

Generating Documents from Partial Configurations

Appendix to “The Benefits of a Feature Model in Banking”

(Contribution to the 24th ACM International Systems and Software Product Line Conference SPLC2020)

CLAUDIA FRITSCH, KfW

RICHARD ABT, KfW

BURKHARDT RENZ, Technische Hochschule Mittelhessen

1 INTRODUCTION

Our paper “The Benefits of a Feature Model in Banking” (<https://doi.org/10.1145/3382025.3414946>) presents a proprietary document generator which composes documents from feature configurations in section 5. This paper describes its exceptional capability of producing documents that include variants by resolving open decisions in appropriate ways.

Normally, the generation of a document from a feature model configuration uses text elements attached to selected features. Our document generator, however, composes documents from partial configurations, also taking undefined features into account. The entire document will then be a composition of text snippets assigned to features.

The documents we look at here are KfW program information sheets. You can think of a program as a set of product variants. The composer will generate a certain program information sheet, if this program is selected in the configuration. The resulting feature model configuration is always a partial configuration, meaning that a few undefined features represent open decisions. Importantly, the composer should consider the undefined features as well as the selected features for generation. To get this right, we need methods to generate text paragraphs from undefined features. Section 2 describes these methods in detail.

2 THREE METHODS TO GENERATE TEXT FROM OPEN DECISIONS

By comparing the program information sheets provided by KfW with the partial configurations resulting from the selection of a program, we understood that we would need three different methods to generate text from open decisions. We have called these methods type A, B and C.

- A. Descendants of this feature define an enumeration and will be itemized.
- B. Children of this feature define a range whose maximum and/or minimum value will be generated.
- C. Permissible combinations of features must be identified and from these, a text will be composed.

We have tagged the root features of subtrees that can contain undefined features as type A, B or C. We have attributed the involved subfeatures with text snippets that can be composed to the sentences found in the program information sheets.

Basically, the composer inspects the partial configuration in the pre-order of the feature model and puts the text snippets attached to selected features one after the other in the document. If the composer detects a selected or undefined feature tagged as type A, B or C, it will treat it as described in one of the following three subsections.

Authors' addresses: Claudia Fritsch, KfW, Palmengartenstr. 5 - 9, Frankfurt am Main, Germany, claudia.fritsch@kfw.de; Richard Abt, KfW, Palmengartenstr. 5 - 9, Frankfurt am Main, Germany, richard.abt@kfw.de; Burkhardt Renz, Technische Hochschule Mittelhessen, Wiesenstr. 14, Gießen, Germany, burkhardt.renz@mni.thm.de.

2.1 Type A

If the composer detects a selected or undefined feature tagged as type A, it will collect its undefined or selected subfeatures and their undefined or selected descendants, concatenate the attached text snippets and include them in the generated document. Usually, the paragraph resulting from type A is an enumeration of the selected and/or undefined features, indented according to the depths of the features in the feature subtree. In a nutshell, it is an enumeration of options.

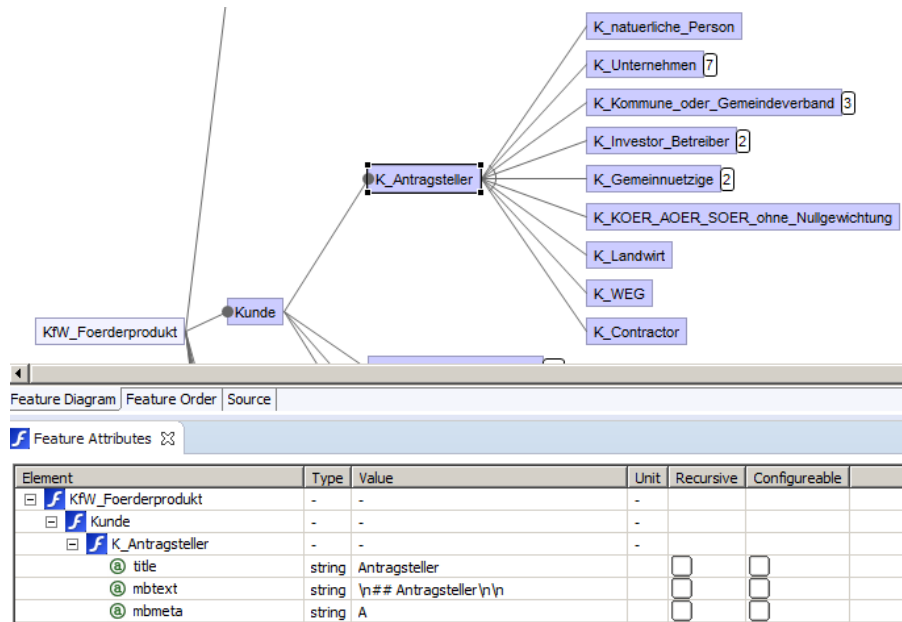


Fig. 1. Example of a type A attribute.

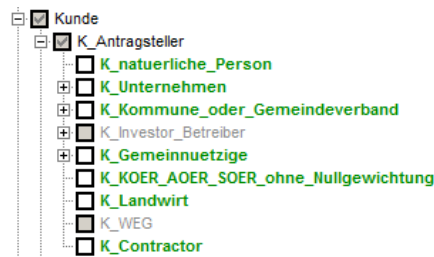


Fig. 2. Example of a type A configuration

Example: If a program allows several types of recipients (e.g., private individuals, private companies and non-profit-making organizations), these features will be undefined after the program selection and propagation. If only one type of recipient may apply, it will be selected after the program selection and propagation. The generated document lists all selected and undefined types of recipients. Therefore, the feature recipient is tagged as type A (A as in Aufzählung or Antragsteller, German for itemization or recipient, respectively). In Fig. 2 the undefined subfeatures are shown as white filled, unticked checkboxes.

2.2 Type B

Subfeatures of a feature tagged as type B define a range. Each subfeature can hold a text snippet for maximum and/or minimum. The composer checks the list of undefined subfeatures and determines the first and last item of the list to generate the text that describes the permitted range. The text snippet describing the minimum for the first undefined feature, and the text snippet describing the maximum for the last undefined feature are included in the generated document.

The features are ordered in the feature model and thanks to FeatureIDE's API the composer gets the list of undefined subfeatures in that same order, i.e. the first item represents the minimum and the last item the maximum of the allowed range. The model must ensure that

- the feature tagged as type B is the parent of an alternative group
- all descendants of the feature tagged as type B are direct subfeatures on the same level and
- the sequence of undefined subfeatures is continuous for any partial configuration.

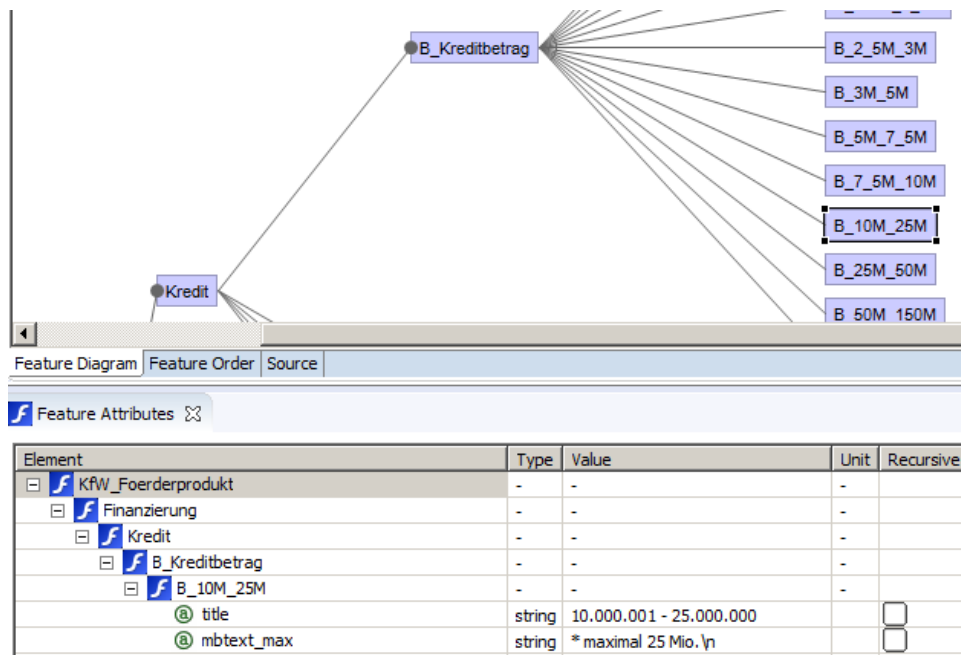


Fig. 3. Example of a type B attribute. The text snippet in attribute mbtext_max will be generated if the maximum loan amount is 25 Mio.

Example: Each retail program defines a maximum loan amount. Some programs also define a minimum loan amount. Figure 4 shows the maximum and minimum loan amounts of our programs. The features represent the left-open, right-closed intervals of loan amounts. If a program is selected, consecutive undefined subfeatures will define an open range, while all non permitted values will be unselected.

The composer prints the text snippet of the “maximum feature”, i.e., the undefined feature at the end of the open range. For that, feature loan amount (German: Kreditbetrag) is tagged as type B (B as in Bereich or Betrag, German for range or amount, respectively), and the “mbtext_max” attribute of each subfeature contains the maximum loan amount.

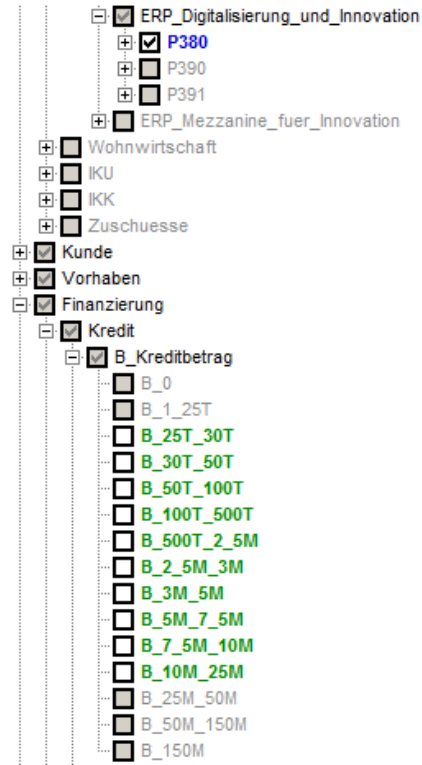


Fig. 4. Example of a type B range. The grey filled checkboxes are deselected, while the white filled checkboxes are undefined. The document generator will print the text snippet given as attribute `mbtext_min` in the first undefined feature, and the text snippet given as attribute `mbtext_max` in the last undefined feature.

2.3 Type C

If two or more features are tagged as type C, the composer determines each permissible combination of undefined subfeatures and concatenates the attached text snippets. Type C always involves more than one subtree.

In this case the composer uses `FeatureIDE` recursively to determine all permissible combinations. The composer selects the first undefined subfeature of the first involved feature and lets `FeatureIDE` propagate the selection on the feature model. The composer continues to select undefined subfeatures of the further involved features until the latter and their subfeatures are decided. The text snippets attached to the now selected features – manually or by propagation – are used to generate the text. After that, the composer backtracks, i.e., cancels all selections it has made, and continues by selecting the next undefined subfeature of the first involved feature.

The features involved in one composition of type C share a common label, so that the composer can

- recognize which features should be part of a combination and
- distinguish between several feature groups of type C.

Example: For each program variant, the combination of loan term (Laufzeit, LZ), grace period (Freijahre, FJ) and interest rate period (Zinsbindung, ZB) essentially determines its interest rate. The features Laufzeit, Freijahre and Zinsbindung are tagged as type C (C as in combination) and share the common label `C_LZ_FJ_ZB` shown in figure 5.

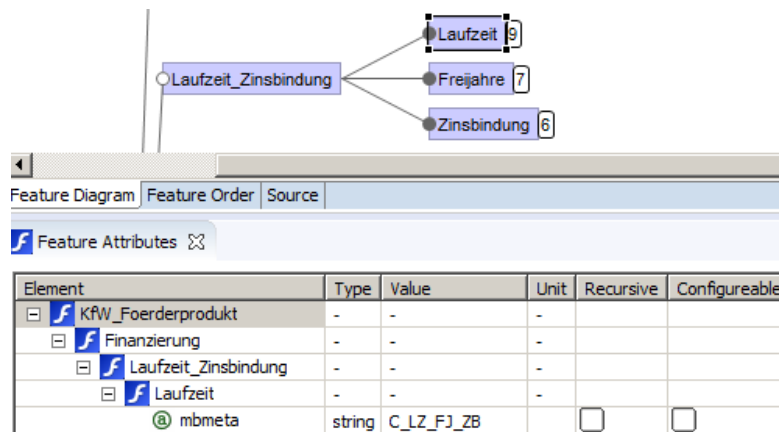


Fig. 5. Example of a type C attribute. Features Freijahre and Zinsbindung also carry label C_LZ_FJ_ZB.



Fig. 6. Example of a type C configuration and text generation. The left column shows the open decisions after a program has been selected. The middle column shows the three permitted combinations for the selected program in a condensed fashion. The right column shows the three generated sentences, in which each line represents the text snippet attributed to the feature in the middle column.

The left hand side of figure 6 shows the partial configuration of Laufzeit, Freijahre and Zinsbindung after a program has been selected. The possible values of Laufzeit, Freijahre and Zinsbindung in this program are undefined (white filled checkboxes) and impossible values are unselected (grey filled checkboxes). In order to determine all permitted combinations, the composer simulates decisions of undefined features.

Starting with LZ_05, and using the propagation mechanism of FeatureIDE, the composer finds the first combination, which is shown in the middle. The composer then backtracks, i.e., it cancels the decision LZ_05, and selects the next possible value, LZ_10. Continuing in this fashion, the composer finds the three permitted combinations shown in the middle column of figure 6. The text snippets associated with the selected features, shown on the right, are concatenated, and result in a proper description of the program variants.

3 SUMMARY

The real advantage of our document generator is its capability to generate documents from partial configurations including text associated with open decisions. We have thus identified three different methods of generating text paragraphs appropriately and introduced them as type A, B and C. In the course of our program standardization, all program information sheets will be generated from the feature model producing documents uniform in structure, wording, design and layout.