

# A Feature Model Metrics-Based Approach to Develop a Software Product Line

Yacine Djebar<sup>1</sup>, Mohamed Kimour<sup>2</sup>, and Nouredine Guersi<sup>3</sup>

<sup>1</sup>Department of Computer Science, University of 08 Mai 1945, Algeria

<sup>2</sup>Department of Computer Science, University of Badji Mokhtar, Algeria

<sup>3</sup>Department of Electronics, University of Badji Mokhtar, Algeria

**Abstract:** *In recent years, the Software Product Line (SPL) is becoming a mainstream strategy in software development. The high reusability and the great derivability by modelling common and variable artefacts are undoubtedly its significant strengths. Taking advantage of these strengths requires a design of efficient product line. Often, most existing SPL design approaches build on feature modelling by analysis of existing similar products. However, existing feature-based modelling techniques lack analysis support for building SPL with regard to different stakeholder's views. In this paper, we propose an approach based on analyzing and assessing process for creating expressive structure of an SPL. Such a process provides stakeholders with a set of optimal structures of SPL in different models and a set of metrics. In doing so, we argue that we facilitate the selection of appropriate predefined products structures under the form of a set of configuration-views.*

**Keywords:** *SPL, feature diagram, multi-criteria analysis, multi-view configuration.*

*Received June 8, 2014; accepted January 27, 2015*

## 1. Introduction

Software Product Line (SPL) concept engineering aims to rationalize the development of families of similar software products. The principal outcome of such engineering paradigm is to promote reuse throughout the development process to reduce time, cost and effort in the development process. In recent years, SPL has been more and more used in industry, as well as in the academic field [12, 20]. SPL offers a unified representation of a conceptually similar set of software systems that share common features and meet the requirements of particular domain [5, 6, 14, 19, 22]. The SPL engineering includes three activities: identification of the variability, derivation of products and management of constraints [2, 7, 8, 23]. According to the paradigm of SPL, developing new software products follows either top-down or bottom-up methodologies by deriving a product from the SPL models or refactoring an existing software architecture towards an SPL [9, 20]. Though the second type of the above-mentioned methodologies is more and more used to build SPL in industry, it lacks guidelines on how to select and evaluate suitable parts of existing legacy software in a practical and economic way [3, 4].

In this paper, we propose an approach based on assessing process to develop a SPL. In such a process, we consider as inputs a predefined set of existing products which form a primary structure an SPL in the form of Feature Diagram (FD) using a variability model. Then, we capture the configurability of predefined products in the form of views.

In order to, evaluate the configurability, an Analysis Model (AM) in accordance with different granularity analysis criteria is used. Results are a set of potential products in form of analysis views. To capture reusability, commonality and differentiability of analysed potential products, the analysis views are transformed by a mapping model into an appropriate form we call correlation views. In order to, evaluate the configurability, an AM in accordance with different granularity analysis criteria is used. Results are a set of potential products in form of analysis views.

In doing so, the measurement of the similarity (reusability and commonality) and the differentiability of potential products according to correlation views can be performed. From such concerns, we establish some metrics to determine optimal (or even desired) views. Views are used to determine optimal FD for each criterion and the global one for the SPL.

The rest of paper is organized as follows. We give an overview of the development process in section 2. Section 3 presents the AM. Section 4 describes a Correlation Model (CM). Section 5 illustrates a mapping model and how analysis views are transformed into correlation views. Section 6 explains the identification and the configuration and the section 7 describes assessment activities. Section 8 gives an overview of a support tool of our approach. Section 9 discusses the proposed approach in the context of related works and in section 10 we overview our future works and conclude the paper.

## 2. An Overview of a Development Process

In addition of a variability model, the key steps and artefacts of the SPL development process employ three other models: an AM, a mapping model and a CM as shown in Figure 1. Typically, the process starts with variability modeling which produces an Initial Feature Diagram (IFD) from a stakeholder’s predefined set of existing similar products.

The second step focuses on the analysis from IFD of the variability contained in internal structure of predefined products according to different analysis criteria. Results are a set of initial configurations of products that are represented as appropriate views in form of tables according to criterion: the Initial Criterion Analysis View (ICAV).

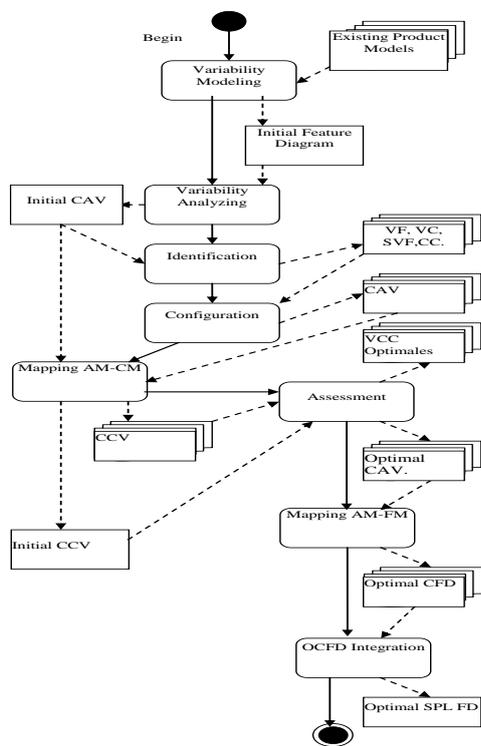


Figure 1. Key steps of our development process.

In the third step, a mapping model allows to transform Initial Criterion Correlation View (ICCV) into a matching representation in a CM: ‘the ICCV’.

Then, the process focuses on the identification of a Variation Framework (VF) and its Configuration Constraints (CC), which allow dividing VF into Sub-Sets (SVF). Identification determines the Variation Configurations (VC) required for each SVF according to CC. VC gives a potential view on the evolution of each initial configuration of product according to CC in the SVF. We’ve noted this form of variation of a product (i):  $(\Delta SP_i)$

By applying a set of VC to the ICAV, we’ll obtain a new analysis view CAV.

This analysis view takes the same representation of ICAV with a consideration of VC changes. This task is performed by the configuration.

The AM-CM mapping model allows transforming every CAV into a CCV. The next step allows assessing with a set of appropriate metrics on the one hand the correlation between internal parts of outputted products and on the other hand between product configurations in terms of similarity, commonality reusability and differentiability.

Then, the measurement of the similarity (reusability and commonality) and the differentiability of products according to correlation views can be performed. The metrics related to reusability will be used to determine Optimal (or even desired) Criterion Correlation Views (OCCV) and Optimal Criterion Analysis View (OCAV).

By applying another mapping between AM and Feature Model (AM-FM), a matching analysis view according to criterion allows establishing Optimal Criterion Feature Diagram (OCFD).

Finally, an SPL optimal feature diagram can be obtained by integrating all of OCFD. To achieve this, we can use FDs composition tools. In our work we’ve opted for the tool used by the approach of [1].

We’ll describe in the following the key steps of our SPL development process.

In order to, present the basic performance of our approach, we focused our illustrations in the rest of paper on using examples from the smart home SPL [15] and specifically the “Lock Control Framework part (SHLC)” that we’ve adapted a FD to our study.

This system allows to control the access to a smart home. In order to, describe variability in SHLC, we consider that it can be designed to be used with different authentication devices i.e., ‘key reader’ and ‘fingerprint reader’ and provides two lock alternatives of the entry door: a ‘manual’ and ‘electronic’. For the latter, it is possible to activate an ‘actuator’ in order to unlock the door after authentication. In addition, a ‘display’ option of a lock system status (by means of screen) is available. This option is provided by the system to remotely communicate with the home owner by one or two modes of communication: ‘WiFi’ and ‘Internet’. The latter alternative requires the activation of a display function. The FD of SHLC, using the classical notation from [7], is illustrated in Figure 2. In the next section; we’ll describe the AM.

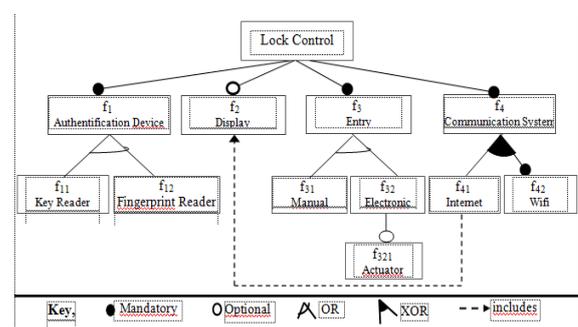


Figure 2. The initial feature diagram of (shlc) spl.

### 3. Analysis Model

This model is used to analyze a SPL FD according to different building way of the internal structure of existing products. To achieve this, we've used a large field of analysis criteria.

The Model represents a configuration of product in an appropriate form required to calculate the structural metrics related to analysis criterion of a considered set of software products even in heterogeneous project contexts.

The analysis is performed differently according to fine-grained and coarse-grained sizes of Analysis Unit (AU) that will be defined in the following section.

#### 3.1. Definition of Analysis Unit

In AM, an internal structure of a product is modelled by a Boolean array in which each cell corresponds to a building unit according to analysis criterion.

All of the analysis units are represented in this array. The value '1' means that AU is present in the configuration of product according to analysis criterion and '0' the other case.

This representation is illustrated in Table 1 for SHLC SPL for the feature as AU.

Table 1. Example of an internal structure of a product.

f <sub>1</sub>	f <sub>2</sub>	f <sub>3</sub>	f <sub>4</sub>	f <sub>11</sub>	f <sub>12</sub>	f <sub>31</sub>	f <sub>32</sub>	f <sub>41</sub>	f <sub>42</sub>	f <sub>321</sub>	f <sub>1</sub>
1	1	1	1	1	0	0	1	1	0	1	1

#### 3.2. Analysis Process

The analysis is performed according to different AU granularity. The feature is used by the fine-grained analysis as AU. The coarse-grained analysis is based on FD hierarchical level and a specific unit that we've called a 'scanning unit' according to analysis criterion.

Each type of analysis is based on a set of criteria. The criteria used in fine-grained analysis are 'Usefulness' and 'Depth'. In the coarse-grained analysis, we employ the Hierarchical Level Coverage (HLC) and the 'Scanning' criteria. Figure 3 shows a structure of AM.

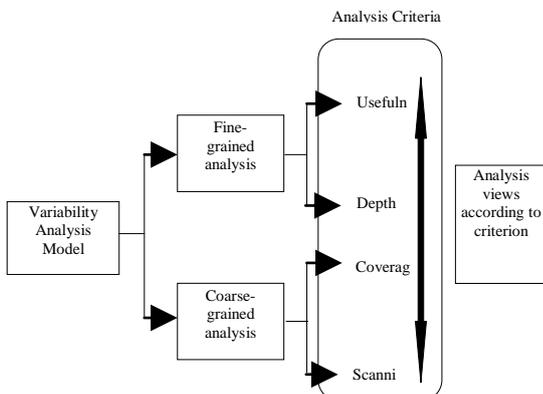


Figure 3. An overview of AM.

##### 1. Fine-Grained Analysis:

- **Usefulness Criterion:** Usefulness criterion allows measuring the degree of presence of features in a product configuration. The resulted configurations are grouped in an appropriate table that represents the Usefulness View (UV) of products. Table 2 shows a usefulness view of a (SHLC) SPL.

Table 2. Uv of SHLC SPL.

UV	f <sub>1</sub>	f <sub>2</sub>	f <sub>3</sub>	f <sub>4</sub>	f <sub>11</sub>	f <sub>12</sub>	f <sub>31</sub>	f <sub>32</sub>	f <sub>41</sub>	f <sub>42</sub>	f <sub>321</sub>	UV
SP <sub>1</sub>	1	1	1	1	1	0	0	1	1	0	1	SP <sub>1</sub>
SP <sub>2</sub>	1	0	1	1	1	0	1	0	0	1	0	SP <sub>2</sub>
SP <sub>3</sub>	1	1	1	1	0	1	0	1	1	1	1	SP <sub>3</sub>

- **Metric Related to Usefulness:** Before defining metric related to usefulness, we'll define two required parameters:

- a) *NF* as a number of *FD* features and *NP* as a cardinal of a set of products under study (*S<sub>p</sub>*).
- b) **The Usefulness Ratio:** The usefulness ratio of a product 'j' in *UV* is noted *UR<sub>j</sub>* or *UR(j)*. It measures the degree of presence of the product components relative to all of *FD* features. It is calculated according to the following equation:

$$UR(P_j) = (\sum_{j=1}^{NF} UV(i, j)) / NF \quad (1)$$

In (SHLC) SPL, We'll have:

$$UR(P_1)=0.72. \text{ Expressed in } \%, UR(P_1)=72\%.$$

- **Depth Criterion:** The depth of a product allows measuring the configuration of product in terms of leaves in a *FD* graph. In the *FD* graph, the leaf is a feature without successors.

Reach more leaves means reach more depth in *FD* graph. To analyze the depth of a product, we need:

1. To consider a structure of (FD) as a graph.
2. To identify the leaves of (FD).

For this, We use the exterior half-degree of features (*dG<sup>+</sup>(fi)*). Leaves correspond to features with (*dG<sup>+</sup>(fi)*)=0. Table 3 illustrates this identification for (SHLC) SPL.

3. To represent by a Boolean array the structure of every product in terms of leaves.
4. To group all of the configurations resulting from the previous steps in an appropriate table that we call a Depth View of products (DV). This view in form of table represents a predefined set of existing products in terms of leaves. Table 4 illustrates an example of this table for (SHLC) SPL.
5. To determine the number of *FD* graph leaves *NL*.
6. To calculate a depth metric that we've called Depth Ratio (DR) of a product with a following equation:

$$DR(P_i) = (\sum_{j=1}^{NF} DV(i, j)) / NL \quad (2)$$

Example of calculating of DR in SHLC SPL:  
 $DR(P_1) = 0.57$  and in % ,  $DR(P_1) = 57\%$ .

2. *The Coarse-Grained Analysis:* For the coarse-grained analysis, we use the HLC and the Scanning (SC) criteria.

- *HLC:* HLC of a product allows measuring the usefulness of a product according to the membership of its building components (features) to the FD hierarchical levels.

Table 3. Identification of leaves in (SHLC) feature diagram.

Feature	f <sub>1</sub>	f <sub>2</sub>	f <sub>3</sub>	f <sub>4</sub>	f <sub>11</sub>	f <sub>12</sub>	f <sub>31</sub>	f <sub>32</sub>	f <sub>41</sub>	f <sub>42</sub>	f <sub>321</sub>
dG <sup>+</sup> (ff)	2	0	2	2	0	0	0	1	1	0	0

Table 4. DV of (SHLC) SPL.

DV	f <sub>1</sub>	f <sub>2</sub>	f <sub>3</sub>	f <sub>4</sub>	f <sub>11</sub>	f <sub>12</sub>	f <sub>31</sub>	f <sub>32</sub>	f <sub>41</sub>	f <sub>42</sub>	f <sub>321</sub>
SP <sub>1</sub>	0	1	0	0	1	0	0	1	0	0	1
SP <sub>2</sub>	0	0	0	0	1	0	1	0	0	1	0
SP <sub>3</sub>	0	1	0	0	0	1	0	0	0	1	1

This criterion estimates the degree of horizontal coverage of product configuration in a FD in terms of feature.

To achieve this, we need to organize FD into Hierarchical Levels (HL). The number of HL is noted NHL. HL<sub>#1</sub> is the highest level and HL<sub>#NHL</sub> is the lowest one.

The considered products are represented in an appropriate view in form of table that we've called 'HL Coverage View' (CV).

For example, in SHLC FD, the number of HL is 3. Thus, we'll have HL<sub>#1</sub>, HL<sub>#2</sub> and HL<sub>#3</sub>. Each HL contains a set of features.

In (SHLC): HL<sub>#1</sub> contains 4 features: f<sub>1</sub>, f<sub>2</sub>, f<sub>3</sub> and f<sub>4</sub>. For determining the product HLC, we need to know:

- The number of features of each HL (HLFN).
- The number of features composing a configuration of every product according to their membership to FD Hierarchical level PHLFN.

Resulted configurations of products are grouped in an appropriate table that represents a HL CV of products under analysis.

In the following, we'll define a formal model for the HL Coverage.

The sets that define (SHLC) SPL according to HL coverage are illustrated as follows:

$$\begin{aligned}
 SP_i &\subset Sf \quad \forall i \in N \text{ and } i \leq NP \\
 SHL_{\#j} &\subset Sf \quad \forall i \in N \text{ and } i \leq NP \\
 (SHL_{\#j}) &\subset Sf, \forall j \in N \text{ and } j \leq NHL \\
 \text{if } f_k &\in (SP_i) \Rightarrow f_k \in (SHL_{\#j}) \forall f_k \in (S_f) \Rightarrow f_k \in (SHL_{\#j}) \\
 S_f &= \{f_1, f_2, f_3, f_4, f_{11}, f_{12}, f_{31}, f_{32}, f_{41}, f_{42}, f_{321}\} \\
 SP_1 &= \{f_1, f_2, f_3, f_4, f_{11}, f_{32}, f_{41}, f_{321}\} \\
 SHL_{\#1} &= \{f_1, f_2, f_3, f_4\}; SHL_{\#3} = \{f_{321}\} \\
 SPHL_{\#1,2} &= \{f_{11}, f_{32}, f_{41}\} \\
 \text{if } f_k &\in (SHL_{\#j}) \Rightarrow \\
 f_k &\notin (SHL_{\#m}) \forall (m \in N) \text{ and } (m \leq NHL)
 \end{aligned}$$

Now, we'll define a formal model for the HL CV.

At the present, we'll define a metric related to HL  $\forall (HL_{\#j})$  a hierarchical level and  $(j \in N)$  and  $(j \leq NHL)$   
 $\Rightarrow (HL_{\#j}) \in (SHL)$  and  $(card(HL_{\#j}) \leq card(S_f))$   
 Coverage Ratio (CR).

$$CV(m, n) = \begin{cases} 1, & \text{if .at .least } \exists f_k \in (SP_m) \text{ and } f_k \in (SHL \neq n) \\ 0, & \text{else} \end{cases} \quad (3)$$

- The CR is calculated from the CV table by a following equation:

$$CR(P_i) = (\sum_{j=1}^{NHL} CV(i, j)) / NP \quad (4)$$

A Product P<sub>1</sub> covers the three HL, it has a HL CR=3/3=1, and 100% if expressed in percentage.

The product P<sub>2</sub> covers two HL and has a HL CR=0.66.

The HL CV of SHLC SPL is shown in Table 5.

Table 5. HL Coverage View (CV) of (SHLC) SPL.

CV	HL <sub>#1</sub>	HL <sub>#2</sub>	HL <sub>#3</sub>
SP <sub>1</sub>	1	1	1
SP <sub>2</sub>	1	1	0
SP <sub>3</sub>	1	1	1

- *The Scanning Criterion:* The scanning criterion allows measuring the internal structure of product according to a particular building unit that we called 'scanning unit' SU.
- *Definition of SU:* The scanning Unit is a basic coarse-grained unit used for analyzing internal structure of product. In a difference of any FD coarse-grained unit, the SU scans the surface of FD graph from the highest HL to the lowest HL with memorizing the arcs find on its path until detecting a first leaf.

The SU can be defined as a simple and elementary path composed of a set of oriented arcs in the FD graph.

The identification process of SU explores all of alternatives in terms of oriented arcs offered across a FD graph.

- The identification method of SU. To identify SU, we need:
  1. To consider a FD as a graph without a root.
  2. To know the leaves of FD defined in i.
  3. To split the FD graph structure into correlated components C<sub>i</sub>.
  4. To establish the relationship between oriented arcs U<sub>j</sub> in C<sub>i</sub>.
  5. To identify SUs of each C<sub>i</sub> according to relationship configuration rules defined in Figure 4.

The relationship between a pair of U<sub>j</sub> may be «XOR», «OR», «And Optional», «And Mandatory». «includes with and optional», «includes with 'and mandatory' and «excludes with and optional».

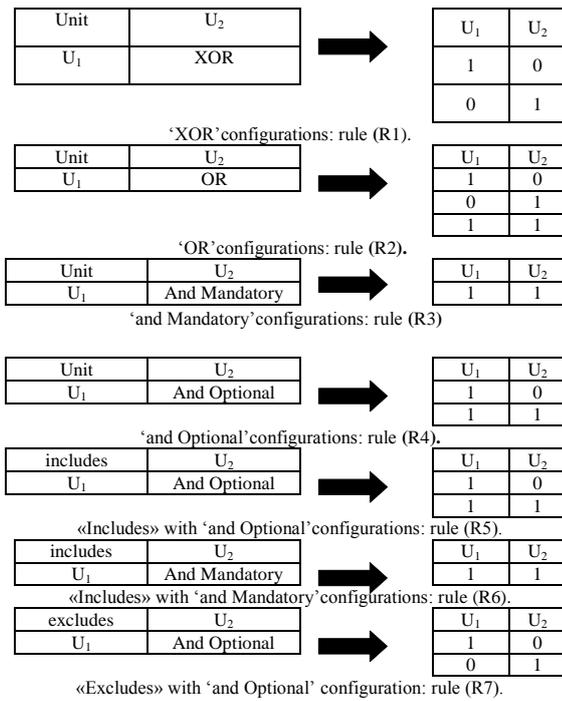


Figure 4. Relationship Configuration (RC) rules.

‘excludes with and Mandatory’ is not a valid configuration.

We’ll apply the SU identification method for SHLC feature diagram.

1. The root of (SHLC) FD is ‘ $f_0$ ’: To know the leaves of the FD graph, we need to determine the exterior half-degrees of the FD graph features ( $dG^+(f_i)$ ).

Results are shown in Table 3. The features:  $f_2, f_{11}, f_{12}, f_{31}, f_{42}, f_{321}$  have ‘0’ as exterior half-degree. They represent the leaves.

2. After splitting the SHLC FD graph into correlated components ( $C_i$ ), We obtained the results illustrated in Figure 5. The produced Sets of ( $C_k$ ) are as follows:

$$C_1 = \{f_1, f_{11}, f_{12}\}, C_2 = \{f_2\}, C_3 = \{f_3, f_{31}, f_{32}, f_{321}\} \text{ and } C_4 = \{f_4, f_{42}, f_{41}, f_2\}.$$

3. Definition of Arcs: In the interest of brevity and clarity, we will only detail the definition of arcs in ( $C_3$ ).

( $U_j$ ) represent arcs in (FD) graph. They are a pair of ‘one feature-one successor’ in ( $C_k$ ).

By applying (RC) rules to a table of configurations. Resulted through ( $U_j$ ) in ( $C_3$ ) are  $U_1, U_2$  and  $U_3$ :

$$U_1 = \{f_3, f_{31}\}, U_2 = \{f_3, f_{32}\}, U_3 = \{f_{32}, f_{321}\}$$

4. The relationship between arcs: Table 6 details the relationship configuration rules (RC) of  $U_1, U_2$  and  $U_3$  considered pair by pair in ( $C_3$ ).
5. Identification of SUs of ( $C_3$ ).

By applying RC rules to a table of configurations resulted after 4, we obtain the configurations shown in red in Table 7.

By removing double configurations, we obtain valid

configurations as illustrated in Table 7 (valid configurations column).

The final results give following valid configurations: ‘101’, ‘011’ and ‘010’.

‘101’ gives a (SU) =  $U_1 = (f_3, f_{31})$

‘011’ gives a second (SU) =  $(U_2, U_3) = (f_3, f_{32}, f_{321})$

‘010’ gives a third (SU) =  $U_2 = (f_3, f_{32})$ .

By applying the same process to the other (FD) correlated components, the final outputted sets of (SU) are as follows:

From ( $C_1$ ):  $SU_1 = \{f_1, f_{11}\}; SU_2 = \{f_1, f_{12}\}$

From ( $C_2$ ):  $SU_3 = \{f_2\}$

From ( $C_3$ ):  $SU_4 = \{f_3, f_{31}\}; SU_5 = \{f_3, f_{32}\}; SU_6 = \{f_3, f_{32}, f_{321}\}$

From ( $C_4$ ):  $SU_7 = \{f_4, f_{41}, f_2\}; SU_8 = \{f_4, f_{42}\}; SU_9 = \{f_4, f_{42}, f_{41}, f_2\}$ .

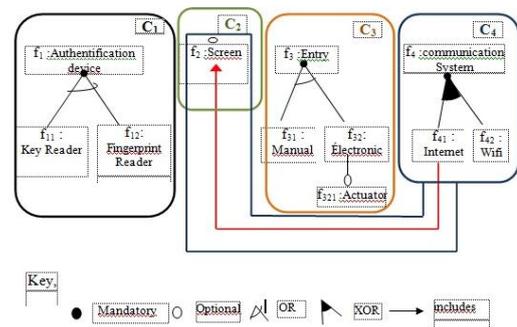


Figure 5. The four correlated components of (SHLC) feature model graph.

Table 6. The relationships between  $U_1, U_2$  and  $U_3$  in ( $C_3$ ).

	$U_1$	$U_2$	$U_3$
$U_1$		XOR	
$U_2$	XOR		
$U_3$		And Opt	

Table 7. The relationship configurations according to RC rules between  $U_1, U_2$  and  $U_3$  in ( $C_3$ ).

Arcs	$U_1$	$U_2$	$U_3$	Configurations	Valid configurations
$U_1$		1		010	
$U_2$	1			100	100
$U_3$			1	011	011

We’ve grouped the internal structure of products in terms of SU in appropriate table we’ve called “Scanning View (SV)”. This view in form of table represents a set of products in terms of SU.

An example of SLHC SV of initial predefined set of products is illustrated in Table 8.

Table 8. SV of (SHLC) SPL.

SV	$SU_1$	$SU_2$	$SU_3$	$SU_4$	$SU_5$	$SU_6$	$SU_7$	$SU_8$	$SU_9$
SP <sub>1</sub>	0	1	0	0	0	1	0	0	0
SP <sub>2</sub>	1	0	0	1	0	0	0	1	0
SP <sub>3</sub>	0	1	1	0	0	1	0	0	1

- Metric Related to Scanning Criterion: For evaluating the structure of product in terms of SU, we use “the Scanning Ratio (SR)”. It represents a degree of SU composition of a product relative to other products.

If we consider NSU as a number of SU, SR can be calculated from a SV table by the following equation:

$$SR(P_i) = \left( \sum_{j=1}^{NSU} SV(i, j) \right) / NSU \quad (5)$$

### 4. Correlation Model

In this part of approach, we've employed a similar model of the relationship model used by [3] that we have adapted to our study in the mapping and evaluation steps. In the next section, we'll describe a structure of the model.

#### 4.1. Internal Structure of Product in a CM

This model perceives an internal structure of product  $p$ : ' $SP_p$ ' as a set of four parts (1, 2, 3 and 4). In the interest of shortness and clarity, we illustrates the structure of a model with a set of three configurations as predefined in a considered set of products:

1. A set of proper elements of a product ' $p$ '. These elements are selected only in a configuration of ' $p$ ' and not in the other product configurations. This set represents the differentiability of ' $p$ '. They are shown in Figure 6 by ( $D_i$ )
2. Sets of shared elements by a pair of configurations of products ( $p_i$  and  $p_j$ ) and not by all of products. These sets represent the similarity between a pair of products. They are shown in Figure 6 by ( $R_{ij}$ ).
3. A set of common elements to all of products. This set represents the commonality. This part is illustrated in Figure 6 by ( $R_{ijk}$ ).
4. Sets of all elements that compose internal structures of products. These sets are used in our approach to link the AM to the CM. They represent a hyphen between these two models. They are illustrated in Figure 6 by  $S_p$ .

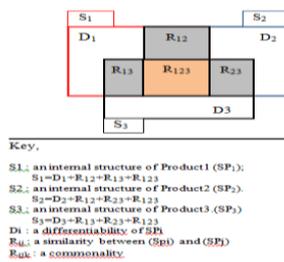


Figure 6. Internal structure of a product in the CM.

#### 4.2. Metrics Related to a CM

Four metrics allows to estimate in terms of AU the size of each part in proportion to other parts:

- $S_i$ : Allows to measure the size of internal structure of a product ' $i$ '.
- $V_i$ : Allows to measure the differentiability in a product ' $i$ '.
- $V_{ij}$ : Allows to measure similarity between two products ' $i$ ' and ' $j$ '.
- $V_{ijk}$ : Allows measuring a commonality.

These metrics are grouped in a particular view of products that we called a CCV according to analysis criterion.

In addition, the model is employed to calculate the configurability of initial predefined set of Products ( $P_i$ ) according to Sub Variable Framework (SVF) and CC of analysis criterion. SVF and CC are defined in section 6.

The configurability result of a predefined Structure of Product ( $SP_i$ ) in SVF is a new structure of product that we noted  $\Delta SP_i$ . The metric related to configurability is a number of potential configurations NC obtained for each SVF.

The CM is used to estimate reusability (similarity and commonality) in software products with consideration to its maximum threshold according to analysis criterion and the configurability with taking account a maximum number of NC.

### 5. A Mapping Model

The mapping model is based on a formalism which allows to transform an analysis view into a correlation view according to analysis criterion.

In the following, we'll define a formal representation of a mapping model.

$\forall i \leq NP$ ,  $SP_i$  is a structure of a product ' $i$ ' represented in initial CAV according to analysis criterion and  $\Delta SP_i$  is a structure of a product ' $i$ ' represented in a CAV of SVF after submitting it to SVF CC.

a definition of ( $S_i$ ) in a CM is as follows:

$$(S_i) = \begin{cases} (UR_i) & \text{if the analysis criterion is Usefulness.} \\ (DR_i) & \text{if the analysis criterion is Depth} \\ (CR_i) & \text{if the analysis criterion is HL Coverage} \\ (SR_i) & \text{if the analysis criterion is the Scanning} \end{cases}$$

- Formal definitions of ( $D_i$ ), ( $R_{ij}$ ), ( $R_{ijk}$ ) and their respective metrics ( $V_i$ ), ( $V_{ij}$ ) and ( $V_{1...n}$ ) are as follows:

$$D_i = \begin{bmatrix} \delta_{i1} \\ \delta_{i2} \\ \vdots \\ \delta_{in} \end{bmatrix} \text{ where } \delta_{ik} = \begin{cases} 1, & \text{if } (i = k) \\ 0, & \text{else} \end{cases} \quad (6)$$

$$\text{if } f_k \in (SP_i) \Rightarrow f_k \notin (SP_j), j \neq i, \quad (7)$$

$$V_i = \text{card}(D_i) / NF$$

$$R_{ij} = \begin{bmatrix} \delta_{ij1} \\ \delta_{ij2} \\ \vdots \\ \delta_{ijn} \end{bmatrix} \text{ where } \delta_{ijk} = \begin{cases} 1, & (k = i) \text{ or } (k = j) \\ 0, & (k \neq i) \text{ or } (k \neq j) \end{cases} \quad (8)$$

$$\text{if } f_k \in (SP_i), f_k \in (SP_j) \text{ and only to } (SP_j), \quad (9)$$

$$V_{ij} = \text{card}(R_{ij}) / NF$$

$$R1, \dots, n = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_n \end{bmatrix} \text{ where } \delta_i = 1, i = 1, \dots, n \quad (10)$$

$$V1, \dots, n = \text{card}(R1, \dots, n) / NF \quad (11)$$

Figure 7 illustrates mechanisms of a mapping model that transforms CAV into CCV for one criterion.

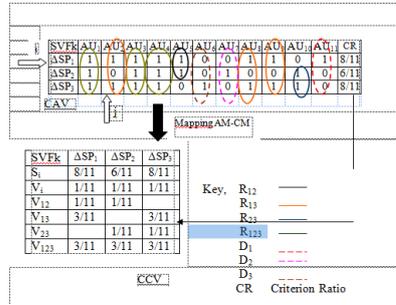


Figure 7. The mapping “analysis model-correlation model”.

## 6. Identification and Configuration Activities

The identification activity allows to identify Variable Framework (VF) from the initial CAV (ICAV).

Identification decomposes VF into sub-sets of configurations (SVF<sub>k</sub>) according to the similarity of dependencies between its elements.

We use these similarities to establish the CC. The identification activity allows also identifying a set of VC from each VF with considering of CC.

The configuration aims to capture configurability of a predefined set of products according to criterion. This task is achieved by varying initial configurations SP<sub>i</sub> in the ICAV according to the criterion VF.

For each VC a new view is established in AM. Thus, ICAV can be configured according to several views. The variable configuration of SP<sub>i</sub> in SVF according to CC is noted ΔSP<sub>i</sub> Figure 7. The number of resulted configurations MC obtained for each SVF in CAV allows to measure a configurability in SVF for each structure of product.

Examples of CC in (SHLC).

- Criterion = “Depth”  
Initial CAV is represented in Table 4.  
VF related to Depth = {f<sub>2</sub>, f<sub>11</sub>, f<sub>12</sub>, f<sub>31</sub>, f<sub>32</sub>, f<sub>42</sub>, f<sub>321</sub>}
- Example of (SVF) without a valid (VF).  
Identification gives a first sub-VF: SVF1 composed of f<sub>11</sub> and f<sub>12</sub>. SVF1 = {f<sub>11</sub>, f<sub>12</sub>}

The (CC) applied to SVF1 in a CAV are as follows:

$$CC1: f_{11}^1 \leftarrow - \rightarrow f_{12} \text{ (exclusive OR).}$$

$$CC2: f_{12} \leftarrow - \rightarrow f_{11}$$

After considering the possible VC of SVF1 in a CAV (framed in Table 9) and after considering the CC defined previously, the variable configurations represented separately by the three frames in Table 9

are not valid. So, the variable configurations VC of (SVF) = {∅} and there is no outputted CAV for this VC.

Table 9. Potential (VC) in SVF1.

DV	f <sub>1</sub>	f <sub>2</sub>	f <sub>3</sub>	f <sub>4</sub>	f <sub>11</sub>	f <sub>12</sub>	f <sub>31</sub>	f <sub>32</sub>	f <sub>41</sub>	f <sub>42</sub>	f <sub>321</sub>
SP <sub>1</sub>	0	1	0	0	1	0	0	1	0	0	1
SP <sub>2</sub>	0	0	0	0	1	0	1	0	0	1	0
SP <sub>3</sub>	0	1	0	0	0	1	0	0	0	1	1

- Example of SVF With valid VF: SVF = {f<sub>2</sub>}  
Identification gives a second sub-VF: SVF2 = {f<sub>2</sub>}  
There is no CC that can be applied to VC of SVF2.

There is only one possible VC for f<sub>2</sub>. It takes “1” for SP<sub>2</sub> shown by a little frame in Table 10 and becomes ΔSP<sub>2</sub>. This VC is valid (no CC for this SVF).

The new depth analysis view CAV produced after considering this VC is represented in Table 10.

Table 10. One possible (VC) in SVF2.

DV	f <sub>1</sub>	f <sub>2</sub>	f <sub>3</sub>	f <sub>4</sub>	f <sub>11</sub>	f <sub>12</sub>	f <sub>31</sub>	f <sub>32</sub>	f <sub>41</sub>	f <sub>42</sub>	f <sub>321</sub>
ΔSP <sub>1</sub>	0	1	0	0	1	0	0	1	0	0	1
ΔSP <sub>2</sub>	0	1	0	0	1	0	1	0	0	1	0
ΔSP <sub>3</sub>	0	1	0	0	0	1	0	0	0	1	1

## 7. Assessing Activity

In order to facilitate understanding different metrics used by the assessing activity and in the interest of shortness and clarity, we will describe them only through the depth criterion for (SHLC) SPL.

The assessing activity is running across two sub-activities: evaluation and optimization.

### 7.1. Evaluation

- Criterion: Depth  
VF = {f<sub>2</sub>, f<sub>11</sub>, f<sub>12</sub>, f<sub>31</sub>, f<sub>32</sub>, f<sub>42</sub>, f<sub>321</sub>}
- CC:  
CC1: f<sub>11</sub><sup>1</sup> ← - → f<sub>12</sub>  
CC2: f<sub>31</sub> ← - → f<sub>32</sub>, f<sub>321</sub> ← - → f<sub>32</sub>  
CC3: f<sub>41</sub><sup>2</sup> - → f<sub>2</sub>

The initial CAV (ICAV) is shown in Table 4.

ICAV includes three configurations.

After applying CC on VF in the Depth ICAV, identification gives 3 different Sub-VFs: SVF1 = {f<sub>2</sub>}; SVF2 = {f<sub>42</sub>} and SVF3 = {f<sub>2</sub>, f<sub>42</sub>}

Table 11. The initial (CCV) of a Depth criterion.

Initial	ΔSP <sub>1</sub>	ΔSP <sub>2</sub>	ΔSP <sub>3</sub>
Si	0.57	0.42	0.57
V <sub>1</sub>	0.14	0.14	0.14
V <sub>12</sub>	0.14	0.14	
V <sub>13</sub>	0.28		0.28
V <sub>23</sub>		0.14	0.14
V <sub>123</sub>	0	0	0
MC	3	3	3
RS	0.42	0.28	0.42
RR	0.42	0.28	0.42

<sup>1</sup> Mutual Exclusion  
<sup>2</sup> Inclusion

The mapping of (ICAV) into a CM gives the Initial CCV shown in Table 11.

(RR) represents the Ratio of Reusability and (RS) a Ratio of Similarity. They will be described in the metrics sub-section.

1. For SVF1: After applying the CC in SVF1, result is one valid VC={f<sub>2</sub>}. By considering VC in SVF1, configuration gives a depth (CAV) and its matching CCV as illustrated respectively in Tables 12 and 13.

Table 12. The depth (CAV) for (VC) = {f<sub>2</sub>} in SVF1.

SFV1	f <sub>1</sub>	f <sub>2</sub>	f <sub>3</sub>	f <sub>4</sub>	f <sub>11</sub>	f <sub>12</sub>	f <sub>31</sub>	f <sub>32</sub>	f <sub>41</sub>	f <sub>42</sub>	f <sub>321</sub>
ASP <sub>1</sub>	0	1	0	0	1	0	0	1	0	0	1
ASP <sub>2</sub>	0	1	0	0	1	0	1	0	0	1	0
ASP <sub>3</sub>	0	1	0	0	0	1	0	0	0	1	1

Table 13. The depth (CCV) for (VC) = {f<sub>2</sub>} in SVF1.

SVF1	ASP <sub>1</sub>	ASP <sub>2</sub>	ASP <sub>3</sub>
Si	0.57	0.57	0.57
V <sub>i</sub>	0.14	0.14	0.14
V <sub>12</sub>	0.14	0.14	
V <sub>13</sub>	0.14		0.14
V <sub>23</sub>		0.14	0.14
V <sub>123</sub>	0.14	0.14	0.14
MC	4	4	4
RS	0.28	0.28	0.28
RR	0.42	0.42	0.42

2. For SVF2: After applying the CC in SVF2, result is one valid (VC)={f<sub>42</sub>}. By considering VC in SVF2, configuration gives the depth CAV and its matching CCV as shown respectively in Tables 14 and 15.

Table 14. The depth (CAV) for (VC) = {f<sub>42</sub>} in SVF2

SFV2	f <sub>1</sub>	f <sub>2</sub>	f <sub>3</sub>	f <sub>4</sub>	f <sub>11</sub>	f <sub>12</sub>	f <sub>31</sub>	f <sub>32</sub>	f <sub>41</sub>	f <sub>42</sub>	f <sub>321</sub>
ASP <sub>1</sub>	0	1	0	0	1	0	0	1	0	1	1
ASP <sub>2</sub>	0	0	0	0	1	0	1	0	0	1	0
ASP <sub>3</sub>	0	1	0	0	0	1	0	0	0	1	1

Table 15. The depth (CCV) for (VC) = {f<sub>42</sub>} in SVF2

SVF2	ASP <sub>1</sub>	ASP <sub>2</sub>	ASP <sub>3</sub>
Si	0.71	0.42	0.57
V <sub>i</sub>	0.14	0.14	0.14
V <sub>12</sub>	0.14	0.14	
V <sub>13</sub>	0.28		0.28
V <sub>23</sub>		0	0
V <sub>123</sub>	0.14	0.14	0.14
MC	4	4	4
RS	0.42	0.14	0.28
RR	0.56	0.28	0.42

3. For SVF3: After applying the CC in SVF3, result is one valid VC={f<sub>42</sub>, f<sub>2</sub>} in SVF3. By considering VC in SVF2, the configuration gives the depth CAV and its matching CCV as illustrated respectively in Tables 16 and 17.

Table 16. The depth (CAV) for (VC) = {f<sub>2</sub>,f<sub>42</sub>} in SVF3.

SFV3	f <sub>1</sub>	f <sub>2</sub>	f <sub>3</sub>	f <sub>4</sub>	f <sub>11</sub>	f <sub>12</sub>	f <sub>31</sub>	f <sub>32</sub>	f <sub>41</sub>	f <sub>42</sub>	f <sub>321</sub>
ASP <sub>1</sub>	0	1	0	0	1	0	0	1	0	1	1
ASP <sub>2</sub>	0	1	0	0	1	0	1	0	0	1	0
ASP <sub>3</sub>	0	1	0	0	0	1	0	0	0	1	1

Table 17. The depth (CCV) for (VC) = {f<sub>2</sub>,f<sub>42</sub>} in SVF3.

SVF3	ASP <sub>1</sub>	ASP <sub>2</sub>	ASP <sub>3</sub>
Si	0.71	0.57	0.57
V <sub>i</sub>	0.14	0.14	0.14
V <sub>12</sub>	0.14	0.14	
V <sub>13</sub>	0.14		0.14
V <sub>23</sub>		0	0
V <sub>123</sub>	0.28	0.28	0.28
MC	5	5	5
RS	0.28	0.14	0.14
RR	0.56	0.42	0.42

- *Assessing Activity Metrics:* In addition of a size of configurability previously defined in a configuration section, the development process allows to generate across assessing activity a certain number of other metrics: the RS between a pair of products, the size of commonality in a set of products, a RR and a ratio of differentiability of a product.
- *The Size of Similarity:* The size of similarity (V<sub>ij</sub> in the CCV of Figure 7) measures a similar internal structure of a pair of products.
- *The Size of Commonality:* The size of commonality is measured for a set of products considered in a CCV. It represents (V<sub>ijk</sub>) in (CCV) of Figure 7.
- *The RS:* The RS of a structure of a product 'j' SP<sub>i</sub> or its varying form ΔSP<sub>j</sub> in SVF is calculated from CCV by a following equation:

$$RS(SP_j) = \sum_{j=1}^{NP} V_{ij} \tag{12}$$

- *The RR:* The RR of a structure of product 'j' SP<sub>j</sub> or its varying form ΔSP<sub>j</sub> in SVF cumulates a RS of SP<sub>j</sub> or ΔSP<sub>j</sub> and a size of commonality. It is calculated from CCV by a following equation:

$$RR(SP_j) = (\sum_{j=1}^{NP} V_{ij}) + (V_{ij...k}) \tag{13}$$

(V<sub>ij...k</sub>) represents a size of commonality and it is previously defined in section 5.

- *The Ratio of Differentiability (RD):* The ratio of differentiability of a structure of product 'j' SP<sub>j</sub> or its varying form ΔSP<sub>j</sub> in SVF measures the distinctiveness in terms of internal structure of this product. It represents the opposite characteristic of commonality. It is represented by (V<sub>i</sub>) in the Table 7 and it is previously defined in section 5.
- MC represents the maximum number of SVF Number of Configurations (NC).
- *Examples:* In initial view ICAV, the number of configurations NC is 3 (Table 10). In SVF2 another configuration from SP2 (Table 10) is added to the initial configurations. The number of configurations becomes 4 in SVF2.

The assessment metrics are grouped in the criterion assessing report .For our case, a depth assessing report is illustrated in Table 18.

Table 18. The depth assessment report.

Depth	ASP <sub>1</sub>	ASP <sub>2</sub>	ASP <sub>3</sub>	MC	Max RS	Max RR
Initial	0.42	0.28	0.42	3	0.42	0.56
SVF1	0.42	0.42	0.42	4		
SVF2	0.56	0.28	0.42	4		
SVF3	0.56	0.42	0.42	5		

7.2. Optimization

Optimization seeks at first to identify the variable sub-frame SVF which holds the maximum threshold of reusability in the assessing report according to the criterion and secondly to determine the optimal SPL feature diagram through the global optimization.

- In the first step, the first output is an optimal criterion sub-frame OSVF. Once identified, the matching CCV of OSVF is considered as an optimal CCV: The OCCV and the corresponding CAV as an optimal CAV: The OCAV.

A mapping of OCAV into feature model gives the optimal view of a feature diagram OVFD according to criterion.

For the depth criterion in SHLC SPL, SVF3 corresponds to OSVF.

The CCV of SVF3 is the OCCV. It allows to determine the corresponding CAV OCAV. The Tables 16 and 17 show these two optimal views for the depth criterion.

A mapping of OCAV into feature model gives the optimal view of a FD. It corresponds to SPL feature diagram illustrated in Figure 8.

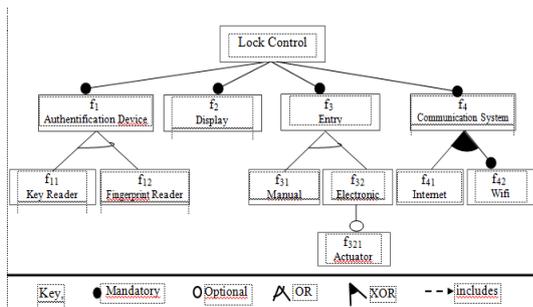


Figure 8. The optimal feature diagram view of (SHLC) SPL according to depth criterion.

By applying the same assessing process to other criteria, results are a set of (OCAV, OCCV, OVFD) for each criterion.

- The global optimization is a final sub-process of optimization. It aims to determine the optimal SPL feature diagram. It is achieved through composition of the four optimal feature diagrams according to criterion with integration mechanisms of [1]. The resulted view corresponds to a global optimal SPL feature diagram.

Now, we'll apply the optimization mechanisms previously described to the rest of criteria in SHLC SPL.

By applying the same assessing process (operated for the depth criterion) to other criteria, the produced results are as follows:

- Criterion: Usefulness.
  - VF={f<sub>1</sub>, f<sub>2</sub>, f<sub>3</sub>, f<sub>4</sub>, f<sub>11</sub>, f<sub>12</sub>, f<sub>31</sub>, f<sub>32</sub>, f<sub>41</sub>, f<sub>42</sub>, f<sub>321</sub>}.
  - CC:

$$\begin{aligned}
 \text{CC1} : f_{11} &\leftarrow - - \rightarrow f_{12} \\
 \text{CC2} : f_{31} &\leftarrow - - \rightarrow f_{32} \wedge f_{321}. \\
 \text{CC3} : f_{321} &- - \rightarrow f_{31}. \\
 \text{sSVF1} &= \{f_2\}, \text{SVF2} = \{f_{41}\}, \\
 \text{SVF3} &= \{f_{42}, f_2\} \text{ and } \text{SVF4} = \{f_{42}\}
 \end{aligned}$$

By applying the same evaluation process used for the depth criterion, the outputted results are grouped in the assessing report of Table 19.

Table 19. The usefulness assessment report.

Usefulness	ASP <sub>1</sub>	ASP <sub>2</sub>	ASP <sub>3</sub>	MC	Max RS	Max RR
Initial	0.72	0.45	0.72	3	0.45	0.81
SVF1	0.72	0.54	0.72	4		
SVF2	0.72	0.63	0.72	4		
SVF3	0.81	0.45	0.72	4		
SVF4	0.81	0.63	0.72	5		

By applying the same optimization process employed for the depth criterion, results show that SVF4 corresponds to OSVF and the SVF4 CAV corresponds to OCAV. A mapping of OCAV into feature model gives the optimal view of a FD. It corresponds to SPL feature diagram illustrated in Figure 9.

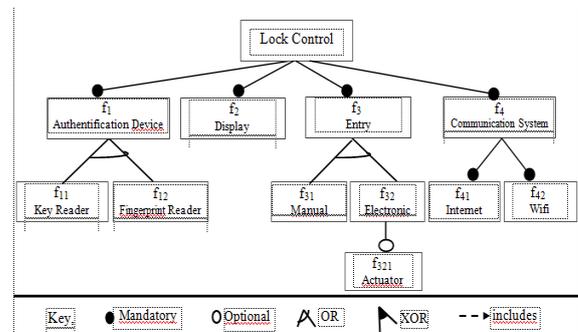


Figure 9. The optimal feature diagram view of SHLC SPL according to usefulness criterion

- Criterion: HL Coverage.
  - For HL<sub>#1</sub> :
    - VF={f<sub>2</sub>}
    - SVF1=VF.
    - One valid (VC) = {f<sub>2</sub>} of (SVF1) in (ICAV).
  - HL<sub>#2</sub>
    - VF={f<sub>11</sub>, f<sub>12</sub>, f<sub>31</sub>, f<sub>32</sub>, f<sub>41</sub>, f<sub>42</sub>}.
    - CC:
    - CC1 : f<sub>11</sub> ← - - → f<sub>12</sub> (exclusive OR).
    - CC2: f<sub>31</sub> ← - - → f<sub>32</sub> (Exclusive OR)

By considering CC in ICAV, the configurations of (f<sub>11</sub>, f<sub>12</sub>) and (f<sub>31</sub>, f<sub>32</sub>) in ICAV do not offer VC.

Then, we can have only 3 SVF:

SVF1={f<sub>41</sub>} with 01 (VC)={f<sub>41</sub>}, SVF2={f<sub>42</sub>} with 01 VC {f<sub>42</sub>} and SVF3 ={f<sub>41</sub>, f<sub>42</sub>} with 01 (VC)={f<sub>41</sub>, f<sub>42</sub>} .

- For HL<sub>#3</sub>:  
VF={f<sub>321</sub>} ; SVF1=VF.

One valid (VC)={f<sub>321</sub>} of SVF1 in ICAV.

The evaluation gives the results grouped in the assessing reports for the three HL in Table 20.

Table 20. The HL coverage reports.

Usefulness	ASP <sub>1</sub>	ASP <sub>2</sub>	ASP <sub>3</sub>	MC	Max RS	Max RR
<b>HL<sub>#1</sub></b>					1	1
<b>Initial</b>	1	0.75	1	3		
<b>SVF1</b>	1	1	1	4		
<b>HL<sub>#1</sub></b>					0.50	0.66
<b>Initial</b>	0.50	0.33	0.50	3		
<b>SVF1</b>	0.50	0.50	0.50	4		
<b>SVF2</b>	0.66	0.50	0.66	4		
<b>SVF3</b>	0.66	0.50	0.50	5		
<b>HL<sub>#1</sub></b>					1	1
<b>Initial</b>	1	0	1	3		

An optimization and a mapping of OCAV into feature model gives the optimal view of a FD illustrated in Figure 10.

- **Criterion:** Scanning.
  - VF={SU<sub>1</sub>, SU<sub>2</sub>, SU<sub>3</sub>, SU<sub>4</sub>, SU<sub>5</sub>, SU<sub>6</sub>, SU<sub>7</sub>,SU<sub>8</sub>, SU<sub>9</sub>}
  - CC:
  - CC1: SU<sub>1</sub> ← -- → SU<sub>2</sub> (exclusive OR).
  - CC3: SU<sub>4</sub> ← -- → SU<sub>5</sub>.
  - CC4: SU<sub>4</sub> ← -- → SU<sub>6</sub> (exclusive OR).
  - CC : SU<sub>7</sub> - - - -> SU<sub>3</sub>
  - CC6: SU<sub>9</sub> - - - -> SU<sub>3</sub>.
  - SVF1={SU<sub>3</sub>}, SVF2={SU<sub>5</sub>}, SVF3={SU<sub>7</sub>}
  - SVF4={SU<sub>8</sub>}, SVF5={SU<sub>9</sub>,SU<sub>3</sub>}.

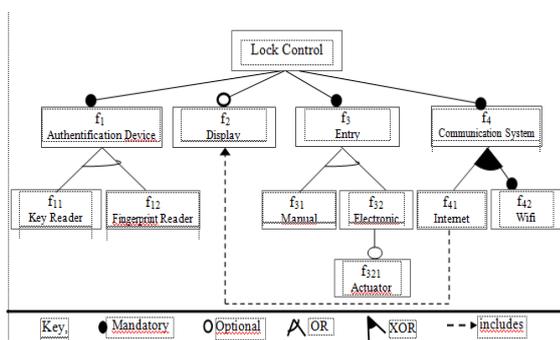


Figure 10. The optimal feature diagram for HL coverage of (SLHC) SPL.

By applying the same evaluation process for this criterion, we'll obtain the assessing report shown in Table 21.

Table 21. The scanning assessment report.

Scanning	ASP <sub>1</sub>	ASP <sub>2</sub>	ASP <sub>3</sub>	MC	Max RS	Max RR
<b>Initial</b>	0.33	0	0.22	3		
<b>SVF1</b>	0.33	0	0.33	5		
<b>SVF2</b>	0.22	0	0.22	4		
<b>SVF3</b>	0.33	0.11	0.22	6	0.44	0.44
<b>SVF4</b>	0.33	0.11	0.22	5		
<b>SVF5</b>	0.22	0.22	0.44	4		

The optimization and the “analysis model-feature model” mapping of (OCAV) give the optimal feature diagram shown in Figure 11

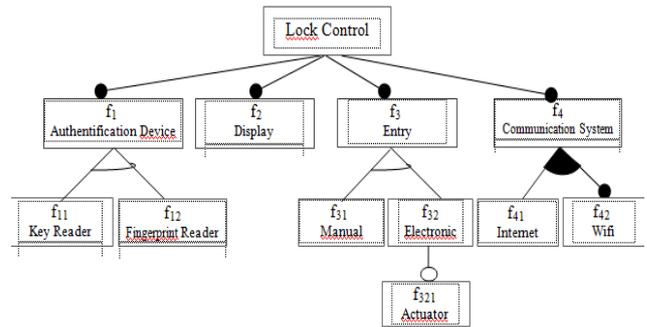


Figure 11. The optimal feature diagram for the scanning criterion of (SHLC) SPL.

- **Global optimization.** By composing the four optimal SPL feature diagrams according to criterion with the integration mechanisms of [1]. The resulted FD view corresponds to the global optimal SPL feature diagram .It is represented in Figure 12.

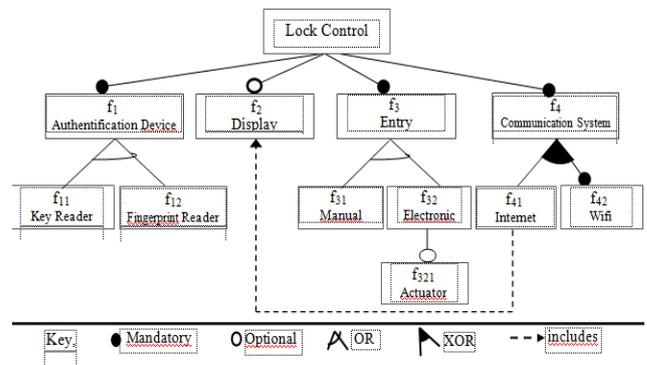


Figure 12. The optimal feature diagram of (SHLC) SPL.

## 8. Our Support-tool

For supporting concepts used by our approach, we've implemented a support-tool called 'FeMoMaS' This first version is focused on creating an initial SPL through adding features, successors and their dependencies. FeMoMaS allows to delete and update features. The building of product is also allowed across configuration of features. The views related to fine-grained analysis are also obtained by a transformation of usual configurations into binary forms as for usefulness view in Figure 13.

Figure 13. FeMoMaS: generation process of the product usefulness metrics.

Figure 14. FeMoMaS: generation process of usefulness CCV.

For this type of granularity, FeMoMaS allows to transform (CAV) into (CCV) as in Figure 14, generate analysis reports and determine optimal views. However, limitations of FeMoMaS concern the inability to composing and integrating FDs. This is related to the fact that the tool is not design to support graphical formalisms. At the current time, we advocate to use the integration of FDs composition approach as proposed by [1].

### 9. Related Works

There are not many approaches related to the practical development of a SPL. Some of them are based either on models or evaluation metrics. Others are based on a combination of the two concepts. We'll focus in follows on a significant of them.

Berger *et al.* [3, 4] are based on bottom-up methodology to create a SPL. They propose a series of metrics related to similarity of existing software products. Evaluation is achieved according to only a relationship model and allows to appreciate the effectiveness of SPL to build. The mainly aim of these works is to evaluate the opportunity to organize the existing products in form of SPL.

Fischbein *et al.* [10] proposed an analogous model (the modal transition systems) of [3] but based on mathematical concepts and focused on the behavior of the SPL instead of its architecture development.

For [21], the mainly aims is a quality of SPL. To achieve this it proposes a specific model of variability

and a set of metrics to assess a quality of already derived products.

Other approaches are only dedicated to measure object-oriented software systems. The work of [9] is based on interface specifications and that of [18] employed the concept of 'service utilization'. In the work of [17], the approach aims to built and apply a quality model to analyze a web application and to evaluate the structural software qualities using the built model. The latter is based on a set of software metrics and a prediction system. The approach provides a significant framework to built quality models but it focused on one software system. It needs adaptation in order to be applied to SPL domain.

Van-der-Hoek *et al.* [18] presented an approach to develop the architecture for a family of systems belonging to a changing domain. The approach is based on use cases and feature modeling. The structure of products in the domain is evolved in the form of subsystem but only UML component views.

Her *et al.* [11] developed a set of metrics dedicated to a core asset. It is structured into a product line architecture and two models: a decision model and a component model.

Khalifaoui *et al.* [13] proposed to use automatic framework of model transformations to generate automatically all valid products from an existing SPL feature diagram. To achieve this, it used a graph grammar with ATOM3 tool. This recent approach is not based on metrics. However, it can be applied to

capture the configurability of existing SPL but it is not efficient to develop an expressive structure of SPL.

Mann and Rock [16] has a same aim of our approach but not focused on reusability metrics to build SPL structure. It is based on the measurement of required effort to develop SPL. The outputted effort metrics are used to integrate potential features and functionalities into an existing SPL.

Our approach is based on a bottom-up methodology. Instead of starting from scratch, it uses a predefined set of existing products like in [3, 4]. The products can be defined by stakeholders regardless of their project perspective. However, in the difference with similar approaches, our mainly aim is to build an optimal SPL through an optimal feature diagram. In order to achieve this aim, our approach develops at first an IFD from a variability modeling contained in existing products. This prior step defines a key representation for the analysis, configuration, assessment and optimization phases.

In addition, our work is not based on only the measurement and evaluation of similarity in product structures but consolidates this activity by a prior analyzing sub-process.

Furthermore and in order to give more credibility to this sub-process, we've structured it into four analysis criteria according to different granularity of product building units. Other differences with comparable works concerns the possibility to determine either optimal criterion views and optimal feature diagrams as well as desired criterion views and FD in addition to the capability for stakeholders to evaluate the required products to derive in order to attain the optimal reusability in the SPL.

## 10. Conclusions and future Work

In this paper, we've presented an approach that enables to create a SPL using a development process in which we use a FD as a key model. After transforming a predefined set of existing products into a primary feature diagram of SPL, the process uses an AM to study an internal structure of products according to different granularity analysis criteria.

Then, a configuration activity allows capturing configurability of the products according to VFs and constraints related to configuration. Results are a set of analysis views of products according to criteria. At this step, a first set of metrics related to configurability and criterion analysis are produced. Therefore, analysis views are adapted by a mapping model into correlation views in order to facilitate the assessment. At this step a set of metrics related to similarity, reusability, commonality and differentiability are produced. They will be used to determine OCCV. The matching analysis views according to criterion are used to determine the OCAVs. The latter are employed by the

'analysis AM s model-feature model' mapping to establish the optimal feature diagram for each criterion.

At the end of the process, the integration of all criterion optimal feature diagrams allows to produce an optimal SPL feature diagram. Thus, we've combined together results from the multi-criterion AM and CM as we consider that this achievement is able to improve substantially the SPL development process. In our further research work we will extend our approach to handle all of dependency types between features ("feature-group of features", "group of features-feature") and between scanning units ("scanning unit-group of scanning units", "group of scanning units-scanning unit") instead of only "feature-feature" and "scanning unit-scanning unit". In addition, we plan to evaluate the proposed approach with practitioners in the practical area to find out whether the approach seems usable and useful in a real-world domain.

## References

- [1] Acher M., Collet P., Lahire P., and France R., "Composing Feature Models," in *Proceeding of Second International Conference on Software Language Engineering*, Denver, pp. 62-81, 2010.
- [2] Atkinson C., Bayer J., Bunse C., Kamsties E., Laitenberger O., Laqua R., Muthing D., Paech B., Wust J., and Zettel J., *Component-based Product Line Engineering with UML*, Addison-Wesley Professional, 2001.
- [3] Berger C., Rendel H., and Rumpe B., "Measuring the Ability to Form a Product Line for a Set of Existing Products," in *Proceeding of 4<sup>th</sup> International Workshop on Variability Modeling of Software-Intensive Systems*, Linz, pp. 151-154, 2010.
- [4] Berger C., Rendel H., Rumpe B., Busse C., Jablonski T., and Wolf F., "Product Line Metrics for Legacy Software in Practice," in *Proceeding of the 14<sup>th</sup> International Software Product Line Conference*, Jeju Island, 2010.
- [5] Bergey J., Cohen S., Donohoe P., and Little R., *Software Engineering Institute (SEI) of Carnegie Mellon Repository, Software Product Lines: Report of the 2009 U.S. Army Software Product Line Workshop*, 2009.
- [6] Bosch J., *Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach*, Addison-Wesley Professional, 2000.
- [7] Clauß M., "Modelling variability with UML," in *Proceeding of Young Researchers Workshop 3<sup>rd</sup> GCSE*, Erfurt, 2001.
- [8] Czarnecki K. and Eisenecker U., *Generative Programming: Methods, Tools and Applications*, Addison-Wesley, 2000.
- [9] Dincel E., Medvidovic N., and Van-der-Hoek A., "Measuring Product Line Architectures," in

- Proceeding of the 4<sup>th</sup> International Workshop on Software Product-Family Engineering*, London, pp. 346-352, 2002.
- [10] Fischbein D., Uchitel S., and Braberman V., "A Foundation for Behavioural Conformance in Software Product Line Architectures," in *Proceeding of ISSTA on Role of software Architecture for Testing and Analysis*, Maine, pp. 39-48, 2006.
- [11] Her J., Kim J., Oh S., Rhew S., and Kim S., "A Framework for Evaluating Reusability of Core Asset in Product Line Engineering," *Informatics Software Technology Elsevier.*, vol. 49, no. 7, pp. 740-760, 2007.
- [12] Hubaux A., Heymans P., Schobbens P., Derrider D., and Abbasi E., "Supporting Multiple Perspectives in Feature-Based Configuration," *Software and Systems Modeling*, vol. 12, no. 3, pp. 641-663, 2013.
- [13] Khalfaoui K., Chaoui A., Foudil C., and Kerkouche E., "Automatic Generation of SPL Structurally Valid Products Using Graph Transformations Approach," in *Proceeding of 4<sup>th</sup> International Conference of Computer Science and its Applications*, Saida, pp. 333-342, 2013.
- [14] Linden F. and Obbink J., "Engineering Software Architectures, Processes and Platforms for System-Families-Introduction to Domain Analysis," in *Proceeding of the International Workshop on Software Architectures for Product Families*, pp. 244-253, 2000.
- [15] Loughran N., Sanchez P., Garcia A., and Fuentes L., "Language Support for Managing Variability in Architectural Models," in *Proceeding of 7<sup>th</sup> International Conference on Software Composition*, Budapest, pp. 36-51, 2008.
- [16] Mann S. and Rock G., "Dealing with Variability in Architecture Descriptions to Support Automotive PL: Specification and Analysis Methods," in *Proceeding of Embedded World Conference*, Nuremberg, 2009.
- [17] Marchetto A. and Trentini A., "A Framework to Built Quality Models for Web Applications," *The International Arab Journal of Information Technology*, vol. 4, no. 2, pp. 168-176, 2007.
- [18] Van-der-Hoek A., Dincel E., and Medvidovic N., "Using Service Utilization Metrics to Assess the Structure of Product Line Architectures," in *Proceeding of 9<sup>th</sup> International Workshop on Enterprise Networking and Computing in Healthcare Industry*, Sydney, pp. 298-308, 2003.
- [19] Van-der-Linden F. and Henk-Obbink J., "ESAPS-Engineering Software Architectures, Processes and Platforms for System-Families," in *Proceeding of the International Workshop on Software Architectures for Product Families*, London, pp. 244-253, 2000.
- [20] Vranic V. and Marco V., "Developing a Product-Line Based Architecture in a Domain under Research," *Slovak University of Technology*, 2006.
- [21] Zhang T., Deng L., Wu J., Zhou Q., and Ma C., "Some Metrics for Accessing Quality of Product Line Architecture," in *Proceeding of Computer Science and Software Engineering International Workshop*, Hubei, pp. 500-503, 2008.
- [22] Ziadi T. and Jezequel J., "Manipulation De Lignes De Produits Logiciels-Une Approche Dirigée Par Les Modèles," *Institut de Recherche en Informatique et Systèmes Aléatoires*, 2005.
- [23] Ziadi T., "Manipulation Des Lignes De Produits En UML," *Thèse de Doctorat*, Université de Rennes1-France, 2004.



**Mohamed Kimour** received his PhD degree on computer science in 2005, from the department of computer science, Badji Mokhtar-Annaba University, Algeria. He is currently a professor and the head of a research project on mobile

healthcare systems at the same university. He is also a head of a research team on networked embedded systems and wireless networks, in the laboratory of embedded systems. He has led many research projects and authored more than 60 conference and journal papers. His research interests include: embedded systems engineering, wireless networks, and embedded vision.



**Nouredine Guersi** received his PhD degree on electronic in 2006, from the department of computer science, Badji Mokhtar-Annaba University, Algeria. He is currently a professor at the electronic department of the same university. He is a head of a team research on dependability and diagnosis of embedded systems. He is interested in control engineering and diagnosis of embedded systems.



**Yacine Djebar** received his MS degree on computer science from the department of computer science, Badji Mokhtar-Annaba University, Algeria. He is currently an associated professor at the 8 May 1945-Guelma University, Algeria.

He is a member of a research team on embedded systems and wireless networks, in the laboratory of embedded systems, Badji Mokhtar-Annaba University. He is also a member of the research project on mobile healthcare systems. His research interests include: software product line engineering, information systems, and embedded systems.