

Supplementary Material: the Configuration Algorithm

In this supplementary material, we provide the algorithm for the configuration of Product Specific (PS) use case and domain models. The algorithm is presented in seven functions. The main function, `config`, is given in Algs. 1 and 2. It collects the configuration decisions from the user and checks if they are consistent. Decisions made by the user are said to be consistent if they do not violate any dependency constraint (i.e., constraints imposed by the *require* and *conflict* relations specified in the Product Line use case diagram). After all decisions are collected and confirmed to be consistent, the algorithm generates the PS use case and domain models (the function calls for *generatePSDiagram*, *generatePSSpecifications* and *generatePSDomainModel* in Algs. 1 and 2). Six functions (Algs. 3-8) are called in `config` to check decision consistency:

- `checkConflictingVP` (Alg. 3) and `checkConflictingUC` (Alg. 4) identify the contradicting decisions with regards to the *conflict* relation for selected variation points and variant use cases;
- `checkRequiringVP` (Alg. 5) and `checkRequiringUC` (Alg. 7) identify the contradicting decisions with regards to the *require* relation for selected variation points and variant use cases, respectively;
- `checkRequiredVP` (Alg. 6) and `checkRequiredUC` (Alg. 8) determine the contradicting decisions with regards to the *require* relation for unselected variation points and variant use cases, respectively;

The check functions calls some other functions (Alg. 9 and Alg. 10) in order to infer use cases which are automatically selected or unselected.

- `inferUnselectedElements` (Alg. 9) infers unselected elements for the use case unselected in the given decision. Inferred, unselected use cases are in the variation points included by the unselected use cases. All the variant use cases of the inferred, unselected variation points are unselected and have no more further decisions to be made.
- `inferSelectedElements` (Alg. 10) infers selected elements for the use case selected in the given decision. Inferred use cases are with a mandatory variability relation in the variation points included by the selected use cases. Inferred, selected variation points are either variation points which have inferred, selected use cases, or mandatory variation points included by inferred, selected use cases.

Alg. 1: config

Inputs : Product Line use case diagram, PLD ,
Set of Product Line use case specifications, PLS ,
Product Line domain model, $PLDM$
Output: Set of Product Specific use case and domain models

- 1 Let PSD be the product specific use case diagram;
- 2 Let PSS be the product specific use case specifications;
- 3 Let $PSDM$ be the product specific domain model;
- 4 Let PSM be the set of product specific use case and domain models;
- 5 Let DC be the empty set for completed decisions for PLD ;
- 6 Let L be the set of pairs of variation points vp and use cases uc such that use cases are essential and they include the variation points, or the variation points are not included by any use case;
- 7 $T \leftarrow L$;
- 8 **while** $L \neq \emptyset$ **do**
- 9 $dp \in L$;
- 10 Let SUC be the set of variant use cases selected in $dp.vp$;
- 11 Let $NSUC$ be the set of variant use cases unselected in $dp.vp$;
- 12 Let d be the quadruple $(dp.vp, dp.uc, SUC, NSUC)$;
- 13 **if** (d satisfies the cardinality constraints in $dp.vp$) **then**
- 14 Let C be the empty set for contradicting decisions;
- 15 $C \leftarrow \text{checkConflictingVP}(dp.vp, DC, d, PLD)$;
- 16 $C \leftarrow C \cup \text{checkRequiringVP}(dp.vp, DC, d, PLD)$;
- 17 $C \leftarrow C \cup \text{checkRequiredVP}(dp.vp, DC, d, PLD)$;
- 18 **foreach** ($u \in SUC$) **do**
- 19 $C \leftarrow C \cup \text{checkConflictingUC}(u, DC, d, PLD)$;
- 20 $C \leftarrow C \cup \text{checkRequiringUC}(u, DC, d, PLD)$;
- 21 **end foreach**
- 22 **foreach** ($u \in NSUC$) **do**
- 23 $C \leftarrow C \cup \text{checkRequiredUC}(u, DC, d, PLD)$;
- 24 **end foreach**
- 25 **if** ($C = \emptyset$) **then**
- 26 $DC \leftarrow DC \cup \{d\}$;
- 27 $L \leftarrow L \setminus \{dp\}$;
- 28 Let $new_p = \{(vp, uc) \mid uc \text{ includes } vp \wedge uc \in SUC \wedge (vp, uc) \notin T\}$;
- 29 $L \leftarrow L \cup new_p$;
- 30 $T \leftarrow T \cup new_p$;
- 31 **else**
- 32 **updateDecisions**($C \cup \{d\}$);
- 33 **end if**
- 34 **else**
- 35 **updateDecisions**($\{d\}$);
- 36 **end if**
- 37 **end while**
- 38 $C \leftarrow \text{overallConsistencyCheck}(DC, PLD)$;
- 39 **if** ($C = \emptyset$) **then**
- 40 $PSD \leftarrow \text{generatePSDiagram}(DC, PLD)$;
- 41 **else**
- 42 **do**
- 43 **updateDecisions**(C);
- 44 $C \leftarrow \text{overallConsistencyCheck}(DC, PLD)$;
- 45 **while** ($C \neq \emptyset$)
- 46 $PSD \leftarrow \text{generatePSDiagram}(DC, PLD)$;
- 47 **end if**

Alg. 2: config(continued)

```
48 /* Configure PS Use Case Specifications*/
49 Let  $U$  be the set of use cases selected in  $DC$  containing variability in their
    specifications;
50 Let  $Y$  be the set of essential use cases containing variability in their
    specifications ;
51  $F \leftarrow U \cup Y$ ;
52 Let  $S$  be the empty set for optional steps and alternative flows selected by the
    user;
53 Let  $O$  be the empty set for decided orders;
54 foreach ( $fc \in F$ ) do
55   Let  $OF$  be the set of selected, optional alternative flows, of  $fc$ , containing
    variability;
56   Let  $b$  be the basic flow, of  $fc$ , containing variability;
57   Let  $AF$  be the set of mandatory alternative flows, of  $fc$ , containing
    variability;
58    $AC \leftarrow OF \cup \{b\} \cup \{AF\}$ ;
59   foreach ( $a \in AC$ ) do
60     Let  $MS$  be the set of mandatory steps in  $a$  ;
61     Let  $SS$  be the set of optional steps selected by the user from  $a$ ;
62     Let  $ORD$  be the set of steps that need to be ordered in  $SS \cup MS$ ;
63     if ( $ORD \neq \emptyset$ ) then
64        $O \leftarrow O \cup \text{updateStepOrder}(ORD)$ ;
65     end if
66      $S \leftarrow S \cup SS \cup OF \cup O$ ;
67   end foreach
68 end foreach
69  $PSS \leftarrow \text{generatePSSpecifications}(S, PLD, PLS)$ ;
70 /* Configure PS Domain Model*/
71 Let  $DE$  be the empty set for optional and variant entities selected by the user;
72 Let  $OE$  be the set of selected, optional domain entities in  $PLDM$ ;
73  $DE \leftarrow DE \cup OE$ ;
74 Let  $V$  be the set of variation domain entities in  $PLDM$ ;
75 foreach ( $vc \in V$ ) do
76   Let  $VE$  be the set of selected, variant domain entities in  $vc$ ;
77    $DE \leftarrow DE \cup VE$ ;
78 end foreach
79  $PSDM \leftarrow \text{generatePSDomainModel}(DE, PLDM)$ ;
80 /* Save all the decisions in the decision model and return the generated models*/
81 Let  $DM$  be the decision model for the configuration;
82  $DM \leftarrow \text{saveDecisions}(DC, S, O, DE)$ ;
83  $PSM \leftarrow \{PSD\} \cup \{PSS\} \cup \{PSDM\} \cup \{DM\}$ ;
84 return  $PSM$ ;
```

Alg. 3: checkConflictingVP

Input : Variation point, vp , Set of decisions, DC ,
Decision, d , Product Line use case diagram, PLD

Output: Set of contradicting decisions

```
1 Let  $RES$  be the empty set for contradicting decisions;
2 Let  $SUC$  be the set of selected variant use cases of variation point  $vp$  in decision
   $d$ ; /*  $d.SUC$  in the quadruple */
3 if ( $SUC \neq \emptyset$ ) then
4   Let  $C$  be the empty set for conflicting elements;
5    $C \leftarrow \text{getConflictingElements}(vp, PLD)$ ; /* get set of elements that conflict
      with  $vp$  */
6   if ( $C \neq \emptyset$ ) then
7     foreach ( $dm \in DC$ ) do
8       Let  $p$  be the variation point in decision  $dm$ ; /*  $dm.vp$  in the
          quadruple */
9       Let  $S$  be the set of selected use cases of variation point  $p$  in decision
         $dm$ ; /*  $dm.SUC$  in the quadruple */
10      if ( $S \neq \emptyset$  and ( $p \in C$ )) then
11         $RES \leftarrow RES \cup \{dm\}$ ;
12      end if
13      foreach ( $u \in S$ ) do
14        Let  $I$  be the empty set for inferred, selected elements;
15         $I \leftarrow \{u\} \cup \text{inferSelectedElements}(u, dm, DC, \emptyset, PLD)$ ;
16        foreach ( $e \in I$ ) do
17          if ( $e \in C$ ) then
18             $RES \leftarrow RES \cup \text{getInvolvedDecisions}(e, dm, d, DC)$ ;
19          end if
20        end foreach
21      end foreach
22    end foreach
23  end if
24 end if
25 return  $RES$ ;
```

Alg. 4: checkConflictingUC

Input : Use Case, uc , Set of decisions, DC ,
Decision, d , Product Line use case diagram, PLD

Output: Set of contradicting decisions

```
1 Let  $RES$  be the empty set for contradicting decisions;
2 Let  $I$  be the empty set for inferred, selected elements;
3  $I \leftarrow \{uc\} \cup \text{inferSelectedElements}(uc, d, DC, \emptyset, PLD)$ ;
4 Let  $C$  be the empty set for conflicting elements;
5 foreach ( $e \in I$ ) do
6    $C \leftarrow C \cup \text{getConflictingElements}(e, PLD)$ ; /* get the set of elements
   conflicting with  $e$ */
7 end foreach
8 if ( $C \neq \emptyset$ ) then
9   foreach ( $dm \in DC$ ) do
10    Let  $vp$  be the variation point in decision  $dm$ ; /*  $dm.vp$  in the quadruple*/
11    Let  $SUC$  be the set of selected variant use cases in  $dm$ ; /*  $dm.SUC$  in the
    quadruple*/
12    if ( $(SUC \neq \emptyset) \text{ and } (vp \in C)$ ) then
13       $RES \leftarrow RES \cup \{dm\}$ ;
14    end if
15    foreach ( $u \in SUC$ ) do
16      Let  $B$  be the empty set for inferred, selected elements;
17       $B \leftarrow \{u\} \cup \text{inferSelectedElements}(u, dm, DC, \emptyset, PLD)$ ;
18      foreach ( $e \in B$ ) do
19        if ( $e \in C$ ) then
20           $RES \leftarrow RES \cup \text{getInvolvedDecisions}(e, dm, d, DC)$ ;
21        end if
22      end foreach
23    end foreach
24  end foreach
25 end if
26 return  $RES$ ;
```

Alg. 5: checkRequiringVP

Input : Variation Point, vp , Set of decisions, DC ,
Decision, d , Product Line use case diagram, PLD

Output: Set of contradicting decisions

```
1 Let  $RES$  be the empty set for contradicting decisions;
2 Let  $R$  be the empty set for required elements;
3 Let  $SUC$  be the set of selected variant use cases in  $d$ ; /*  $d.SUC$  in the
   quadruple*/
4 if ( $SUC \neq \emptyset$ ) then
5    $R \leftarrow \text{getRequiredElements}(vp, PLD)$ ;
6   if ( $R \neq \emptyset$ ) then
7     foreach ( $dm \in DC$ ) do
8       Let  $p$  be the variation point in the decision  $dm$ ; /*  $dm.vp$  in the
         quadruple*/
9       if (( $p$  is unselected in  $DC \cup \{d\}$ ) and (there is no more decision to be
         made for  $p$ ) and ( $p \in R$ )) then
10         $RES \leftarrow RES \cup \text{getInvolvedDecisions}(p, dm, d, DC)$ ;
11      end if
12      Let  $NSUC$  be the set of unselected variant use cases in  $dm$ ; /*
         $dm.NSUC$  in the quadruple*/
13      foreach ( $v \in NSUC$ ) do
14        Let  $EX$  be the empty set for inferred, unselected elements;
15         $EX \leftarrow \{v\} \cup \text{inferUnselectedElements}(v, d, DC, \emptyset, PLD)$ ;
16        foreach ( $e \in EX$ ) do
17          if (( $e$  is unselected in  $DC \cup \{d\}$ ) and (there is no more
            decision to be made for  $e$ ) and ( $e \in R$ )) then
18             $RES \leftarrow RES \cup \text{getInvolvedDecisions}(e, dm, d, DC)$ ;
19          end if
20        end foreach
21      end foreach
22    end foreach
23  end if
24 end if
25 return  $RES$ ;
```

Alg. 6: checkRequiredVP

Input : Variation Point, vp , Set of decisions, DC ,
Decision, d , Product Line use case diagram, PLD

Output: Set of contradicting decisions

```
1 Let  $RES$  be the empty set for contradicting decisions;
2 Let  $R$  be the empty set for required elements;
3 if ( $vp$  is unselected in  $DC \cup \{d\}$ ) and (there is no more decision to be made
   for  $vp$ ) then
4   foreach ( $dm \in DC$ ) do
5      $R \leftarrow \text{getRequiredElements}(dm.vp, PLD)$ ;
6     Let  $SUC$  be the set of selected elements in  $dm$ ; /*  $dm.SUC$  in the
       quadruple */
7     if ( $SUC \neq \emptyset$ ) and ( $vp \in R$ ) then
8        $RES \leftarrow RES \cup \{dm\}$ ;
9     end if
10    foreach ( $u \in SUC$ ) do
11      Let  $I$  be the set of inferred, selected elements (variant use cases and
        variation points);
12       $I \leftarrow \{u\} \cup \text{inferSelectedElements}(u, dm, DC \cup \{d\}, \emptyset, PLD)$ ;
13      foreach ( $e \in I$ ) do
14        if ( $vp \in \text{getRequiredElements}(e, PLD)$ ) then
15           $RES \leftarrow RES \cup \text{getInvolvedDecisions}(e, dm, d, DC)$ ;
16        end if
17      end foreach
18    end foreach
19  end foreach
20 end if
21 return  $RES$ ;
```

Alg. 7: checkRequiringUC

Input : Use Case, uc , Set of decisions, DC ,
Decision, d , Product Line use case diagram, PLD

Output: Set of contradicting decisions

```
1 Let  $RES$  be the empty set for contradicting decisions;
2 Let  $I$  be the empty set for inferred, selected elements (variant use cases and
  variation points);
3  $I \leftarrow \{uc\} \cup \text{inferSelectedElements}(uc, d, DC, \emptyset, PLD)$ ;
4 Let  $R$  be the empty set for required elements;
5 foreach ( $e \in I$ ) do
6    $R \leftarrow R \cup \text{getRequiredElements}(e, PLD)$ ; /* get set of elements required
   by  $e$  */
7 end foreach
8 if ( $R \neq \emptyset$ ) then
9   foreach ( $dm \in DC$ ) do
10    Let  $vp$  be the the variation point in  $dm$ ; /*  $dm.vp$  in the quadruple */
11    if ( $(vp$  is unselected in  $DC \cup \{d\})$  and  $($ there is no more decision to be
      made for  $vp)$  and  $(vp \in R)$ ) then
12       $RES \leftarrow RES \cup \text{getInvolvedDecisions}(vp, dm, d, DC)$ ;
13    end if
14    Let  $NSUC$  be the set of unselected elements in  $dm$ ; /*  $dm.NSUC$  in the
      quadruple */
15    foreach ( $n \in NSUC$ ) do
16      if ( $(n$  is not selected in  $DC \cup \{d\})$  and  $($ there is no more decision to
        be made for  $n)$  and  $(n \in R)$ ) then
17         $RES \leftarrow RES \cup \text{getInvolvedDecisions}(n, dm, d, DC)$ ;
18      end if
19    end foreach
20  end foreach
21 end if
22 return  $RES$ ;
```

Alg. 8: checkRequiredUC

Input : Use Case, uc , Set of decisions, DC ,
Decision, d , Product Line use case diagram, PLD

Output: Set of contradicting decisions

```
1 Let  $RES$  be the empty set for contradicting decisions;
2 if ( $(uc$  is not selected in the completed decisions  $DC$ ) and ( $there is no further$ 
    $decision to be made for  $uc$ )) then
3   Let  $EX$  be the empty set for inferred, unselected elements;
4    $EX \leftarrow \{uc\} \cup \text{inferUnselectedElements}(uc, d, DC, \emptyset, PLD)$ ;
5   foreach ( $dm \in DC$ ) do
6     Let  $SUC$  be the set of selected use cases in  $dm$ ;
7     Let  $vp$  be the variation point in  $dm$ ;
8     if ( $SUC \neq \emptyset$ ) and ( $(EX \cap \text{getRequiredElements}(vp, PLD)) \neq \emptyset$ ) then
9        $RES \leftarrow RES \cup \{dm\}$ ;
10    end if
11    foreach ( $u \in SUC$ ) do
12      Let  $I$  be the empty set for inferred, selected elements;
13       $I \leftarrow \{u\} \cup \text{inferSelectedElements}(u, dm, DC \cup \{d\}, \emptyset, PLD)$ ;
14      foreach ( $e \in I$ ) do
15        if ( $EX \cap \text{getRequiredElements}(e, PLD) \neq \emptyset$ ) then
16           $RES \leftarrow RES \cup \text{getInvolvedDecisions}(e, dm, d, DC)$ ;
17        end if
18      end foreach
19    end foreach
20  end foreach
21 end if
22 return  $RES$ ;$ 
```

Alg. 9: inferSelectedElements

Input : Traversed element (Variation Point or Use Case), *elem*,
Set of decisions, *DC*,
Decision, *d*,
Set of previously traversed elements, *PR*,
Product Line use case diagram, *PLD*

Output: Set of inferred, selected elements

```
1 Let RES be the empty set for inferred, selected elements;
2 if (elem  $\notin$  PR) then
3   PR  $\leftarrow$  PR  $\cup$  {elem};
4   if (elem is a Variation Point) then
5     Let SU be the set of selected use cases of elem in {d}  $\cup$  DC;
6     if ((elem is mandatory) or (SU  $\neq$   $\emptyset$ )) then
7       RES  $\leftarrow$  {elem};
8       Let Y be the set of variant use cases of elem with a mandatory
          variability relation;
9       Let T be the set of variant use cases of elem selected in DC  $\cup$  {d};
10      U  $\leftarrow$  Y  $\cup$  T;
11      foreach (uc  $\in$  U) do
12        | RES  $\leftarrow$  RES  $\cup$  inferSelectedElements(uc, DC, d, PR, PLD);
13      end foreach
14    end if
15  else
16    if (elem is a Use Case) then
17      if ((elem is mandatory) or (elem is selected in {d}  $\cup$  DC)) then
18        RES  $\leftarrow$  {elem};
19        Let V be the set of variation points included by elem in PLD;
20        foreach (vp  $\in$  V) do
21          | RES  $\leftarrow$  RES  $\cup$  inferSelectedElements(vp, DC, d, PR,
22            | PLD);
23        end foreach
24      end if
25    end if
26  end if
27 return RES;
```

Alg. 10: inferUnselectedElements

Input : Traversed element (Variation Point or Use Case), *elem*,
Set of decisions, *DC*,
Decision, *d*,
Set of the previously traversed elements, *PR*,
Product Line use case diagram, *PLD*

Output: Set of inferred, unselected elements

```
1 Let RES be the empty set for inferred, unselected elements;
2 if (elem  $\notin$  PR) then
3   PR  $\leftarrow$  PR  $\cup$  {elem};
4   if (elem is a Variation Point) then
5     Let SU be the set of variant use cases of elem selected in {d}  $\cup$  DC;
6     if ((SU =  $\emptyset$ ) and (there is no more decisions to be made for use cases of
7       elem)) then
8       RES  $\leftarrow$  {elem};
9       Let U be the set of the use cases of elem that are unselected in
10        DC  $\cup$  {d};
11        foreach (uc  $\in$  U) do
12          RES  $\leftarrow$  RES  $\cup$  inferUnselectedElements(uc, DC, d, PR,
13            PLD);
14        end foreach
15      end if
16    else
17      if (elem is a Use Case) then
18        if ((elem is unselected in {d}  $\cup$  DC) and (there is no more decisions
19          to be made for elem)) then
20            RES  $\leftarrow$  {elem};
21            Let V be the set of variation points included by elem in PLD;
22            foreach (vp  $\in$  V) do
23              RES  $\leftarrow$  RES  $\cup$  inferUnselectedElements(vp, DC, d, PR,
24                PLD);
25            end foreach
26          end if
27        end if
28      end if
29    end if
30  return RES;
```
