante PP1 a choisi RS2, qui représente le produit le plus simple. La deuxième partie prenante PP2 n'a choisi aucune règle et la troisième partie prenante PP3 a choisi la RS5 qui permet de prioriser ses choix explicites.
Selon le schéma d'application (Figure V.7), la situation représente le cas où plusieurs règles ont été sélectionnée (RS2, RS5). Comme présenté dans le Tableau V.7, l'application de ces règles retourne plusieurs MCSs en résultat. RS2 retourne le MCS qui élimine le maximum de caractéristiques qui est *{Https, Ms, XML, Database, Sec}* . RS5 retourne le MCS qui priorise les choix explicites de PP3 ("Ms" et "Database".) qui est *{Https, ¬ Ms, XML, ¬ Database, Sec }*.
Les résultats obtenus sont différents (aucun MCSs en commun). Par conséquent, la règle d'arbitrage est appliquée sur la liste des MCSs retournée. Le MCS de résolution est donc sélectionné selon cette règle. Comme illustré dans le Tableau V.7, PP3 est la partie prenante priorisée, par conséquent, le MCS qui respecte ses choix de configuration est choisi comme MCS de résolution qui est : {Https, ¬ Ms, XML, ¬ Database, Sec}. Les choix de configuration présents dans ce MCS sont supprimés

Table V.7 – Résultat d'application des règles

Partie prenante	**Partie Priorisée**	**Règle choisie**	**Résultat d'application**	**MCS de résolution**
PP1	PP3	RS2	{XML, Database, Ms, Sec, Https}	{Https, ¬ Ms, XML, ¬ Database, Sec}
PP2		Aucune	-	
PP3		RS5	{Https, ¬ Ms, XML, ¬ Database, Sec}	

de la configuration initiale pour la valider. En conséquence, la configuration finale valide ne contient que la liste des caractéristiques suivantes :

Configuration finale = WebServer $\wedge$ Content $\wedge$ Static $\wedge DriverProg \wedge Database \wedge Performance \wedge$ Ms

V.5 Conclusion

Ce chapitre a présenté le processus de configuration de la méthode Colla-Config qui permet à plusieurs parties prenantes de configurer une ligne de produits de manière collaborative. Pour la gestion des conflits, l'approche Colla-Config propose un processus de résolution dynamique qui permet d'identifier et résoudre les conflits selon les préférences des parties prenantes en utilisant la méthode de calcul des

MUS-MCS. Les différentes étapes du processus de résolution ont été illustrées avec l'exemple de la ligne des Portails web.
Le chapitre suivant évalue l'approche Colla-Config à travers une expérimentation contrôlée.

VI

Evaluation

Sommaire

VI.1 Introduction **85**

VI.2 Présentation de l'outil Colla-Config **85**

VI.3 Principe d'expérimentation **86**

VI.4 Question de recherche **87**

VI.5 Protocole expérimental pour l'évaluation de l'utilisabilité **87**

VI.6 Expérimentation **90**

VI.7 Résultats **97**

VI.8 Les menaces à la validité **100**

VI.9 Conclusion **103**

VI.1 Introduction

Ce chapitre porte sur l'évaluation de l'approche Colla-Config, un outil a été développé qui permet la configuration collaborative des modèles de lignes de produits. L'approche Colla-Config est évaluée en terme d'utilisabilité.

VI.2 Présentation de l'outil Colla-Config

Afin d'évaluer l'approche proposée, un outil qui permet la configuration collaborative a été développé. Cet outil, baptisé « Colla-Config », permet de configurer les modèles de caractéristiques des lignes de produits selon le processus de l'approche proposée (cf. chapitre 5).
Colla-Config permet aux utilisateurs de :

— Exprimer leurs préférences en sélectionnant une ou plusieurs règles de substitution.

— Exprimer leurs choix de configuration en spécifiant la liste des caractéristiques désirées et non désirées.

— Voir le résultat de résolution des conflits selon leurs préférences.

— Voir la configuration finale valide.

La Figure VI.1 présente l'architecture globale de Colla-Config. La technologie utilisée pour l'implémentation de Colla-Config est Vue js.
Cette architecture est composée des éléments suivants :

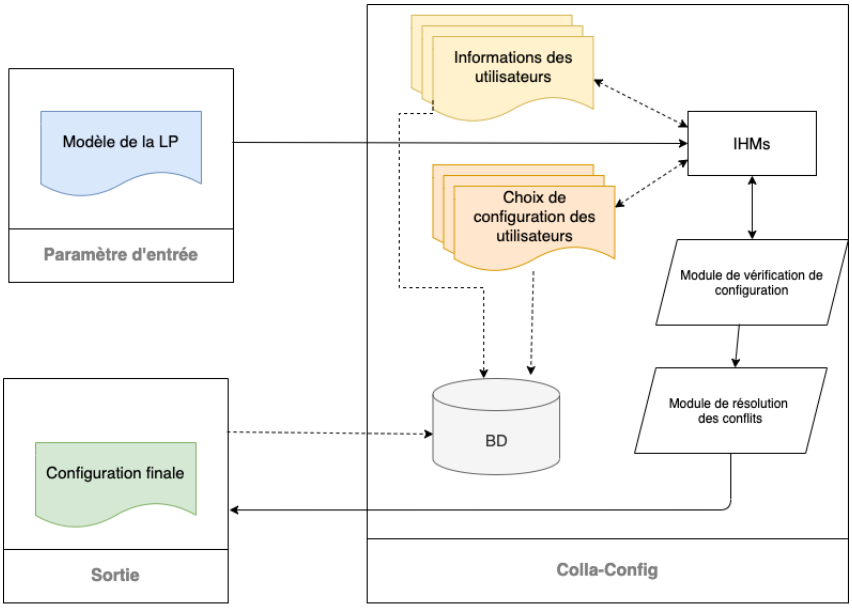


FIGURE VI.1 – Architecture globale de Colla-Config.

- Paramètre d'entrée : Fichier xml représentant le modèle de la ligne à configurer.
- Colla-Config :
  - Les interfaces graphiques de l'utilisateur (IHM) : permettant aux utilisateurs d'introduire leurs informations personnelles, choisir les règles de substitutions et faire les choix de configuration.
  - Module de vérification de la configuration : ce module est responsable de fusion des choix des utilisateurs et la vérification de la consistance de la configuration en utilisant un solveur SAT.
  - Module de résolution des conflits : ce module est responsable de calcul de la liste de MCSs et la sélection de MCS de résolution selon les règles sélectionnées par les utilisateurs.
  - BD : la base de données contenant les informations des utilisateurs et les configurations effectuées (finales).
- Sortie : la configuration finale corrigée avec le MCS de résolution choisi.

VI.3 Principe d'expérimentation

L'expérimentation réalisée dans cette thèse est une expérimentation contrôlée menée avec un groupe de doctorants au sein du Centre de Recherche en Informatique. Cette expérimentation consiste à évaluer l'utilisabilité de l'outil qui supporte l'approche Colla-Config. L'utilisabilité est définie par l'Organisation Internationale de Normalisation (Standardization, 1998) comme la capacité d'un produit à être utilisé par des utilisateurs particuliers pour atteindre des objectifs spécifiques avec efficacité, efficience et satisfaction dans un contexte d'utilisation déterminé.

Le processus d'expérimentation comporte quatre grandes étapes qui sont : la formulation de question de recherche, lacréation du protocole expérimental, l'exécution du protocole puis l'interprétation des résultats (cf. Figure VI.2). Les différentes étapes sont détaillées dans les sections suivantes.

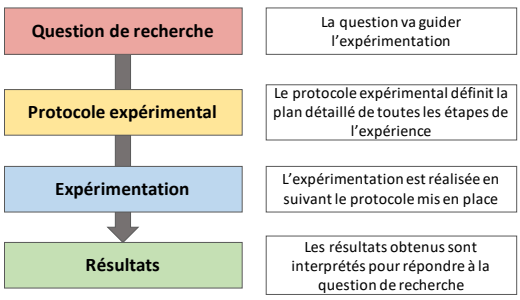


FIGURE VI.2 – Le processus d'expérimentation.

VI.4 Question de recherche

L'objectif de cette expérimentation est d'évaluer Colla-Config, notamment la technique de résolution des conflits. Pour guider le protocole d'expérimentation, la question de recherche suivante a été défini :
-**QR** Colla-Config est-il un outil utile pour supporter la configuration collaborative des lignes de produits ?
La réponse à cette question est obtenue à partir de l'analyse et l'interprétation des informations recueillies par des moyens quantitatifs et qualitatifs mises en place durant l'expérimentation.

VI.5 Protocole expérimental pour l'évaluation de l'utilisabilité

Le protocole expérimental décrit les différentes étapes du processus d'évaluation de l'utilisabilité de l'approche Colla-Config. L'utilisabilité a été définie de différentes manières dans la littérature. On distingue quelques définitions générale à partir de trois normes différentes (Seffah et al., 2006) :
- *Un ensemble d'attributs qui influent sur l'effort nécessaire à l'utilisation et sur l'évaluation individuelle de cette utilisation, par un ensemble déclaré ou implicite d'utilisateurs* (ISO/IEC 9126, 1991).
- *La mesure dans laquelle un produit peut être utilisé par des utilisateurs spécifiés pour atteindre des objectifs spécifiés avec efficacité, efficience et satisfaction dans un contexte d'utilisation spécifié* (ISO 9241-11,1998).
- *La facilité avec laquelle un utilisateur peut apprendre à utiliser, préparer les entrées et interpréter les sorties d'un système ou d'un composant* (IEEE Std.610.12-1990).
L'évaluation d'utilisabilité permet de mesurer à quel point on est satisfait quand on utilise un système logiciel. Pour l'approche Colla-Config, l'idée principale est de tester la facilité d'utilisation de l'outil et donc d'avoir un aperçu de la facilité ou de la difficulté à suivre et à comprendre l'approche Colla-Config. Dans le cadre de cette évaluation, il a été décidé de développer et de mener un test d'utilisabilité en utilisant le format ISO/IEC Common Industry Format (CIF) pour les tests d'utilisabilité(ISO/IEC, 2006). Cette évaluation sera assurée à travers la définition d'un ensemble de métriques dont les valeurs seront déduites de l'expérimentation.

VI.5.1 Processus d'évaluation

Le test d'utilisabilité a été conçu comme un processus comportant huit activités (voir Figure VI.3). Le principe d'évaluation consiste à mener une expérimentation contrôlée avec un groupe de participants qui ont un niveau similaire de connaissance de l'ingénierie des lignes de produits. Dans une première partie de l'expérimentation, les participants vont configurer un produit selon l'approche Colla-Config. Dans une deuxième partie, ils vont configurer un produit selon l'approche de (Mendonca et al., 2008). Ceci permettra de comparer le principe des deux approches et les évaluer via le retour d'expérience des participants.
Les différentes données recueillies du processus d'évaluation seront utilisées pour calculer les différentes mesures relatives aux métriques de l'utilisabilité notamment : l'efficacité, l'efficience et la satisfaction. Les différentes activités du processus d'évaluation d'utilisabilité sont présentées dans la Figure VI.3 :

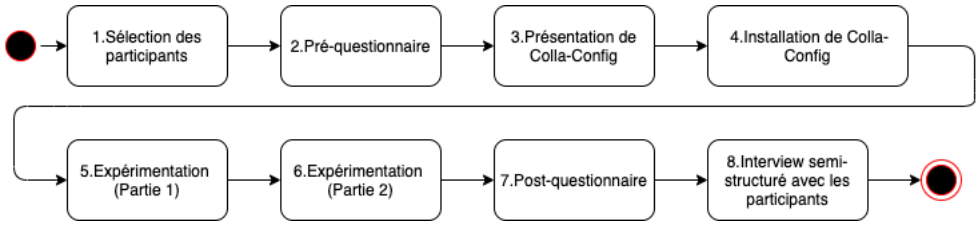


FIGURE VI.3 – Processus de test d'utilisabilité.

1. **Sélection des participants :** les participants sont onze doctorants en informatique qui ont participé de manière volontaire et appartiennent à différents laboratoires de recherche (CRI en France , RIADI en Tunisie et GIDTIC en Colombie). Les doctorants ayant accepté de participer à l'expérimentation ont tous des connaissances variées sur l'ingénierie des lignes de produits.
2. **Pré-questionnaire :** il permet de savoir le niveau de connaissance des doctorants dans la configuration des lignes de produits notamment la configuration collaborative et s'assurer qu'ils ont un minimum de connaissance leur permettant de faire l'expérimentation sur l'outil Colla-Config (voir Annexe).
3. **Présentation de l'outil :** cette présentation est importante pour former les participants sur les concepts principaux nécessaires pour pouvoir mener cette expérimentation. Elle consiste en une première partie à initier les participants à des sujets tels que l'ingénierie des lignes des produits, la modélisation des caractéristiques et la configuration collaborative des lignes de produits. Dans une deuxième partie, la présentation illustre une petite démonstration qui explique le fonctionnement de Colla-Config à travers l'illustration d'un petit exemple.

4. **Installation :** c'est l'étape de préparation de l'environnement technique (voir Annexe). Elle consiste à initier les participants à un document qui décrit une série d'étapes pour mettre en place une configuration collaborative en utilisant l'outil Colla-Config.
5. **Expérimentation partie 1 :** durant cette étape, les participants utilisent l'outil pour configurer un produit de manière flexible selon l'approche Colla-Config.
6. **Expérimentation partie 2 :** cette deuxième partie de l'expérimentation consiste à configurer un produit selon une autre approche dont le principe de configuration collaborative est différent de celui de Colla-Config. En conséquence, les participants utilisent l'outil pour configurer un produit selon l'approche de (Mendonca et al., 2008). Contrairement à Colla-Config, cette approche illustre le principe de configuration à ordre prédéfini avec une résolution systématique des conflits où les décisions de configuration prises antérieurement auront un impact sur les décisions ultérieures.
7. **Post-questionnaire :** après l'expérimentation, les doctorants répondent à un questionnaire (voir Annexe) qui porte des questions sur l'environnement de l'expérimentation, de la satisfaction générale, question sur le principe de configuration avec l'approche Colla-Config et celui avec l'approche simulée.
8. **Interview :** consiste à poser des questions ouvertes sur l'utilisabilité de l'outil supportant Colla-Config : qu'est ce que les participants ont aimé le plus ? qu'est ce qu'ils n'ont pas aimé ? qu'est ce que ils proposent comme amélioration ? qu'est ce que ils proposent comme nouveauté ?

VI.5.2 Les métriques d'utilisabilité

L'utilisabilité est de plus en plus reconnue comme un facteur de qualité important pour les systèmes logiciels interactifs.
Afin d'évaluer celle de l'outil Colla-Config, trois métriques ont été définies qui sont :
- Efficacité : c'est la capacité de produire un résultat souhaité ou la capacité de produire la sortie souhaitée. Lorsque quelque chose est jugé efficace, cela signifie qu'il a un résultat prévu ou attendu. Elle se mesure à travers :

— *Taux d'achèvement de tâche :* consiste à mesurer le taux de réalisation de chaque tâche du processus de configuration par les différents participants. Il est généralement identifié sous la forme d'une mesure binaire de la réussite d'une tâche (codée 1) ou de l'échec d'une tâche (codée 0). Le taux d'achèvement d'une tâche est donc calculé en divisant le nombre de participants qui ont achevé la tâche avec succès par le nombre total de participants qui ont tenté

de l'accomplir. Par exemple, si neuf participants sur dix terminent une tâche avec succès, le taux d'achèvement est de 90%.

— *Les erreurs* : une erreur est définie comme étant toute tâche non correctement accomplie ou non accomplie. Pour chaque participant, il faut donc identifier la liste des tâches accomplies et celles non accomplies.

— *Assistance :* est définie comme toute sorte d'aide permettant de guider les participants vers la prochaine étape. Durant l'expérimentation les participants peuvent poser des questions ou demander de l'aide pour la réalisation d'une des tâches de Colla-Config. Le taux moyen d'assistance est calculé en utilisant la formule suivante : $\frac{\text{Nombre d'assistances}}{\text{Nombre de participants}}$.

- Efficience : c'est la capacité de produire le produit désiré sans gaspillage de ressources et/ou temps. Pour mesurer l'efficience il faut mesurer :

— *Temps de réalisation de tâche :* c'est le temps nécessaire pour compléter chaque tâche par chaque participant. Une fois calculé, le temps moyen de tâche est ensuite calculé pour chaque participant (ceci est utilisé pour calculer taux d'efficience).

— *Efficience du taux d'achèvement :* elle est calculée en utilisant la formule suivante : $\frac{\text{Taux moyen d'achèvement de tâche}}{\text{Temps moyen de réalisation de tâche}}$.

- Satisfaction : elle est mesurée en utilisant des résultats du post-questionnaire et de l'interview effectué avec les participants pour déduire la satisfaction en mesurant la perception des participants de la facilité d'utilisation, la facilité d'apprentissage, la facilité de mémorisation et la satisfaction subjective.
Les différentes mesures présentées ci-dessus, seront calculées et identifiées tout au long de l'expérimentation.

VI.6 Expérimentation

L'expérimentation [1] a duré 3 heures durant laquelle onze doctorants en informatique ont été volontaires à la participation. Certains ont poursuivi un cours sur les lignes de produits pendant le Master et d'autres travaillent sur l'ingénierie des lignes de produits comme axe de recherche.

Avant de commencer l'expérimentation, les participants ont pris 15 minutes pour répondre au pré-questionnaire afin de recueillir des informations sur leur niveau.
Le pré-questionnaire a montré que les participants ont des connaissances basiques de concept de configuration collaborative des lignes de produits mais ils manquent de connaissance de concept de conflit.

1. Tous les artéfacts de l'expérimentation sont disponibles via ce lien `https://bit.ly/3fFxC2t`

Afin d'initier les participants à ces concepts, une session de formation de 50 minutes a eu lieu. Durant laquelle, les concepts de l'ingénierie des lignes de produits, la configuration collaborative et les conflits ont été introduits suivis d'une présentation du principe de l'approche Colla-Config. À la fin de la formation, une petite démonstration illustrant l'utilisation de l'outil a été présentée.
La deuxième session a duré environ 30 minutes et commencé par l'installation et le paramétrage de l'outil. Ici, les participants ont reçu un document présentant une série d'étapes à suivre pour installer Colla-Config.

VI.6.1 Expérimentation partie 1

Cette première partie d'expérimentation consiste à utiliser l'outil pour configurer un produit selon l'approche Colla-Config. Après avoir installé l'outil, les participants ont commencé la première partie de l'expérimentation qui a duré 30 minutes. Durant cette session, un document avec la partie 1 de l'expérimentation a été partagé avec les participants. Il leur a été demandé d'accomplir quatre tâches. Par conséquent, un administrateur de test accompagne les participants tout au long de l'expérimentation afin de les assister et répondre à leurs questions.

Les tâches de la première partie de l'expérimentation ont consisté à : s'enregistrer, configurer le modèle des portails web, détecter et résoudre les conflits et voir le résultat final.

1. *S'enregistrer (Tâche 1-partie 1) :* les participants sont invités à introduire leurs informations personnelles et sélectionner les règles de substitution souhaitées via l'interface d'enregistrement présentée dans la Figure VI.4.
2. *Configurer le modèle de ligne des portails web (Tâche 2-partie 1) :* chaque participant sélectionne les caractéristiques désirées et non désirées en important le modèle de la ligne des portails web comme présenté dans la Figure VI.5. Le modèle des portails web configuré a été créé en utilisant VariaMos [2], un outil en ligne qui permet entre autres de modéliser des modèles de caractéristiques de manière graphique en utilisant la palette à droite comme illustré par la Figure VI.6. Le modèle créé peut être importé en format XML et utilisé pour des fins d'analyse et de configuration.
3. *Détecter et résoudre les conflits (Tâche 3-partie 1) :* cette tâche est à la charge du chef de produit qui dans un premier temps, fusionne les choix de tous les participants pour vérifier la consistance de la configuration via l'interface présentée dans la Figure VI.7. Les choix fusionnés sont synthétisés dans un tableau affichant les informations et les choix de chaque participant comme

2. `https://variamos.com/`

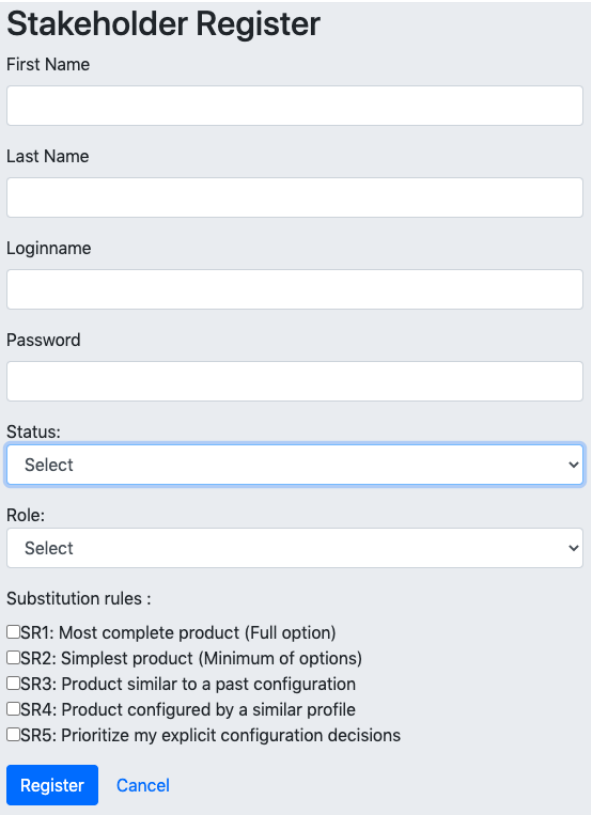


FIGURE VI.4 – Interface d'enregistrement.

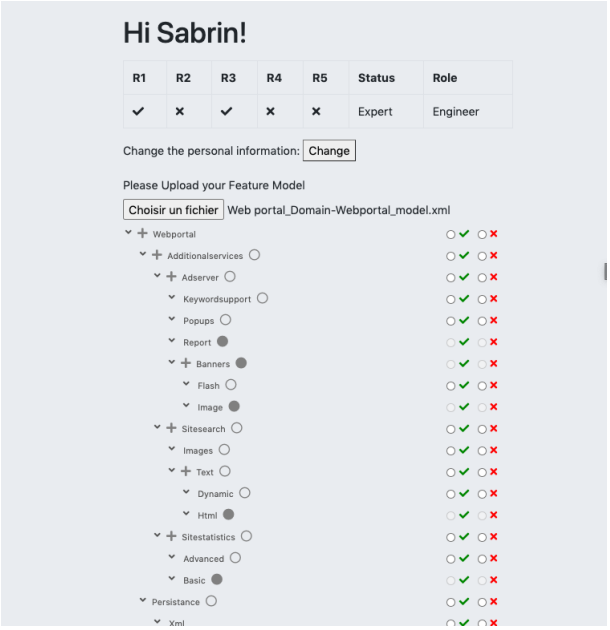


FIGURE VI.5 – Interface de configuration.

affiché sur la Figure VI.8). Le chef de produit choisit le participant priorisé et lance le processus de résolution qui résout les conflits selon l'approche proposée (cf. chapitre 5). Ensuite, la liste des MCSs calculée est affichée comme indiquée

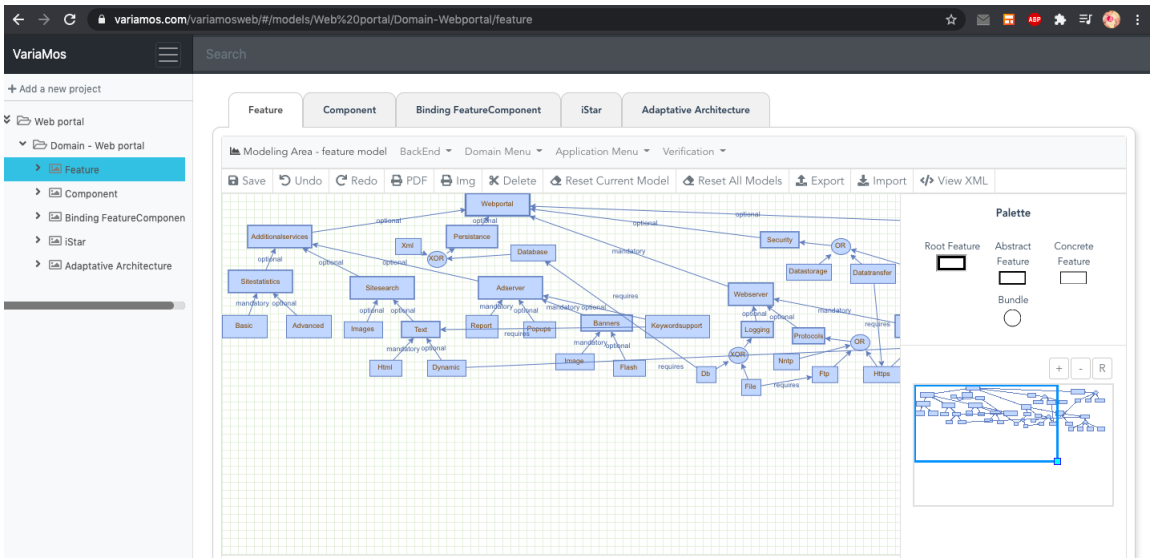


FIGURE VI.6 – Interface de création du modèle des portails web.

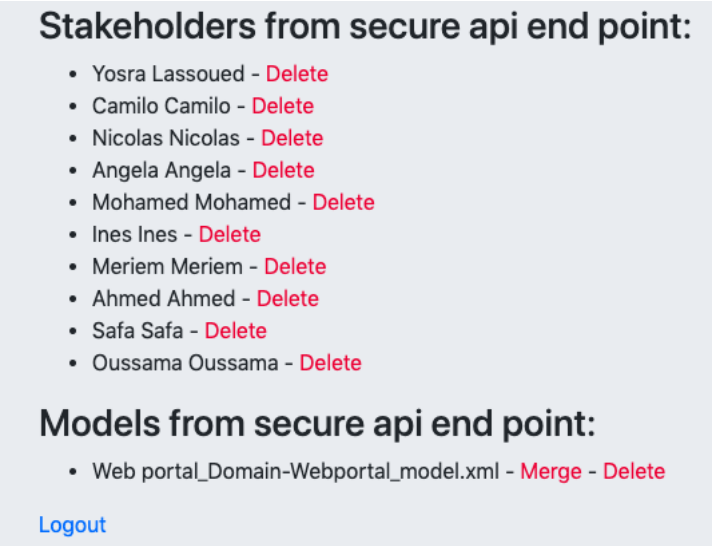


FIGURE VI.7 – Interface de fusion des choix des participants.

dans la Figure VI.9. Cette liste présente les différentes combinaisons possibles des choix conflictuels dont la suppression résout les conflits détectés. Ensuite, le MCS de résolution est retourné , comme indiqué dans la Figure VI.10, et les choix qu'il contient sont supprimés de la configuration finale qui est envoyée à tous les participants.

4. *Voir le résultat (Tâche 4-partie 1) :* finalement, les participants peuvent voir le résultat final après la résolution des conflits. Les participants accèdent de nouveau à l'espace de configuration et trouvent la configuration finale comme présentée dans la Figure VI.11.

List of stakeholders

Please select one critical stakeholder: Select

☐Send the final configuration to all the stakeholders.

First Name	Last Name	R1	R2	R3	R4	R5	Status	Role	Desired features	Undesired features
Yosra	Lassoued	✗	✗	✗	✗	✗	Novice	User	["Webportal", "Additionalservices", "Adserver", "Report", "Banners", "Image", "Webserver", "Content", "Static"]	["Keyword support", "Popups", "Flash", "Sitesearch", "Text", "Dynamic", "Html", "Sitestatistics", "Advanced", "Persistance", "Database", "Active", "Jsp", "Protocols", "Https", "Logging", "Db", "Security", "Datatransfer", "Performance", "Ms"]
Camilo	Camilo	✗	✓	✗	✗	✓	Novice	Engineer	["Webportal", "Additionalservices", "Adserver",	["Keyword support", "Popups",

FIGURE VI.8 – Interface de synthèse des choix des participants.

localhost:8080 indique

Conflict detected.
The list of MCS is:
[["Xml","Database","Active"],["Xml","Database","!Active"],["Xml","!Database","Active"],["Xml","!Database","!Active"],["!Xml","Database","Active"],["!Xml","Database","!Active"]]

OK

FIGURE VI.9 – Interface de résolution des conflits.

localhost:8080 indique

The final selected MCS is !Xml,Database,Active

OK

FIGURE VI.10 – Interface de résultat de résolution.

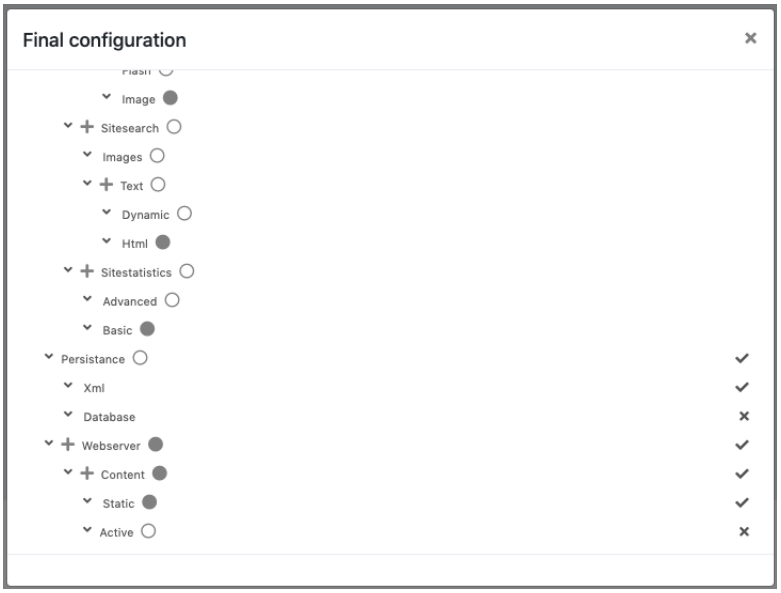


FIGURE VI.11 – Interface de configuration finale.

VI.6.2 Expérimentation partie 2

Cette deuxième partie d'expérimentation consiste à utiliser l'outil pour illustrer l'approche de (Mendonca et al., 2008) qui consiste à configurer un produit de manière séquentielle selon un plan de configuration où chaque participant configure un module du modèle des portails web. Ce principe de configuration appelé configuration par étapes a été initialement proposé par (Czarnecki et al., 2005) et amélioré ensuite par (Mendonca et al., 2008) qui propose un processus détaillé de décomposition du modèle et la création du plan de configuration.

La deuxième partie de l'expérimentation a duré 35 minutes. Un document décrivant les étapes de l'expérimentation a été partagé avec les participants.
En conséquence, trois tâches ont été demandées aux participants qui sont : attribuer les modules aux participants, configurer le module attribué et résoudre les conflits et retourner le résultat.

1. *Attribuer les modules aux participants (Tâche 1-partie 2) :* selon (Mendonca et al., 2008), le chef de produit est le responsable d'attribution des modules selon un plan de configuration. La décomposition du modèle (Figure VI.12) et le plan de configuration (Figure VI.13) proposés par (Mendonca et al., 2008) ont été adoptés dans cette expérimentation. Le participant ayant le rôle de chef de produit a affecté les modules aux participants comme indiqué par la Figure VI.13.

2. *Configurer le module attribué (Tâche 2-partie 2) :* chaque participant s'est servi de l'outil Colla-Config pour exprimer seulement les caractéristiques désirées dans le module qui lui a été attribué. Les caractéristiques non désirées ne sont

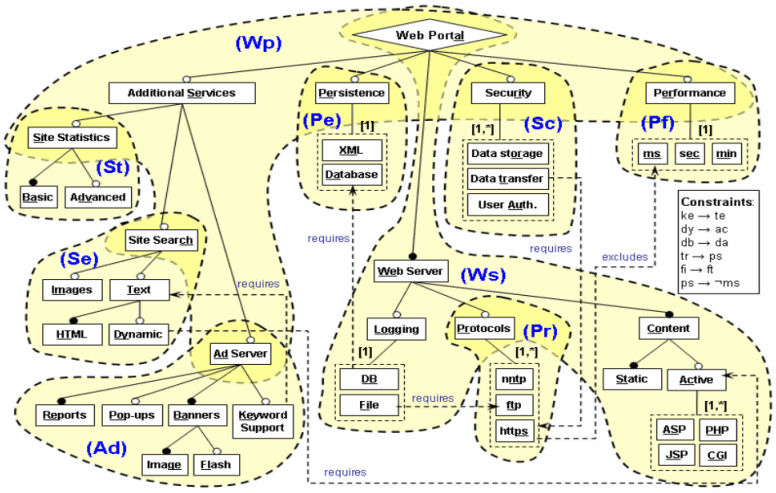


FIGURE VI.12 – Décomposition du modèle des portails web (Mendonca et al., 2008).

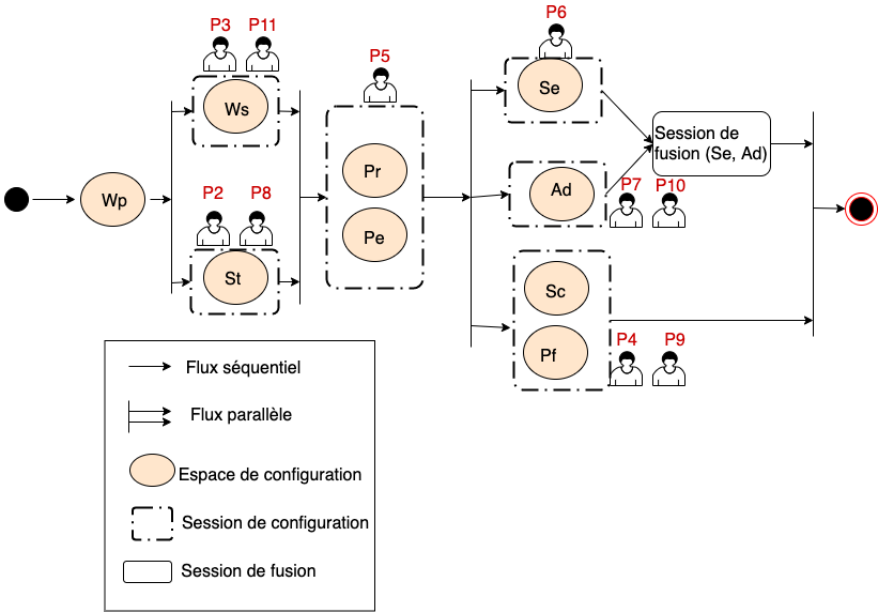


FIGURE VI.13 – Plan de configuration(Mendonca et al., 2008).

pas considérées car l'approche de (Mendonca et al., 2008) ne propose pas cette option de configuration.

3. *Résoudre les conflits (Tâche 3-partie 2) :* étant donné que le processus de configuration a un ordre prédéfini, les choix exprimés antérieurement sont superposés sur les choix ultérieures ce qui permet de résoudre les conflits de manière systématique. Dans les sessions configurées en parallèle telles que (Se) et (Ad) dans la Figure VI.13, les participants P7 et P8 responsables de la configuration de ces deux sessions ont recours à une session de fusion durant laquelle ils négocient les alternatives possibles pour résoudre le conflit détecté. à la fin du processus de configuration, le chef de produit partage un document qui

synthétise pour chaque participant la liste des choix respectés et choix non respectés.
À la fin de l'expérimentation, les participants ont pris 15 minutes pour répondre à un post-questionnaire qui contient des questions sur l'environnement expérimental, satisfaction générale, la performance de l'outil et le principe de configuration. Il leur a été également demandé de répondre en 30 minutes à quatre questions ouvertes portant sur l'utilisabilité de Colla-Config. Toutes les réponses fournies ont été analysées pour répondre à la question de recherche.

VI.7 Résultats

VI.7.1 Résultats de performance

Les onze participants ont accompli avec succès l'ensemble de sept tâches (voir deuxième colonne du Tableau VI.1). Trois tâches sont demandées au chef de produit (Participant P1) et les autres quatre tâches sont demandées aux autres dix participants comme indiqué respectivement par Tableau VI.2 et Tableau VI.3. Six participants ont accompli les tâches sans assistance(voir quatrième colonne du Tableau VI.1). Au total, six assistances ont été données aux participants, dont quatre ont été demandées par quatre différents participants (P2, P6, P7 et P8) à la tâche 2-Partie 1 (configuration du produit selon Colla-Config), qui a duré un peu plus de temps par rapport les autres tâches (voir Figure VI.14). Les deux autres assistances ont été demandées par le participant ayant le rôle de chef de produit durant l'illustration de l'approche de (Mendonca et al., 2008) notamment au niveau de la tâche 1-Partie 2 (attribuer les modules aux participants) et tâche 3-Partie 2 (résolution

Table VI.1 – Résultats de performance des participants

Participant	Taux d'achèvement des tâches	Erreur	Assistance	Efficience
P1 (C.P)	100	0	2	6,383
P2	100	0	1	26,6
P3	100	0	0	16,6
P4	100	0	0	26,6
P5	100	0	0	21,052
P6	100	0	1	23,53
P7	100	0	1	25
P8	100	0	1	28,571
P9	100	0	0	18,181
P10	100	0	0	23,529
P11	100	0	0	30,769

Table VI.2 – Durée des tâches en minutes pour le chef de produit

Participant	Tâche 3 (partie 1)	Tâche 1 (partie 2)	Tâche 3 (partie 2)	Temps moyen de tâche
P1 (C.P)	2	15	30	15,666

Table VI.3 – Durée des tâches en minutes pour les participants

Participant	Tâche1 (partie 1)	Tâche 2 (partie 1)	Tâche 4 (partie 1)	Tâche 2 (partie 2)	Temps moyen de tâche
P2	3	5	3	4	3,75
P3	3	10	3	8	6
P4	2	5	2	6	3,75
P5	4	8	5	2	4,75
P6	2	5	6	4	4,25
P7	4	4	4	4	4
P8	3	4	4	3	3,5
P9	5	7	6	4	5,5
P10	4	5	5	3	4,25
P11	3	4	4	2	3,25

des conflits) qui était la plus compliquée (le chef de produit a passé 30 minutes pour accomplir cette tâche, comme indiqué dans la Figure VI.14). La Figure VI.14 montre également que les participants passent peu de temps à élaborer les différentes tâches qui ont duré en moyenne environ 4 minutes chacune.

Selon les temps moyens d'accomplissement des différentes tâches, on peut déduire que :

- La configuration (tâche 2- partie 2) par l'approche de (Mendonca et al., 2008) est plus rapide car chaque participant ne configure qu'une partie du modèle. En revanche, la configuration (tâche 2- partie 1) par Colla-Config dure un peu plus de temps car chaque participant configure plus qu'un module du modèle.
- Dans l'approche Colla-Config, le chef de produit n'intervient qu'à la résolution des conflits (Tâche 3-partie 1) alors que dans l'approche de (Mendonca et al., 2008) il est responsable de la tâche de répartition des rôles de configuration entre les participants (Tâche 1-partie 2) vu que cette approche est basée sur un ordre prédéfini.
- La résolution des conflits dans l'approche de (Mendonca et al., 2008) est une tâche ardue car la consistance des choix de chaque participant dépend des choix du participant précédent. En effet, à chaque étape de configuration, le chef de produit vérifie les choix de configuration du participant en cours en fonction des choix antérieures.

Un résumé de résultats de performance est indiqué dans le Tableau VI.4. Le temps total moyen nécessaire pour accomplir les tâches est 17 minutes. Par conséquent, il n'y a pas eu d'erreurs durant la configuration car tous les participants ont accompli correctement toutes les tâches avec un temps moyen de 4 minutes par tâche.

Finalement, il est important de souligner que tous les participants étaient novices à la configuration collaborative des lignes de produits. Ainsi, les résultats de cette expérimentation fournissent des preuves préliminaires que Colla-Config est un outil facilement utilisable qui supporte correctement l'approche proposée (QR) et peut être également utilisé pour illustrer d'autres approches de configuration telle que celle de (Mendonca et al., 2008).

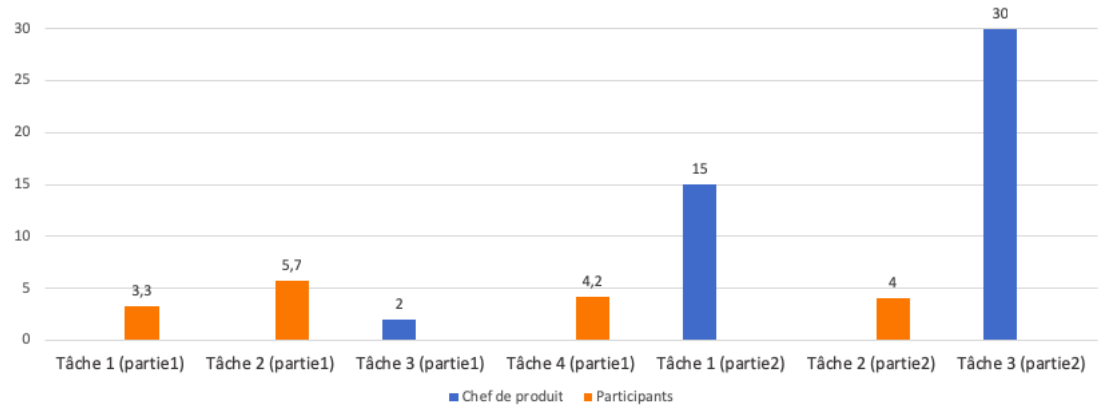


FIGURE VI.14 – Temps moyen pour accomplir chaque tâche par les participants.

TABLE VI.4 – Résumé de résultats de performance des 10 participants

Mesure	Taux d'achèvement des tâches	Temps moyen de tâche	Temps moyen total des tâches	Erreur	Assistance	Efficience
Moyenne	100.000	4,3	17,2	0,000	0,4	24,043

VI.7.2 Résultats de satisfaction

Les résultats de satisfaction ont été obtenus à partir de deux sources, le post-questionnaire et les questions de l'interview semi-structuré. Premièrement, 9 des 15 questions du post-questionnaire ont été analysées. Des notes ont été attribuées à chaque participant pour les 9 questions, sur la base de quatre dimensions d'évaluation d'utilisabilité : facilité d'apprentissage, facilité d'utilisation, facilité de mémorisation et satisfaction subjective. Les scores moyens des réponses des participants sont présentés dans le Tableau VI.5. La moyenne des réponses aux questions pour chaque dimension est présentée dans la Figure VI.15. Le résultat le plus satisfaisant concerne "la satisfaction subjective " avec une moyenne de 4,3 suivie par " la facilité d'utilisation" de Colla-Config, avec une moyenne de 4. Pour les questions qui portent sur le principe de configuration et résolution des conflits, 10 participants sur 11 ont préféré la configuration sans répartition de rôles et sans ordre prédéfini (principe Colla-Config) qu'à la configuration basée sur workflow (principe l'approche de (Mendonca et al., 2008)). En outre, 9 participants sur 11 ont choisi la stratégie de résolution des conflits de Colla-Config. Les deux participants qui ont choisi le principe de résolution selon l'ordre de configuration avaient les choix respectés.
Pour les résultats des autres questions du post-questionnaire, ils sont explorés pour identifier les menaces à la validité présentées dans la section suivante.
Les réponses aux questions de l'interview semi-structuré ont montré qu'en général, les participants ont apprécié la simplicité de l'outil, la facilité de son utilisation et son potentiel à illustrer l'approche Colla-Config. Ils ont mentionné que c'est une bonne stratégie de considérer les préférences des utilisateurs dans le processus de ré-

Table VI.5 – Scores moyens de satisfaction des participants

Participant	Facilité d'apprentissage	Satisfaction subjective	Facilité d'utilisation	Facilité de mémorisation
P1	4	4,33	3,5	4
P2	3,5	4,66	5	4
P3	4	4,33	4	3,5
P4	4	5	4,5	4
P5	4	4,33	3,5	3
P6	4	4,33	4,5	4
P7	4	4	3,5	3
P8	4	4,33	5	3,5
P9	3,5	4	4	3
P10	3,5	3,66	3,5	3
P11	3,5	4,33	3	3
Moyenne	3,81	4,3	4	3,45

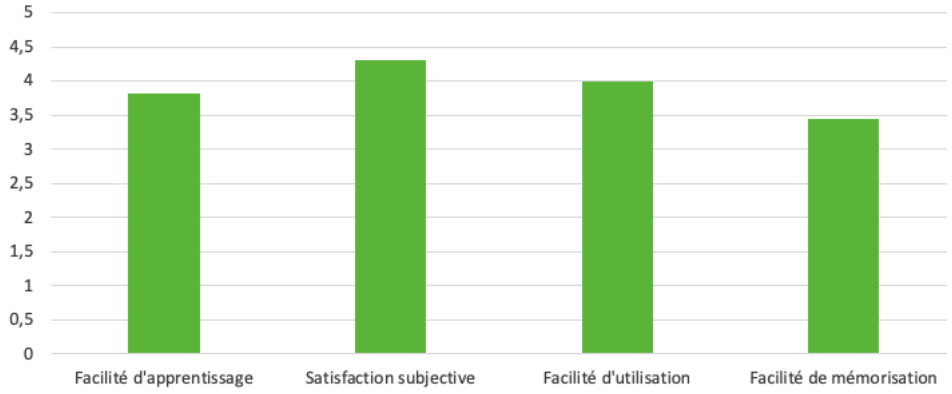


Figure VI.15 – Résultats des questions de satisfaction.

solution des conflits. Quelques recommandations ont également été proposées par les participants pour améliorer l'outil : (i) l'interactivité : parmi les points principaux évoqués par les participants qui proposent d'améliorer l'outil pour qu'il permette aux utilisateurs d'interagir en temps réel et partager leurs choix à chaque étape de configuration. (ii) la gestion des conflits : certains participants ont proposé de fournir des informations explicites pour chaque participants lui permettant de savoir la liste de ses choix respectés et la liste des choix qui ne le sont pas. Une autre proposition est de donner aux utilisateurs la possibilité de réviser leurs choix avant de procéder à la résolution automatique ce qui permet de réduire le nombre de conflits détectés. (iii) Accessibilité : la plupart des participants ont également proposé de transformer l'outil en application web ce qui permettra d'utiliser des téléphones portables et/ou des tablettes pour ouvrir l'application et éviter certains paramétrages et installation de logiciel.

VI.8 Les menaces à la validité

La validité des résultats obtenus lors d'une expérience dépend des facteurs liés aux paramètres de l'expérience. Dans ce cas, quatre types de menaces à la validité

proposés par (Cook and Campbell, 1979) sont identifiés :

- **La validité de la conclusion** concerne la relation entre le traitement et le résultat et vérifie s'il existe une relation statistique entre eux.
- **La validité interne** concerne la relation entre le traitement et le résultat et vérifie s'il existe un lien de causalité entre eux.
- **La validité conceptuelle** concerne la relation entre la théorie et l'observation et vérifie si le traitement reflète la cause de conception et le résultat reflète l'effet de conception.
- **La validité externe** est la généralisation des résultats de l'étude pour d'autres études en vérifiant la relation causale entre la cause et l'effet de conception et entre le traitement et les résultats.

VI.8.1 La validité de la conclusion

Les menaces de validité de la conclusion consistent à tirer les bonnes conclusions sur les résultats d'un traitement appliqué à une expérience (Wohlin et al., 2012). Ces menaces sont dues à la faiblesse des tests statistiques. Toutefois, l'expérimentation menée est de nature qualitative et non quantitative. De plus, étant donné que l'objectif principal de l'étude est d'étudier le comportement et les opinions des utilisateurs d'un outil, les méthodes de recherche qualitative sont bien adaptés. L'analyse des données recueillies dépend encore de notre interprétation.

VI.8.2 La validité interne

La validité interne fait référence à l'influence des variables externes de l'expérience sur la conception et les résultats de l'expérience. Dans ce cas, cette menace potentielle pour la validité a été atténuée par la définition et la validation de notre protocole expérimental en suivant attentivement un processus de test d'utilisabilité bien structuré en utilisant le format SO/CEI CIF pour les tests d'utilisabilité(ISO/IEC, 2006). La validité interne dans ce cas concerne les menaces suivantes :

- **Le nombre de participants.** Le nombre de participants peut sembler relativement faible. Cependant, le ISO/CEI CIF pour les tests d'utilisabilité indique que "huit participants ou plus sont recommandés(ISO/IEC, 2006)".
- **La taille du modèle.** Le modèle configuré est un modèle basique qui a été choisi pour deux raisons. (i) le modèle a également été utilisé dans l'approche de configuration collaborative proposée par (Mendonca et al., 2008) pour expliquer le principe de décomposition d'un modèle et la création du plan de

configuration correspondant. Étant donné que la deuxième partie de l'expérimentation consiste à configurer un produit selon cette approche, il a été préférable d'adopter les mêmes décomposition/plan pour s'assurer que l'approche a été correctement appliquée. (ii) les participants sont des novices à la configuration collaborative, un modèle basique leur convient pour qu'ils puissent comprendre son contenu et pouvoir le configurer le plus simple possible.

VI.8.3 La validité conceptuelle

La validité conceptuelle désigne la généralisation des résultats de l'expérimentation à un concept ou une théorie (Wohlin et al., 2012). La validité conceptuelle dans ce cas concerne les menaces suivantes :

- **L'insuffisance d'explication pré-opérationnelle des concepts.** Afin d'éviter cette menace, tous les concepts nécessaires pour comprendre les différentes étapes de l'expérimentation ont été bien définis. Les participants ont été également accompagnés tout au long de l'expérimentation pour répondre à leurs questions.
- **L'insuffisance des compétences pour exécuter les tâches.** Cette menace a été écartée par les résultats du pré-questionnaire des participants et la formation dédiée.
- **L'attente des participants.** Les attentes des participants pourrait biaiser les résultats. Cette menace été écartée par les questions variées du post-questionnaire qui portent sur des dimensions différentes d'évaluation d'utilisabilité.

VI.8.4 La validité externe

La validité externe est la capacité de généraliser les résultats de l'expérience à d'autres situations et à une population plus large que celle de l'expérience elle-même. Cette menace a été atténuée par le fait que ISO/CEI CIF (ISO/IEC, 2006) a précisé que 8 participants ou plus sont recommandés pour mener l'expérimentation. Toutefois, les résultats obtenus représentent les 11 participants. Si on augmente le nombre de population, les résultats dépendent toujours des opinions des participants sur l'outil et leurs attentes par rapport le produit final.

VI.9 Conclusion

Ce chapitre a présenté une description de l'architecture de l'outil Colla-Config. Puis, il a présenté l'application de l'approche sur le cas réel du portail web par le biais de l'outil implémenté. Ensuite, une analyse des résultats expérimentaux a été effectuée. Ces expérimentations ont été réalisées dans un contexte d'évaluation de l'utilisabilité selon la norme ISO/CEI CIF. L'application de l'approche Colla-Config sur l'exemple du portail web a montré sa faisabilité et les performances de l'outil en terme de temps de réponse et simplicité d'utilisation.

VII

Conclusion

Sommaire

VII.1Contributions 105

VII.2Perspectives 106

VII.1 Contributions

La thèse a abordé la prise en compte des préférences des parties prenantes dans la résolution des conflits lors de la configuration collaborative d'une ligne de produits. Elle a fourni une réponse aux problématiques relatives à la configuration collaborative qui ont été discutées dans l'introduction de ce mémoire.

L'approche proposée Colla-Config (Collaborative Configuration) présente (i) une définition formelle de la notion de conflit de configuration, (ii) une classification des conflits de configuration, (iii) un algorithme de détection et résolution des conflits, et (vi) un processus flexible pour la configuration collaborative.

Les travaux de cette thèse s'inscrivent dans un courant qui cherche à considérer les préférences des parties prenantes lors de la configuration collaborative des lignes de produits notamment dans la résolution des conflits. A l'instar de ces approches, l'approche Colla-Config propose une stratégie basée sur les préférences pour résoudre les conflits de configuration. Cependant, Colla-Config se distingue par l'expression riche et explicite des préférences et l'aboutissement de son processus de résolution peu importe le nombre et le type de conflit.

Les règles de substitution proposées peuvent être combinées afin d'enrichir leur pouvoir d'expression des préférences. La résolution consiste à résoudre un problème de satisfaction des contraintes où les choix de configuration des parties prenantes ainsi que les contraintes de la LP sont formalisés en tant que problème SAT. Ce dernier, est résolu en calculant les MCSs et en y appliquant les règles choisies suivant un schéma d'application spécifique pour identifier le MCS de résolution.

Dans l'approche Colla-Config, la correspondance entre les règles et les MCSs montre leur richesse sémantique en identifiant l'ensemble des MCSs potentiels pour la résolution des conflits détectés. Cette correspondance ne sert pas seulement à aider à spécifier les MCSs, mais également à assurer que le MCS de résolution est issu d'une prise en compte systématique et méthodologique des différentes préférences exprimées. En effet, à chaque règle de substitution correspond une sorte de MCS particulier traduisant la préférence en un semble de choix de configuration à abandonner. À l'issu de cette correspondance, l'ensemble de MCSs obtenu est étudié selon un schéma spécifique d'application des règles. Ce schéma traite tous les cas de figure résultants de l'application des règles sur l'ensemble de MCSs (MCS totalement en commun, partiellement en commun, aucun MCS en commun..).

L'approche Colla-Config a été évaluée dans le cadre d'une expérimentation contrôlée menée au sein de Centre de Recherche en Informatique (CRI), ce qui a permis de montrer sa faisabilité.

VII.2 Perspectives

Le travail présenté dans cette thèse peut être poursuivi dans plusieurs directions :

VII.2.1 Outillage

L'approche Colla-Config peut être outillée de manière à rendre la configuration collaborative répartie et en temps réel pour les utilisateurs. Ceci peut être réalisé sur deux plans :

- **Répartition :** la répartition consiste à proposer un outil qui supporte la configuration distribuée, où les utilisateurs peuvent entretenir la configuration tout en étant géographiquement délocalisés. Pour ce faire, l'outil doit supporter une architecture client-serveur : côté client représente l'interface dédiée aux utilisateurs pour exprimer et envoyer leur choix de configuration, côté serveur sera dédiée au chef de produit pour traiter les choix des configuration et résoudre les conflits détectés à l'instar de Colla-Config où l'analyse de la configuration et la résolution des conflits sont traités dans l'interface du chef de produits.
- **Processus de configuration en temps réel :** Ceci consiste à l'implémentation d'un outil qui permet la configuration selon l'approche Colla-Config mais en temps réel où les utilisateurs partagent instantanément les différents états du processus de configuration. Ceci permet aux utilisateurs d'être au courant des différents choix de configuration et leur prévient en temps réel de tout conflit détecté.

VII.2.2 Extension de l'approche Colla-Config

L'approche Colla-Config ouvre la voie vers l'amélioration de la technique de résolution proposée à deux niveaux :

- **Au niveau des règles de substitution :** il est possible d'ajouter de nouvelles règles qui considèrent les préférences en terme d'autres critères autre que ceux considérés dans Colla-Config. De plus, les règles proposées peuvent être enrichies en tenant compte de manière dynamique de l'aspect comportemental des utilisateurs durant le processus de configuration. En outre, les informations pertinentes peuvent être capitalisées sous forme d'ontologie et serviront

à déduire de manière automatique les préférences des utilisateurs. Ceci dit qu'au lieu de solliciter les utilisateurs à choisir les règles, ces dernières seront automatiquement sélectionnés selon le comportement de chaque utilisateur.

- **Au niveau de méthode de calcul de MCSs :** dans Colla-Config, l'ensemble de MCSs a été calculé en utilisant l'approche de (Liffiton and Sakallah, 2008), notamment l'algorithme qui permet d'identifier l'ensemble de MUSs (correspond aux conflits) et en déduire la liste de MCSs. Il est également possible d'utiliser d'autres algorithmes de calcul de MCSs identifés dans la littérature tels que : (Previti and Marques-Silva, 2013), (Mencía et al., 2015), (Liffiton et al., 2015) et (Bacchus and Katsirelos, 2015) etc. Ces differents travaux sont des exentions de l'approche principale proposée par (Liffiton and Sakallah, 2008) et adoptée dans cette thèse.

VII.2.3 Évaluation

L'approche Colla-Config a été expérimentée sur un modèle réel de ligne de produits (Ligne des protails web) en suivant un processus d'évaluation d'utilisabilité selon la norme (ISO/IEC, 2006) avec un groupe de 11 doctorants. D'autres expérimentations devraient être menées dans un cadre industriel pour évaluer la scalabilité.

VII.2.4 Intégration aux approches existantes

L'approche Colla-Config peut être intégrée à différentes approches de configuration collaborative telles que : l'approche de (Junior et al., 2011)qui emploie des agents mobiles pour assurer la communication entre les utilisateurs. Ces agents peuvent être également adaptés pour communiquer les règles de substitution aux utilisateurs et leur retourner le MCS de résolution. L'approche de (Stein et al., 2014) propose une stratégie de résolution des conflits basée sur les préférences. Colla-Config peut être intégrée à cette approche en combinant les stratégies d'élicitation des préférences pour identifier le MCS de résolution. L'approche de (Xiong et al., 2012) propose un algorithme qui calcule les valeurs de correction d'une configuration inconsistante. Il est possible que ces valeurs soient identifiés autrement à travers le calcul des MCSs selon les préférences comme proposée par Colla-Config. Ainsi, l'approche Colla-Config sera complétée par la formalisation de mode de l'interaction des utilisateurs durant la configuration et offrira en contre partie la capacité à exprimer les préférences de configuration de manière explicite et résoudre les conflits selon ces préférences.

VII.2.5 Intégration à des outils existants

L'approche Colla-Config peut être également intégrée à différents outils de configuration de lignes de produits afin d'assurer la configuration collaborative et renforcer les mécanismes de résolution des conflits. Par exemple, SPLOT [1](Software Product Lines Online Tools) un outil dévéloppé par (Mendonça et al., 2009) qui permet la configuration interactive où un utilisateur prend des décisions de configuration sur un modèle de caractéristiques et le moteur de configuration propage automatiquement ces décisions pour assurer leur cohérence. Colla-Config peut être intégrée à cet outil afin de permettre la configuration collaborative où les modèles de caractéristiques générés automatiquement par SPLOT peuvent être directement configurés de manière collaborative. Aussi, elle peut être intégrée aux outils d'analyse et de modélisation de variabilité des LdPs afin d'apporter les moyens techniques pour opérationnaliser la configuration collaborative des produits. En effet, Colla-Config supporte la configuration des modèles de LdPs générés par VariaMos [2], un outil web de modélisation de variabilité et de raisonnement sur les modèles de LdPs (Mazo et al., 2015). Des travaux d'intégration de Colla-Config dans la version web de VariaMos devraient être poursuivis.

1. `http://www.splot-research.org/`
2. `https://variamos.com`

Bibliographie

Abri, A. A., Jamoussi, Y., Kraiem, N., and Khanjari, Z. A. (2016). Comprehensive classification of collaboration approaches in e-learning. *Telematics and Informatics*.

Afonso, N., Galindo, J. A., and Benavides, D. (2020). Configuration assisted through conversational agents (chatbots and voicebots). In *Proceedings of the Workshop On Configuration*, CONFWS'20, page 29–34.

Afzal, U., Mahmood, T., and Shaikh, Z. (2016). Intelligent software product line configurations. *Comput. Stand. Interfaces*, 48(C) :30–48.

Andrey, L., Festor, O., and Lahmadi, A. (2006). Evaluation du passage à l'échelle des systèmes de gestion : métriques et modèles. In *Colloque Francophone sur l'Ingénierie des Protocoles - CFIP 2006*, Tozeur/Tunisia. Eric Fleury and Farouk Kamoun, Hermes.

Arrow, K., Sen, A. K., and Suzumura, K., editors (2002). *Handbook of Social Choice and Welfare*, volume 1. Elsevier, 1 edition.

Atkinson, C., Bayer, J., Bunse, C., Kamsties, E., Laitenberger, O., Laqua, R., Muthig, D., Paech, B., Wüst, J., and Zettel, J. (2002). *Component-Based Product Line Engineering with UML*. Addison-Wesley Longman Publishing Co., Inc., USA.

Baader, F., Calvanese, D., Mcguinness, D., Nardi, D., and Patel-Schneider, P. (2007). *The Description Logic Handbook : Theory, Implementation, and Applications*. Cambridge University Press, New York, USA.

Bacchus, F. and Katsirelos, G. (2015). Using minimal correction sets to more efficiently compute minimal unsatisfiable sets. In Kroening, D. and Păsăreanu, C. S., editors, *Computer Aided Verification*, pages 70–86, Cham. Springer International Publishing.

Bachmeyer, R. C. and Delugach, H. S. (2007). A conceptual graph approach to feature modeling. In *Proceedings of the 15th International Conference on Concep-*

tual Structures : Knowledge Architectures for Smart Applications, ICCS '07, page 179–191, Berlin, Heidelberg. Springer-Verlag.

Bagheri, E., Di Noia, T., Ragone, A., and Gasevic, D. (2010). Configuring software product line feature models based on stakeholders' soft and hard requirements. In *Software Product Lines : Going Beyond, the 14th International Conference*, pages 16–31, Berlin, Heidelberg. Springer-Verlag.

Batory, D., Benavides, D., and Ruiz-Cortes, A. (2006). Automated analysis of feature models : Challenges ahead. *Commun. ACM*, 49(12) :45–47.

Bingliang, Y., Renwang, L., and Xianmei, W. (2010). Study on product collaborative configuration design platform for mass customization. In *2010 2nd IEEE International Conference on Information Management and Engineering*, pages 489–493.

Camarinha-Matos, L. and Afsarmanesh, H. (2008). *Collaborative networks : Reference modeling*. New York, NY : Springer.

Cerón, R., Arciniegas, J. L., Ruiz, J. L., Dueñas, J. C., Bermejo, J., and Capilla, R. (2004). Architectural modelling in product family context. In Oquendo, F., Warboys, B. C., and Morrison, R., editors, *Software Architecture*, pages 25–42, Berlin, Heidelberg. Springer Berlin Heidelberg.

Chen, S., Liu, L., and Tseng, M. M. (2009). Product configuration via negotiation for mass customization : An interactive goal programming approach. In *Industrial Engineering and Engineering Management, 16th International Conference*, pages 999–1003.

Clements, P. and Northrop, L. (2001). *Software Product Lines : Practices and Patterns*. Longman Publishing.

Cook, T. and Campbell, D. (1979). *Quasi-Experimentation : Design and Analysis Issues for Field Settings*. Houghton Mifflin.

Czarnecki, K., Helsen, S., and Eisenecker, U. (2005). Staged configuration through specialization and multilevel configuration of feature models. In *Software Process : Improvement and Practice*.

Daoud, M., Tamine, L., Boughanem, M., and Chebaro, B. (2008). Construction des profils utilisateurs à base d'une ontologie pour une recherche d'information personnalisée. In *COnférence en Recherche d'Infomations et Applications - CORIA 2008, 5th French Information Retrieval Conference, Trégastel, France, March 12-14, 2008. Proceedings*, pages 225–240.

de Groote, X. (1994). The flexibility of production processes : A general framework. *Management Science*, 40(7) :933–945.

Deelstra, S., Sinnema, M., and Bosch, J. (2005). Product derivation in software product families : a case study. *Journal of Systems and Software*, 74(2) :173 – 194. The new context for software engineering education and training.

Dhungana, D., Grünbacher, P., and Rabiser, R. (2011). The dopler meta-tool for decision-oriented variability modeling : A multiple case study. *Automated Software Engg.*, 18(1) :77–114.

Djebbi, O. (2011). *L'ingénierie des exigences par et pour les lignes de produits*. PhD thesis, Université Paris1 Panthéon-Sorbonne.

Djebbi, O. and Salinesi, C. (2007). Red-pl, a method for deriving product requirements from a product line requirements model. In *Proceedings of the 19th international conference on Advanced information systems engineering*, pages 279–293.

Dou, R., Zhang, Y., and Nan, G. (2016). Customer-oriented product collaborative customization based on design iteration for tablet personal computer configuration. *Computers and Industrial Engineering*, 99 :474–486.

Edded, S., Sassi, S. B., Mazo, R., Salinesi, C., and Ghezala, H. B. (2019). Collaborative configuration approaches in software product lines engineering : A systematic mapping study. *Journal of Systems and Software*, 158 :110422.

Edded, S., Sassi, S. B., Mazo, R., Salinesi, C., and Ghezala, H. B. (2020). Preference-based conflict resolution for collaborative configuration of product lines. In Ali, R., Kaindl, H., and Maciaszek, L. A., editors, *Proceedings of the 15th International Conference on Evaluation of Novel Approaches to Software Engineering, ENASE 2020, Prague, Czech Republic, May 5-6, 2020*, pages 297–304. SCITEPRESS.

Flege, O. (2000). System family architecture description using the uml. Technical report.

Franzago, M., Ruscio, D. D., Malavolta, I., and Muccini, H. (2017). Collaborative model-driven software engineering : A classification framework and a research map. In *Proceedings of the IEEE 40th International Conference on Software Engineering*.

Gauch, S., Chaffee, J., and Pretschner, A. (2003). Ontology-based personalized search and browsing. *Web Intelli. and Agent Sys.*, 1(3-4) :219–234.

Ghédira, K. (2012). *Constraint Satisfaction Problems : CSP Formalisms and Techniques*. ISTE Ltd and John Wiley & Sons Inc.

Griss, M. L., Favaro, J., and Alessandro, M. d. (1998). Integrating feature modeling with the rseb. In *Proceedings of the 5th International Conference on Software Reuse*, ICSR '98, page 76, USA. IEEE Computer Society.

Gurp, J. V., Bosch, J., and Svahnberg, M. (2001). On the notion of variability in software product lines. In *Proceedings of the Working IEEE/IFIP Conference on Software Architecture*, WICSA '01, page 45, USA. IEEE Computer Society.

Holl, G., Grunbacher, P., Elsner, C., and Klambauer, T. (2012). Supporting awareness during collaborative and distributed configuration of multi product lines. In *Proceedings of the 19th Asia-Pacific Software Engineering Conference - Volume 01*, APSEC '12, pages 137–147, Washington, DC, USA. IEEE Computer Society.

Hubaux, A., Classen, A., and Heymans, P. (2009). Formal modelling of feature configuration workflow. In *Proceedings of the 13th International Software Product Lines Conference (SPLC'09)*, pages 221–230, San Francisco, CA, USA. Carnegie Mellon University.

Hubaux., A., Heymans, P., Schobbens, P., and Deridder, D. (2010). Towards multi-view feature-based configuration. In *Proceedings of the 16th International Working Conference on Requirements Engineering :Foundation for Software Quality (REFSQ'10)*, pages 106–112, Essen, Germany. Springer-Verlag.

Hubaux, A. and P.Heymans (2009). On the evaluation and improvement of feature-based configuration techniques in software product lines. In *Proceeding of the 31st International Conference on Software Engineering (ICSE'09), Companion Volume, Doctoral Symposium*, pages 367–370, Vancouver, BC, Canada. IEEE.

ISO/IEC (2006). Software engineering—software product quality requirements and evaluation (square). Technical report, Common Industry Format (CIF) for usability test reports.

Jarke, M., Mylopoulos, J., Schmidt, J. W., and Vassiliou, Y. (1992). DAIDA : an environment for evolving information systems. *ACM Trans. Inf. Syst.*, 10(1) :1–50.

Jarke, M., Pohl, K., Jacobs, S., Jr., J. A. B., Assenova, P., Holm, P., Wangler, B., Rolland, C., Plihon, V., Schmitt, J., Sutcliffe, A. G., Jones, S., Maiden, N. A. M., Till, D., Vassiliou, Y., Constantopoulos, P., and Spanoudakis, G. (1993). Requirements engineering : An integrated view of representation, process, and domain. In *Software Engineering - ESEC '93, 4th European Software Engineering Conference, Garmisch-Partenkirchen, Germany, September 13-17, 1993, Proceedings*, pages 100–114.

Jennings, N. R. (2001). An agent-based approach for building complex software systems. *Commun. ACM*, 44(4) :35–41.

Jézéquel, J.-M. (1998). Reifying configuration management for object-oriented software. In *Proceedings of the 20th International Conference on Software Engineering*, ICSE '98, page 240–249, USA. IEEE Computer Society.

Jézéquel, J.-M. (1999). Reifying variants in configuration management. *ACM Trans. Softw. Eng. Methodol.*, 8(3) :284–295.

Johannesson, P. and Perjons, E. (2014). *An Introduction to Design Science*. Springer International Publishing.

Junior, C. M., Cirilo, E., and Lucena, C. (2011). Assisted user-guidance in collaborative and dynamic software product line configuration. pages 143–156.

Kang, K., Cohen, S., Hess, J., Novak, W., and Peterson, A. (1990). Feature-oriented domain analysis (foda) feasibility study. Technical Report CMU/SEI-90-TR-021, Software Engineering Institute.

Kang, K. C., Kim, S., Lee, J., Kim, K., Shin, E., and Huh, M. (1998). Form : A feature- ;oriented reuse method with domain- ;specific reference architectures. *Annals of Software Engineering*, 5 :143–168.

Kang, K. C., Lee, K., Lee, J., and Kim, S. (2003). Feature-oriented product line software engineering : Principles and guidelines. In *Pohang University of Science and Technology*.

Khalil, K. M., Abdel-Aziz, M., Nazmy, T. T., and Salem, A. (2014). Intelligent techniques for resolving conflicts of knowledge in multi-agent decision support systems. *ArXiv*, abs/1401.4381.

Kitchenham, B. A. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical report, Keele University and Durham University.

Kostadinov, D. (2007). *Personnalisation de l'information : une approche de gestion de profils et de reformulation de requêtes. (Data Personalization : an approach for profile management and query reformulation).* PhD thesis, Versailles Saint-Quentin-en-Yvelines University, France.

Liffiton, M. H., Previti, A., Malik, A., and Marques-Silva, J. (2015). Fast, flexible mus enumeration. *Constraints*, 21 :223–250.

Liffiton, M. H. and Sakallah, K. A. (2008). Algorithms for computing minimal unsatisfiable subsets of constraints. *J. Autom. Reason.*, 40(1) :1–33.

Marques-Silva, J., Heras, F., Janota, M., Previti, A., and Belov, A. (2013). On computing minimal correction subsets. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pages 615–622. AAAI Press.

Mazo, R. (2014). Avantages et limites des modèles de caractéristiques dans la modélisation des exigences de variabilité. *Journal "Génie Logiciel"*, 111 :42–48.

Mazo, R., Grünbacher, P., Heider, W., Rabiser, R., Salinesi, C., and Diaz, D. (2011). Using constraint programming to verify dopler variability models. In *Procee-*

dings of the 5th Workshop on Variability Modeling of Software-Intensive Systems, VaMoS '11, page 97–103, New York, NY, USA. Association for Computing Machinery.

Mazo, R., MuÃ±oz-FernÃ¡ndez, J., Rincon, L., Salinesi, C., and Tamura, G. (2015). Variamos : an extensible tool for engineering (dynamic) product lines. pages 374–379.

Mencía, C., Previti, A., and Marques-Silva, J. (2015). Literal-based mcs extraction. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 1973–1979. AAAI Press.

Mendonca, M., Bartolomei, T., and Cowan, D. (2008). Decision-making coordination in collaborative product configuration. In *ACM symposium on applied computing*, pages 108–113.

Mendonça, M., Branco, M., and Cowan, D. D. (2009). S.P.L.O.T. : software product lines online tools. In *Companion to the 24th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA 2009, October 25-29, 2009, Orlando, Florida, USA*, pages 761–762.

Mendonca, M., Cowan, D., and Oliveira, T. (2007). Process-centric approach for coordinating product configuration decisions. In *Proceedings of the 40th Hawaii International Conference on System Sciences (HICSS'07)*, pages 1–10.

Montanari, U. and Rossi, F. (1988). *Fundamental Properties of Networks of Constraints : A New Formulation*, pages 426–449. Springer New York, New York, NY.

Morgado, A., Liffiton, M., and Marques-Silva, J. (2013). Maxsat-based mcs enumeration. In *Proceedings of the 8th International Conference on Hardware and Software : Verification and Testing*, HVC'12, pages 86–101, Berlin, Heidelberg. Springer-Verlag.

Nurcan, S. (2004). A conceptual framework for intention driven flexible workflow modeling. In *The fifth Workshop on Business Process Modelling, Development, and Support (BPMDS'04)*, Riga, Latvia.

Ochoa, L., González-Rojas, O., and Thüm, T. (2015). Using decision rules for solving conflicts in extended feature models. In *Proceedings of the 2015 ACM SIGPLAN International Conference on Software Language Engineering*, SLE 2015, pages 149–160, New York, NY, USA. ACM.

Ochoa, L. and González-Rojas, O. (2016). Program synthesis for configuring collaborative solutions in feature models. pages 98–108.

Osman, A., Phon-Amnuaisuk, S., and Ho, C. K. (2009). *Investigating Inconsistency Detection as a Validation Operation in Software Product Line*, pages 159–168. Springer Berlin Heidelberg, Berlin, Heidelberg.

Ouali, S., Kraiem, N., and Ghezala, H. B. (2011). Framework for evolving software product line. *International Journal of Software Engineering and Applications (IJSEA)*, 2(2).

Pereira, J. A. (2017). Runtime collaborative-based configuration of software product lines. In *Proceedings of the 39th International Conference on Software Engineering Companion*, ICSE-C '17, pages 94–96.

Pereira, J. A., P.Matuszyk, S.Krieter, M.Spiliopoulou, and G.Saake (2018). Personalized recommender systems for product-line configuration processes. *Computer Languages, Systems and Structures*, 54 :451–471.

Petersen, K., Feldt, R., Mujtaba, S., and Mattsson, M. (2008). Systematic mapping studies in software engineering. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*.

Petersen, K., Vakkalanka, S., and Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering : An update. *Information and Software Technology*, 64 :1–18.

Pohl, K., Bockle, G., and Linden, F. (2005). *Software Product Line Engineering : Foundations, Principles, and Techniques*.

Previti, A. and Marques-Silva, J. (2013). Partial mus enumeration. In *AAAI*.

Prieto-Diaz, R. and Freeman, P. (1987). Classifying software for reusability. *IEEE Software*, 4(1) :6–16.

Rabiser, R., Grunbacher, P., and Dhungana, D. (2010a). Requirements for product derivation support : Results from a systematic literature review and an expert survey. *Information and Software Technology*.

Rabiser, R., GrÃŒnbacher, P., and Holl, G. (2010b). Improving awareness during product derivation in multi-user multi product line environments. In *Automated Configuration and Tailoring of Applications, the 1st International Workshop*, pages 1–5.

Rabiser, R., Wolfinger, R., and Grünbacher, P. (2009). Three-level customization of software products using a product line approach. In *Proceedings of the 42nd IEEE Annual Hawaii International Conference on System Sciences*, pages 1–10.

Riebisch, M., Bollert, K., Streitferdt, D., and Philippow, I. (2002). Extending feature diagrams with uml multiplicities. In *Proceedings of the 6th World Conference on Integrated Design and Process Technology*.

Rinconn, L., Giraldo, G., Mazo, R., Salinesi, C., and Diaz, D. (2015). Method to identify corrections of defects on product line models. *Electronic Notes in Theoretical Computer Science*, 314 :61 – 81. CLEI 2014, the XL Latin American Conference in Informatic.

Rolland, C. (1998). A comprehensive view of process engineering. In *Proceedings of the 10th International Conference on Advanced Information Systems Engineering*, CAiSE '98, pages 1–24, Berlin, Heidelberg. Springer-Verlag.

Rolland, C. (2007). Capturing system intentionality with maps. In *Conceptual Modelling in Information Systems Engineering*. Springer-Verlag.

Rolland, C., Achour, C. B., Cauvet, C., Ralyte, J., Sutcliffe, A., Maiden, N., Jarke, M., Haumer, P., Pohl, K., Dubois, E., and Heymans, P. (1998). A proposal for a scenario classification framework. *Requirements Engineering*, 3 :23–47.

Roschelle, J. and D.Teasley, S. (1995). The construction of shared knowledge in collaborative problem solving. pages 69–197.

Sabin, D. and Weigel, R. (1998). Product configuration frameworks-a survey. *IEEE Intelligent Systems*, 13(4) :42–49.

Salinesi, C., Mazo, R., Diaz, D., and Djebbi, O. (2010). Using integer constraint solving in reuse based requirements engineering. In *Proceedings of the 18th IEEE International Requirements Engineering Conference (RE)*, pages 243–251.

Schafer, J., D.Frankowski, Herlocker, J., and S.Sen (2007). *Collaborative Filtering Recommender Systems*, pages 291–324. Springer Berlin Heidelberg, Berlin, Heidelberg.

Seffah, A., Donyaee, M., Kline, R., and Padda, H. (2006). Usability measurement and metrics : A consolidated model. *Software Quality Journal*, 14 :159–178.

Soltani, S., Asadi, M., Gašević, D., Hatala, M., and Bagheri, E. (2012). Automated planning for feature model configuration based on functional and non-functional requirements. In *Software Product Line, the 16th International conference*, SPLC '12, pages 56–65, New York, NY, USA. ACM.

Sowa, J. F. (1984). *Conceptual Structures : Information Processing in Mind and Machine*. Addison-Wesley Longman Publishing Co., Inc., USA.

Standardization, I. O. F. (1998). *ISO 9241-11 : Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) : Part 11 : Guidance on Usability*. ISO.

Stein, J., Nunes, I., and Cirilo, E. (2014). Preference-based feature model configuration with multiple stakeholders. In *Proceedings of the 18th International Software Product Line Conference*, pages 132–141.

Tamine-Lechani, L. and Bahsoun, W. (2006). Définition d'un profil multidimensionnel de l'utilisateur : Vers une technique basée sur l'interaction entre dimensions. In *COnférence en Recherche d'Infomations et Applications - CORIA 2006, 3rd French Information Retrieval Conference, Lyon, France, March 15-17, 2006. Proceedings*, pages 225–236.

Triki, R., Mazo, R., and Salinesi, C. (2014). Combining configuration and recommendation to enable an interactive guidance of product line configuration. pages 141–160.

Wang, H., Li, Y. F., Sun, J., Zhang, H., and Pan, J. (2005). A semantic web approach to feature modeling and verification. In *In Workshop on Semantic Web Enabled Software Engineering (SWESE'05*.

Weiss, D. M. and Lai, C. T. R. (1999). *Software Product-Line Engineering : A Family-Based Software Development Process*. Addison-Wesley Longman Publishing Co., Inc., USA.

White, J., Dougherty, B., Schmidt, D. C., and Benavides, D. (2009). Automated reasoning for multi-step feature model configuration problems. SPLC '09, page 11–20, USA. Carnegie Mellon University.

Wohlin, C., Runeson, P., Hst, M., Ohlsson, M. C., Regnell, B., and Wessln, A. (2012). *Experimentation in Software Engineering*. Springer Publishing Company, Incorporated.

Xiong, Y., Hubaux, A., She, S., and Czarnecki, K. (2012). Generating range fixes for software configuration. In *Proceedings of the 34th International Conference on Software Engineering (ICSE'12)*, pages 58–68, Zurich, Switzerland. IEEE Computer Society.

Zhang, H. and Babar, M. (2009). Adopting systematic reviews in software engineering : An evidence-based report. Technical report, Lero Software Engineering Research Centre.

Ziadi, T. (2011). *Manipulation de Lignes de Produits en UML*. PhD thesis, Université de Rennes 1,.

Ziadi, T., Jezequel, J.-M., and Fondement, F. (2003). Product line derivation with uml.

.1 Liste des études sélectionnées 2005-2019

S1. K. Czarnecki, S. Helsen, U.W. Eisenecker, Staged configuration through specialization and multi-level configuration of feature models, in : Software Process : Improvement and Practice, 2005.

S2. M. Mendonca, K. Czarnecki, T. Oliveira, D. Cowan, Towards a framework for collaborative and coordinated product configuration, in Proceeding of the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications, 2006, pp. 774-775.

S3. M. Mendonça, D. Cowan, T. Oliveira, A Process-Centric Approach for Coordinating Product Configuration Decisions, in Proceedings of the 40th Hawaii International Conference on System Sciences (HICSS'07), 2007, pp. 1-10.

S4. M. Mendonca, T. Bartolomei, D. Cowan, Decision-making coordination in collaborative product configuration, in Proceedings of the 2008 ACM symposium on applied computing, 2008, pp. 108-113.

S5. M.Mendonca, D.Cowan, W.Malyk, T.Oliveira, Collaborative product configuration : formalization and efficient algorithms for dependency analysis, Journal of Software, 2008, pp. 69-82.

S6. A.Classen, A. Hubaux, P. Heymans, Analysis of Feature Configuration Workflows, in : Proceedings of the 17th IEEE International Requirements Engineering Conference, 2009.

S7. S.Chen, L. Liu, M. Tseng, Product configuration via negotiation for mass customization : An interactive goal programming approach. in : Proceedings of the 6th International Conference on Industrial Engineering and Engineering Management, 2009.

S8. M. Mendonca, D. Cowan, Decision-making coordination and efficient reasoning techniques for feature-based configuration, Science of Computer Programming, 75, (2010), pp. 311-332 .

S9. E. Bagheri, T.D. Noia, A. Ragone, D. Gasevic, Configuring software product line feature models based on stakeholders' soft and hard requirements, in : Proceedings of the 14th International Software Product Line Conference. Springer, Berlin. 2010.

S10. E. Bagheri, M. Asadi, D. Gasevic, S. Soltani, Stratified analytic hierarchy process : Prioritization and selection of software features, in : Proceedings of the 14th International Conference on Software Product Lines, 2010, pp. 300-315,.

S11. A. Hubaux, P. Heymans, P.Y. Schobbens, D. Deridder, Towards Multi-view Feature-Based Configuration. in : Proceedings of the 16th International Working Conference on Requirements Engineering :Foundation for Software Quality (REFSQ'10), Springer-Verlag, Essen, Germany, 2010, pp. 106-112.

S12. R. Rabiser, P. Grunbacher, G. Holl, Improving Awareness during Product Derivation in Multi-User Multi Product Line Environments, in : Proceedings of the 1st International Workshop on Automated Configuration and Tailoring of Applications (ACoTA), 2010.

S13.Y. Bingliang, L.Renwang, and W. Xianmei. Study on product collaborative configuration design platform for mass customization', in : Proceedings of the 2nd IEEE International Conference on Information Management and Engineering (ICIME), 16–18 April, Chengdu, China, pp.489-493. 2010.

S14. E.K.Abbasi, A.Hubaux, P. Heymans, A Toolset for Feature-Based Configuration Workflows, in : Proceedings of the 15th International Software Product Line Conference, 2011.

S15. S. Soltani, M. Asadi, H. Marek, D. Gasevic, E. Bagheri, Automated Planning for Feature Model Configuration based on Stakeholder's Business Concerns, in : Proceedings of the 26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011), KS, USA, 2011, pp. 536-539.

S16. D. Dhungana, D. Seichter, G. Botterweck, R. Rabiser, P. Gruenbacher, D. Benavides, J. A. Galindo, Configuration of multi product lines by bridging heterogeneous variability modeling approaches, in : Proceedings of the 15th International Software Product Line Conference (SPLC'11), IEEE, 2011, pp. 120-129.

S17. C.M. Junior, E. Cirilo, C. Lucena, Assisted user-guidance in collaborative and dynamic software product line configuration, in : Proceedings of the IberoAmerican Conference on Software Engineering (CIbSE), 2011, pp.143-156.

S18. S. Soltani, M. Asadi, D. Gasevic, M. Hatala, E. Bagheri, Automated planning for feature model configuration based on functional and non-functional requirements, in : Proceedings of the 16th International Software Product Line Conference, 2012, pp. 56-65.

S19. G.Holl, D.Thaller, P.Grunbacher, C.Elsner, Managing emerging configuration dependencies in multi product lines. In :Proceedings of the 6th International Workshop on Variability Modelling of Software-Intensive Systems, Germany, 2012, pp. 3-10.

S20. C.Elsner, Light-weight tool support for staged product derivation, in :Proceedings of the 16th International Software Product Line Conference, 2012, pp. 146-155.

S21. G. Holl, P. Grunbacher, C. Elsner, T. Klambauer, Supporting awareness during collaborative and distributed configuration of multi product lines, in :Proceedings of the 19th Asia Pacific Software Engineering Conference (APSEC), Washington, DC, USA, 2012, pp. 137-147.

S22. G. Holl, P. Grunbacher, C. Elsner, T. Klambauer, M. Vierhauser ,Constraint Checking in Distributed Product Configuration of Multi Product Lines, in : Proceedings of the 20th Asia Pacific Software Engineering Conference (APSEC). 2013.

S23. H. Zhang, R. Lin, H. Zou, F. Yang, Y. Zhao, The Collaborative Configuration of Service-Oriented Product Lines Based on Evolutionary Approach. in : Proceedings of the IEEE International Conference on Services Computing. 2013.

S24. H. Zhang, R. Lin B. Huang, X.Qi, Fuzzy multi-objective collaborative evaluation for business process family configuration. The Journal of China Universities of Posts and Telecommunications, 2013, pp. 11-16.

S25. A. Hubaux, P. Heymans, P.Y. Schobbens, D. Deridder, E. Abbasi, Supporting multiple perspectives in feature-based configuration. Software and Systems Modeling, 2013, pp.1–23.

S26. T.Klambauer, G Holl, P Grunbacher, Monitoring system-of-systems requirements in multi product lines. Requirements Engineering, Foundation for Software Quality , 2011, pp 379-385.

S27. J. Stein, I. Nunes, E. Cirilo, Preference-based feature model configuration with multiple stakeholders , in : Proceedings of the 18th International Software Product Line Conference, 2014, pp. 132-141.

S28. E. Bagheri, F. Ensan, Dynamic Decision Models for Staged Software Product Line Configuration. Requirements Engineering, 19(2), 2014, pp. 187-212.

S29. M. Noorian, E.Bagheri, W. Du, From intentions to decisions : Understanding stakeholders' objectives in Software Product Line configuration. in : Proceedings of the 26th International Conference on Software Engineering and Knowledge Engineering, Vancouver, BC, Canada, July, 2013, pp. 671-677.

S30. C. Henard, M.Papadakis, M.Harman , Y.Le Traon, Combining Multi-Objective Search and Constraint Solving for Configuring Large Software Product Lines, in : Proceedings of the 37th IEEE International Conference on Software Engineering. 2015.

S31. J. A. Galindo, D. Dhungana, R. Rabiser, D. Benavides, G. Botterweck, P. Grunbacher, Supporting Distributed Product Configuration by Integrating Heterogeneous Variability Modeling Approaches. Information and Software Technology, 2015, pp. 78-100.

S32. L.Ochoa, O.Gonzalez-Rojas, and T. Thum, Using decision rules for solving conflicts in extended feature models, in : Proceedings of the 2015 ACM SIGPLAN International Conference on Software Language Engineering, 2015, pp.149-160.

S33. L. Hotz, Y. Wang, M. Riebisch, O. Gotz, J. Lackhove, Evaluation across multiple views for variable automation systems, in : Proceedings of the 19th International Conference on Software Product Line, 2015, pp. 311-315.

S34. R.Dou, Y. Zhang, G.Nan, Customer-oriented product collaborative customization based on design iteration for tablet personal computer configuration. Computers & Industrial Engineering, 99, 2016, pp. 474-486.

S35. B. Zhang, M. Becker, Supporting product configuration in application engineering using EXConfig., in : Proceedings of the 20th International Systems and Software Product Line Conference, 2016, pp. 324-327.

S36. M.Noorian, E.Bagheri, W.Du, Quality-centric feature model configuration using goal models, in : Proceedings of the 31st Annual ACM Symposium on Applied Computing, April , 2016, pp.04-08.

S37. L.Ochoa, O.Gonzalez-Rojas, Program Synthesis for Configuring Collaborative Solutions in Feature Models, On the Move to Meaningful Internet Systems : OTM 2016 Workshops, 2016, pp. 98-108.

S38. J. Alves Pereira, Runtime Collaborative-Based Configuration of Software Product Lines, in : Proceedings of the 39th International Conference on Software Engineering Companion (ICSE-C), 2017.

S39. J.Alves Pereira, A collaborative-based recommender system for configuration of extended product lines, in : Proceedings of the 39th International Conference on Software Engineering Companion (ICSE-C), 2017.

S40. E.Lucas, T. Oliveira, K.Farias, CollabRDL : A language to coordinate collaborative reuse. Journal of Systems and Software. 131, 2017, pp. 505-527.

S41. J.Alves Pereira, P.Matuszyk, S.Krieter, M.Spiliopoulou, G.Saake, Personalized Recommender Systems for Product-line Configuration Processes. Journal of Computer Languages, Systems and Structures.2018, pp.451-471.

Pre-questionnaire
Before starting the experiment, please take 15 minutes to carefully answer this short questionnaire to determine your profile and knowledge so that the final results can be effectively interpreted and analyzed. We would like to inform you that your answers will be used for statistical analysis and their anonymity will be respected

What is your level of experience in the following field: Software Product Lines Engineering (SPLE).

☐ Less than one year

☐ Between 1 et 5 years

☐ Between 5 et 10 years

Configuration of SPLE?

☐ It doesn't ring a bell ☹

☐ I've heard about ☺

☐ Yes, I know what it is about ☺

Collaborative configuration of SPLE is:

☐ No idea

☐ Many people configuring a single product with each other

☐ Many people that each configure a product

☐ One person configuring multiple products

☐ One person configuring a single product

In collaborative configuration of SPLE, a conflict is:

☐ No idea

☐ Two or more choices that cannot be wrong at the same time in a configuration

☐ Two or more choices that cannot be true at the same time in a configuration

☐ Two or more choices that may or not be true at the same time in a configuration

What do you think conflict resolution in the collaborative configuration is all about?

.....
.....

The Colla-Config thesis team would like to thank you for your participation. ☺

Installation and set-up

1. Install nodeJS from: https://nodejs.org/en/download/
2. Install Visual Studio Code from: https://code.visualstudio.com/download
3. After opening Visual Studio Code editor, type the following command line on its terminal: ***npm install -g @vue/cli***

4. Check the vue version by typing ***vue –version***

   If vue version more than 3. Then
5. On the terminal type ***vue create PROJECT_NAME***

6. If you have any problem (e.g version mismatch) type ***npm install -g vue@latest*** (You will receive a request to choose which default template you want to use. (Babel / Eslint or Manuel) choose Babel / Eslint. This will make it easier to find errors in your code)

   Else If vue version less than 3. Then

5.On the terminal type ***npm install -g @vue/cli-init***

6.On the terminal type ***vue init webpack PROJECT_NAME***

7. Download and unzip the project available by clicking on the following link: https://github.com/sabrinacri/Colla-Config-tool.git.

8. Go to Visual Studio Code and import the unzipped project as follows: File ->Open-> select the project you just downloaded.

9. Always on the terminal type ***cd PROJECT_NAME*** (be sure of the path)

10. Once you are under the project type ***npm install***
11. Type ***npm run (or npm start)***

Post- questionnaire

Part 1 Experimental environment (ease of learning)

- **The use of the tool:**

  Very Easy
  Easy
  Complicated
  Very complicated
  Strongly complicated

- **The configuration principle using the tool:**

  Very Clear
  Clear
  Slightly sophisticated
  Sophisticated
  Very Sophisticated

Part 2 Overall Satisfaction (Subjective satisfaction)

- **How did you find the experimentation process?**

  Simple and easy to follow
  Barely understandable
  Complicated
  Very complicated
  Strongly complicated

- **On a scale of 0 to 5, how strongly you are satisfied with the final product?**

  0
  1
  2
  3
  4
  5

- **On a scale of 0 to 5, how strongly you are convinced of the principle of conflict resolution?**

  0
  1
  2
  3

4
5

Part 3 Tool Performance (ease of use)

- **On a scale of 0 to 5, how would you rate the speed of execution, the speed of the tool (the performance of the tool)?**
  0
  1
  2
  3
  4
  5
- **Do you have specific examples of problems with response times and access to the tool?**
  .....
  .....

- **On a scale of 0 to 5, how would you rate the ergonomics / ease of use of the tool?**
  0
  1
  2
  3
  4
  5
- **Do you have specific examples of ergonomics / usability issues?**
  .....
  .....

Part 4 General questions about the tool

- **On a scale of 0 to 5, how strongly would you recommend using the tool?**
  0
  1
  2
  3
  4
  5

- **On a scale of 0 to 5, how strongly do you remember the configuration processes principle (part 1 and 2)? (ease of remembering)**

  0
  1
  2
  3
  4
  5

- **Which configuration process did you like the most? (ease of remembering)**

  Free order (experiment part 1)
  With order, imposed by the product manager (experiment part 2)
  Indifferent

- **On a scale of 0 to 5, how strongly do you remember the conflict resolution processes principle (part 1 and 2)? (ease of remembering)**

  0
  1
  2
  3
  4
  5

- **Which method of conflict resolution did you prefer? (ease of remembering)**

  Automatic with consideration of your preferences (experiment part 1)
  systematic according to the order of configuration (experiment part 2)
  Both
  None of the above
  Indifferent

- **Could you briefly explain what did you retain from the Colla-Config configuration principle and conflict resolution process in the experiment part 1? (ease of remembering)**

  .....
  .....