The Impact of Non-Functional Attributes on the Analysis Operations of Feature Models

I. Aachour¹, L. Labled², and H. Ben Ghazela¹

¹Computer Science Department, Manouba University/ ENSI/ Lab. RIADI-GDL, Manouba, Tunisia
²Computer Science Department, Tunis University/ ISG/ Lab. RIADI-GDL, Tunis, Tunisia

Abstract - The functional aspect of a system is very important. In fact, it defines different features of the system, but it does not negate the reality of the non-functional aspects of it and that has an impact on this functional aspect. This aspect has been largely treated with classical systems but not enough with the product lines.

So we had the idea to study the impact of non-functional attributes on the analysis operations of feature models. In this work, we have resumed analysis operations of feature models listed in the literature. Moreover, we studied the effect of adding the non-functional attributes on these operations by giving examples. So this has enabled us to emphasize the presence of three types of constraints namely constraint value, constraint attribute-attribute and constraint feature-attribute.

Finally, we have deduced that some operations are not affected, others are affected and there is also the emergence of new one(s).

Keywords: Non-functional attributes, analysis operations, extended feature models.

1 Introduction

The consideration of non-functional attributes is crucial in features models. In fact, this models are the basis of our reference architecture. This is why we are interested in studying the impact of adding non-functional attributes (NFA) on analysis operations of features models. We based our work on the first challenge of Benavides [1] "Include feature attribute relationships for analyses on feature models and propose new operations of analysis leveraging extended feature models".

In this work, we will resume analysis operations described by Benavides [1] and we will present the effect of non-functional attributes on this operation and an example.

Section 2 concerns a short descriptive of the Extended Feature Model (EFM). In Section 3, we present the constraint on the attributes of the extended feature models. Section 4, deals with detailing the impact of non-functional attributes on the analysis operations of the feature models. Finally, Section 5 summarizes our work and outlines our prospects for future work.

2 Extended feature model

This model is a feature model (see figure 1). Each feature can be enriched by attributes. Each attribute has a type and a domain. Each feature can have three types of relations with their son (mandatory, optional, relation group that can be expressed by an alternative or an or-relationship). In addition, features can be connected by a relation of necessity (requires) or of exclusion (excludes) [1, 4].

3 Constraints on the attributes of the extended feature models

The NFA of the extended feature model may present constraints. These constraints are either attribute values or relations attribute-attribute or relations feature-attribute. And the presence of these constraints can influence the analysis operations of the extended feature models.

To explain these constraints, we present the following examples for each case:
- Value constraint: a feature with attribute run time must have value <= 10ms or belonging to the interval [10ms 5ms ..].
- Attribute-attribute constraint: a feature F1 with attribute accessibility (requires) a run time > 15ms of another feature F2.
- Feature-attribute constraint: a feature registration request (requires) a storage capacity >= 50 Mega Byte of feature archiving.

4 The impact on non-functional attributes on analysis operations of an extended feature models

Studying the impact of NFA on the analysis operations of feature model, we noticed that some operations are affected, others are not and there is also the emergence of new one(s). We list below the various operations [1] while quoting for each one its input, its output, its role, the effect of NFA and an example.
4.1 Void feature model

**Input:** Feature model  
**Output:** Empty or not  
**Role:** To see if the feature model present at least one product.  
**Impact of NFA:** The presence of one of the three types of constraints can influence this operation in the sense that a constraint may omit the presence of a product. This is valid if the product that has the constraint is the only product in the model and its omission will cause the empty model.  
**Example:** If the value of an attribute $a_1$ should belong to the interval $[\min .. \max]$. If a product $P_1$ has the attribute $a_1$ with value less than $\min$ so the product $P_1$ is invalid.

4.2 Valid product

**Input:** Feature model and a product  
**Output:** A product belong or not to the feature model.  
**Role:** To see if the product belong or not to the list of all products representing the feature model.  
**Impact of NFA:** The presence of one of the three constraints mentioned in section 3 may affect the validity of the product.  
**Example:** In the feature model, the value of an attribute $a_1$ should belong to the interval $[\min .. \max]$. If a product $P_1$ has the attribute $a_1$ with value less than $\min$ so the product $P_1$ is invalid.

4.3 Valid partial configuration

**Input:** Feature model and partial configuration  
**Output:** Configuration invalid or not  
**Role:** Check the validity of a partial configuration is to verify that the configuration has no contradiction as the presence of a requires relation between a feature in the set $S$ and a feature in the set $R$ (for the meaning of $S$ and $R$ refer to the explanation of the term configuration).  
**Impact of NFA:** Configuration can have features with NFA. These are connected by constraints of type attribute-attribute or type feature-attribute and that causing contradictions.  
**Example:** Let $F_1$ a feature belonging to the set $S$ connected by a relation requires to an attribute $a_2$ of a feature $F_2$ belonging to the set $R$. The requires relationship present a contradiction.

4.4 All products

**Input:** feature model  
**Output:** All products are represented by the feature model

---

1 Configuration: Given a feature model with a set of features $F$, a configuration is the pair $(S, R)$, where $S, R \subseteq F$, where $S$ is the set of features that can be selected and $R$ the features that should not be presented with $S \cap R = \emptyset$. Full configuration is represented by $S \cup R = F$ and Partial configuration represented by $(S \cup R) \subseteq F$. 
**Role:** This operation generates all products that the feature model can represent.

**Impact of NFA:** The presence of one of the three constraints mentioned in section 3 can vary the list of products generated and that by omitting products that do not meet these constraints.

**Example:** Let an attribute a1 of feature F1 connected by a relation of exclusion to an attribute a2 of feature F2. The relation of exclusion has spread to the features F1 and F2 and the product with both F1 and F2 is omitted.

### 4.5 Number of products

**Input:** Feature model

**Output:** The number of products represented by the model

**Role:** This operation counts the number of all the products that can represent the feature model.

**Impact of NFA:** The presence of one of this three constraints which are mentioned in section 3 affects this process in the same way as its influence on the previous operation. Indeed, it has the same behavior as the operation All products except that instead of listing all the products, the operation gives their total number.

**Example:** In the feature model, the value of an attribute a1 of a feature F1 should belong to the interval [min .. max]. If a product P1 with an attribute a1 with value greater than max then the product P1 is not counted.

### 4.6 Filter

**Input:** Feature model and configuration

**Output:** The set of derivatives of the feature model including the initial configuration

**Role:** From the feature model, this operation generates products that meet the initial configuration.

**Impact of NFA:** The presence of one of the three constraints mentioned in section 3 may affect this operation. In fact, some products including the feature model and the configuration can be excluded because of the constraints of their attributes. In fact, this has the same principle that operations All products and Number of products.

**Example:** We can adopt the same examples that the operations All products and Number of products.

### 4.7 Anomalies detection

The literature [1] postponed five analysis operations to detect anomalies in the feature model such as redundancies or contradictions.

**Input:** feature model

**Output:** information about the detected anomaly

#### 4.7.1 Dead feature

This is a feature that does not appear in any product of the line products. This anomaly is caused by misuse of the requires and the excludes constraints of features.

**Impact of NFA:** The constraints of type attribute-attribute or type feature-attribute can make a dead feature.

**Example:** Whether a mandatory feature F1 connected by a relationship excludes with an attribute of an optional feature F2. So, F2 is necessarily a dead feature.

#### 4.7.2 Conditionally dead features

This is a feature that becomes dead under certain circumstances such as the selection of another feature.

**Impact of NFA:** Here also, the constraints of type attribute-attribute or type feature-attribute can make a conditionally dead feature.

**Example:** Considering a feature F1 connected by a relationship excludes with an attribute of a feature F2. Assuming we always select the feature F1, the relationship of the exclusion will be propagated to feature F2. So, F2 will be conditionally dead feature.

#### 4.7.3 False optional features

This is a feature that is included in all products of the product line.

**Impact of NFA:** Here also, the constraints of type attribute-attribute or type feature-attribute can make a false optional feature.

**Example:** Whether a mandatory feature F1 connected by a relationship requires with an attribute of an optional feature F2. The inclusion relation will be propagated to feature F2 that is always present. So, F2 is a false optional feature.

#### 4.7.4 Wrong cardinalities

A group of features described as wrong cardinalities, is a group of cardinality that can not be instantiated. For example, we have an alternative of three features: A, B, and C. Which two are mutually exclusive and we have a cardinality <1..3>. So, the selection of three features is not possible.

**Impact of NFA:** Here also, the constraints attribute and attribute-relationship-attribute feature can present a wrong cardinalities.

**Example:** Considering an alternative of three features F1, F2 and F3, that has a cardinality <1..3> and whose feature F1 is connected by a relationship excludes with an attribute of a feature F3. The exclusion Relationship will be propagated to the feature F3 and we cannot select both the three features F1, F2 and F3.

#### 4.7.5 Redundancies

A feature model that contains redundancies is a feature model that represents the same information in many ways.

**Impact of NFA:** Here also, the constraints of type attribute-attribute or type feature-attribute may cause duplication.

**Example:** Considering a mandatory feature F1 connected by a relationship requires with an attribute of a mandatory feature F2. The inclusion relation is propagated to feature F2 which is already mandatory.

### 4.8 Explanations

**Input:** Feature model and an analysis operation

**Output:** An explanation of the operation answer
Role: Explanations are generally related to anomalies and are explanations of the cause of these anomalies.
Impact of NFA: Here also, the constraints of type attribute-attribute or of type feature-attribute may be the cause of the problem. Such as causing a dead feature.
Example: The cause of a dead feature may be an exclusion relation of the constraint of type attribute-attribute or of type feature-attribute.

4.9 Corrective explanations

Input: Feature model and an analysis operation
Output: A list of corrections to explanations
Role: This operation suggests a list of corrections to the anomalies identified.
Impact of NFA: Assuming that the constraints of type attribute-attribute or of type feature-attribute may be the cause of the problem, their removal may be a correction.
Example: Correcting a dead feature may be the deleting of the exclusion relation of the constraints of type attribute-attribute or of type feature-attribute.

4.10 Feature model relationships

Input: Two feature models
Output: Information on how these two models are linked

4.10.1 Refactoring

A feature model is a refactoring of another, if they represent the same set of products even though they have different structures.

Impact of NFA: Here also, the constraints of type attribute-attribute or of type feature-attribute can influence the refactoring relationship between two feature models and that by altering or enhancing this relation
Example 1: an exclusion constraint of type attribute-attribute or of type feature-attribute of a feature model FM1 which has no equivalent in another feature model FM2. This varies the list of products of FM1.
Example 2: an exclusion constraint of type attribute-attribute or of type feature-attribute of a feature model FM1 has the same effect on the list of products as a relation of exclusion of two features.

4.10.2 Generalization

A feature model FM1 is a generalization of another feature model FM2, if all products of FM1 maintain and extend all products of FM2.
Impact of NFA: Here also, the constraints of type attribute-attribute or of type feature-attribute may affect the generalization of two feature models.
Example: Considering two identical feature models FM1 and FM2. Adding an exclusion constraint of type attribute-attribute or of type feature-attribute on feature model FM2, we will vary the list of products of FM2 by eliminating at least two products. Thus, FM1 is a generalization of FM2.

4.10.3 Specialization

A feature model FM1 is a specialization of another feature model FM2, if all products of FM1 is a subset of products of FM2.
Impact of NFA: Here also, the constraints of type attribute-attribute or of type feature-attribute can influence the specialization of two feature models.
Example: Considering two identical feature models FM1 and FM2. Adding an exclusion constraint of type attribute-attribute or of type feature-attribute on feature model FM1, we will vary the list of products of FM1 by eliminating at least two products. Thus, FM1 is a specialization of FM2.

4.10.4 Arbitrary edit

There is no relationship between the two feature models.
Impact of NFA: we believe that the constraints of the NFA did not affect this relationship. Indeed, the constraints alone can not make two feature models as arbitrary edit.

4.11 Optimization

Input: Feature model and objective function
Output: The product that meets the best to the criteria established by the objective function
Role: This suggests for a product a set of features that maximize or minimize the value of an attribute of a given feature.
Impact of NFA: this operation is only useful in the context of an extended feature model. Indeed, it is according to the values of the attributes and within both the constraints on features and constraints on the attributes that we select or omit some feature.
Example: Assuming that we have a cost minimization function we must choose the features with minimum cost: having an attribute with a minimum cost.

4.12 Core features

Input: Feature model
Output: The set of features present in all products of the product line.
Role: For a given feature model, this operation list all features that appear in all products of the product line. This is useful for determining the features that will be developed in the first place and which will form the reference architecture.
Impact of NFA: Here also, the constraints of type attribute-attribute or of type feature-attribute can make a feature as core feature and this by forcing its publication in all the products of the product line by the presence of a relationship requires.
Example: a mandatory feature F1 connected by a relationship requires to an attribute of an optional feature F2. The inclusion relation is propagated to feature F2 that is always present and belong to the list of core features.
4.13 Variant features

**Input:** Feature model  
**Output:** The set of features not present in all products of the product line.  
**Role:** For a given feature model, this operation list all features that do not appear in all products of the product line.  
**Impact of NFA:** We think that the constraints of the NFA had no effect on this type of operation.

4.14 Atomic sets

**Input:** Feature model  
**Output:** List of atomic sets.  
**Role:** Giving a feature model, this operation lists the atomic sets. A set is a group of atomic features (at least one) considered as a single unit in some analysis. Intuitively, the mandatory features and their parents are grouped in an atomic set. This operation provides a lightweight version of the feature model that will make more efficient use of other analysis operations.  
**Impact of NFA:** Here also, the constraints of type attribute-attribute or of type feature-attribute may influence the membership or not of a feature in an atomic set.  
**Example:** Giving a mandatory feature F1 connected by a requires relationship with an attribute of an optional feature F2. The inclusion relation is propagated to feature F2. So, F2 will be present in the same atomic set of F1.

4.15 Dependency analysis

**Input:** Feature model and a partial configuration  
**Output:** New configuration  
**Role:** From the feature model and the partial configuration, this operation generates a new configuration highlighting the features to include and exclude and taking into account the constraints of the feature model.  
**Impact of NFA:** Here also, the constraints of type attribute-attribute or of type feature-attribute can influence the structure of the new configuration.  
**Example:** Let a feature model FM, a partial configuration PC of FM and a feature F1 belonging to the set S. If F1 is connected by a requires relationship with an attribute of a feature F2. The inclusion relation is propagated to feature F2. So, F2 will belong to the set S of the new configuration.

4.16 Multi-step configurations

**Input:** Feature model, an initial configuration, a final configuration, a K step configurations to meet a global constraint and a function determining the cost of transition to a configuration from step T to step U.  
**Output:** An ordered list of K configurations representing the different stages of transition from initial configuration to the final configuration.  
**Role:** Based on various inputs, this operation offers an ordered list of K configurations representing the different stages of transition from initial configuration to the final configuration.  
**Impact of NFA:** The constraint value influence the global constraint. Also, the constraints of type attribute-attribute or of type feature-attribute may influence the structure of intermediate configurations.  
**Example:** If the feature model includes constraints related to attribute-attribute and to attribute-feature must be respected and this affects the list of configurations presented as a result.

4.17 Other operations

In this section, we include operations that have calculations based on the values of previous operations.

4.17.1 Homogeneity

**Input:** Feature model  
**Output:** Homogeneity degree of of the feature model  
**Role:** This is the complement of the ratio between the number of unique features (a feature is unique if it appears only in one product) in a product by the total number of products in the feature model. A feature model is more homogeneous than the number of unique features in a product is minimal.  
**Impact of NFA:** The presence of attributes and their constraints affects indirectly the result of this operation since it affects the operation of calculating the number of products: Number of products.

4.17.2 Commonality

**Input:** Feature model and configuration  
**Output:** The products percentage represented by the feature model and including the input configuration.  
**Role:** This is the ratio of product including the input configuration by the total number of products of the feature model. This transaction enables us to classify the features that will be developed in the first place and decide who will be part of the basic architecture.  
**Impact of NFA:** The presence of attributes and constraints affect indirectly the result of this operation since it affects the operation of calculating the number of products: Number of products and also the operation that gives the products of feature model including the initial configuration: Filter.

4.17.3 Variability factor

**Input:** Feature model  
**Output:** The ratio of the number of products by 2^n which n is the number of features considered  
**Role:** This is the ratio of the number of products by 2^n which n is the number of features considered. In particular, 2^n indicates the potential number of products represented by the feature model and assuming that any combination of features is allowed. Generally, the root and the features that are not leaves are not considered. A small factor indicates that the number of combinations is very limited compared to the total number of potential products.  
**Impact of NFA:** The presence of attributes and constraints affecting indirectly the result of this operation since it affects the operation of calculating the number of products: Number of products.
4.17.4 Degree of orthogonality

**Input:** Feature model and a sub-tree

**Output:** Degree of orthogonality

**Role:** According to Czarnecki and al. [2], the degree of orthogonality is the ratio of the total number of products of the feature model by the number of products of the sub-tree knowing that only local constraints of the sub-tree are considered.

**Impact of NFA:** The presence of attributes and constraints affects indirectly the result of this operation because it affects the operation of calculating the number of products: Number of products.

4.17.5 Extra constraint representativeness (ECR)

**Input:** Feature model

**Output:** Degree of representativeness of the constraints of the tree

**Role:** This determines the degree of representativeness of the constraints of the tree. Mendonça and al. [5] defines Extra Constraint Representativeness (ECR) as the ratio of the number of features involved in the constraint (the repeated features are only counted once) by the number of features of the feature model.

**Impact of NFA:** The presence of attributes and constraints affects the result of this operation. En fact, it affects the operation of calculating the number of products: Number of products and also the number of features involved within the constraints (constraint of features and attributes).

4.17.6 Lowest common ancestor (LCA)

**Input:** Feature model and a set of features

**Output:** The feature being the lowest common ancestor of input features

**Role:** This determines the lowest common ancestor of input features. Mendonça and al. [5] defines the lowest common ancestor (LCA) of a set of features as the common ancestor which is farthest from the root: LCA (FM \{f_1, ..., f_n\}).

**Impact of NFA:** We think that the attributes have no effect on this operation.

4.17.7 Root features

**Input:** Feature model and a set of features

**Output:** The set of features that are roots in the feature model.

**Role:** This determines the set of features which are the roots of the feature model. Considering l = LCA (FM \{f_1, ..., f_n\}) Mendonça, and al. [5] define the roots of all the features roots (FM,\{f_1,...,f_n\}) as the subset of the features of the son of l and ancestor of the set \{f_1,...,f_n\}.

**Impact of NFA:** We can think that the attributes have no effect on this type of operation.

4.18 Attribute values

**Input:** a product and a attribute

**Output:** Values list of the attribute

**Role:** This is a new operation that lists all values of the attribute. This operation is useful if we want to do calculations on the values of an attribute for a given product.

**Example:** We can for a given product need the list values of the attribute cost to calculate the total cost.

5 Conclusion

In this work, we resumed analysis operations of feature models founded in the literature and we studied for each operation the impact of adding a NFA and basing ourselves on an example.

In fact, this work is only in its infancy and we are testing this impact on the Flame tool (FAMA Formal Framework) [3]. This will allow us to analyze the EFM and to add operations not yet taken into account.

Also, we plan to work more on the representation of NFA oriented quality.

6 References


