

Consolidated Variant Generation with SERGe

Michaela Meier
Presentation: 29.11.2018

Consolidated Variant Generation with SERGe

old „variant generation“ renamed to
Solitary Variant Generator

new, additional variant generation:
Consolidated Variant Generator

Consolidated Variant Generation with SERGe

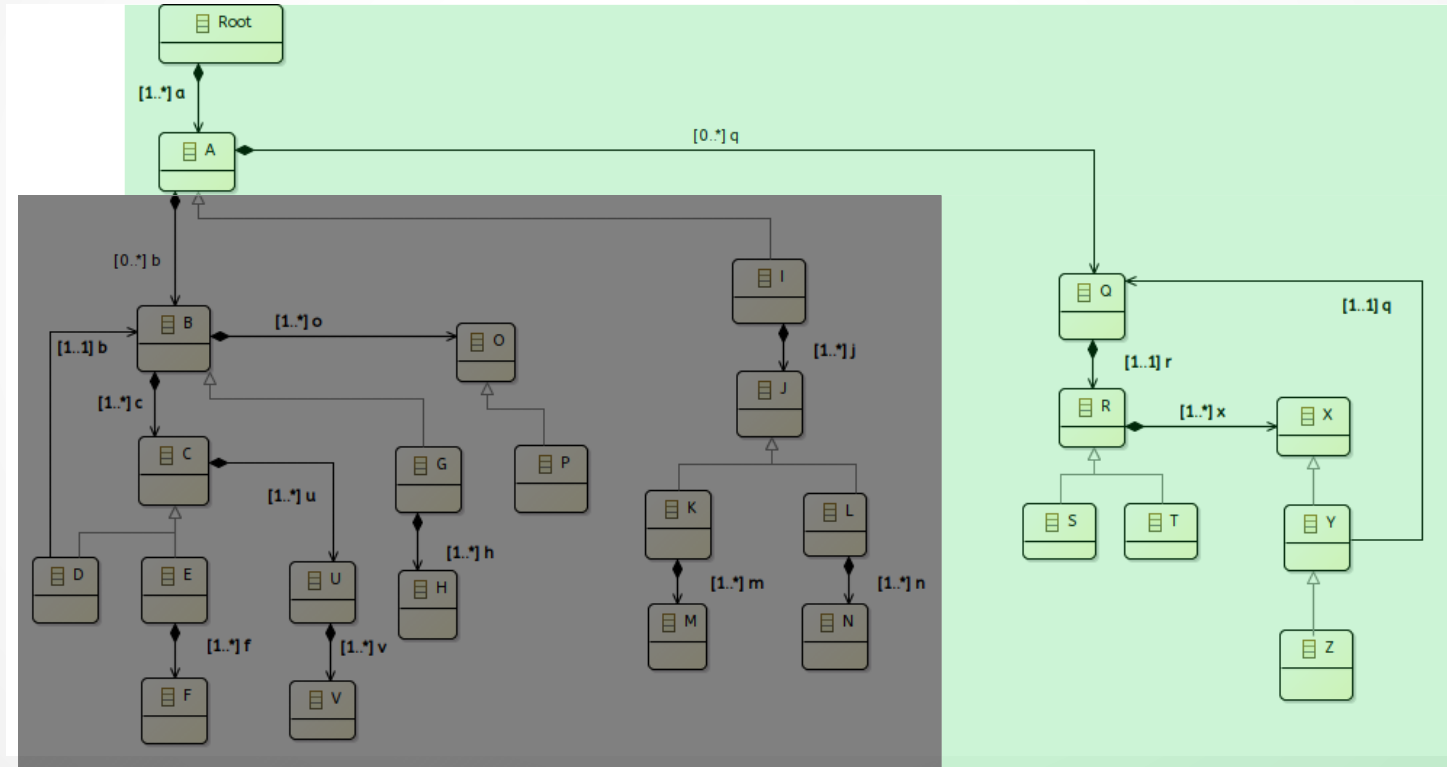
Default: disabled

To enable it:

```
Org.sidiff.editrule.generator.serge > Configuration > enable_consolidatedvariants = true;
```

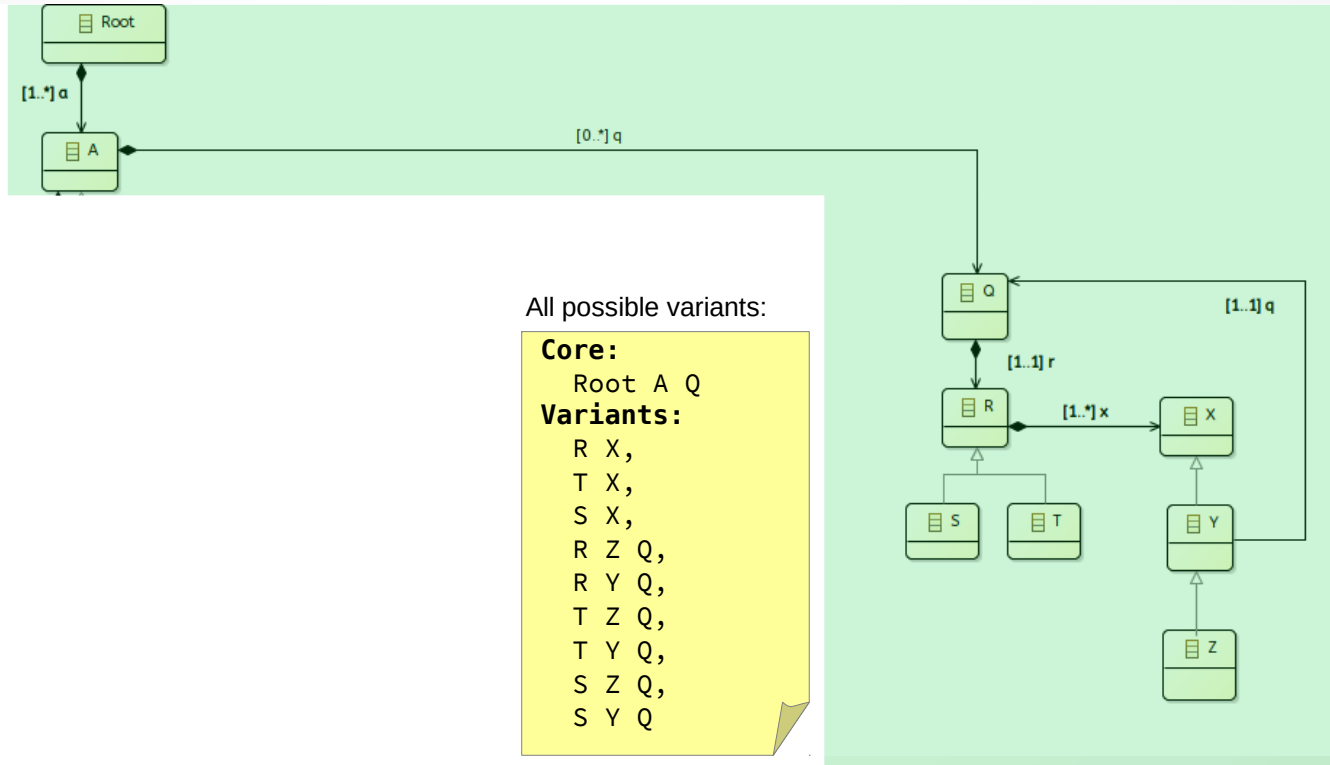
Consolidated Variant Generation with SERGe

Create_Q_In_A(q)



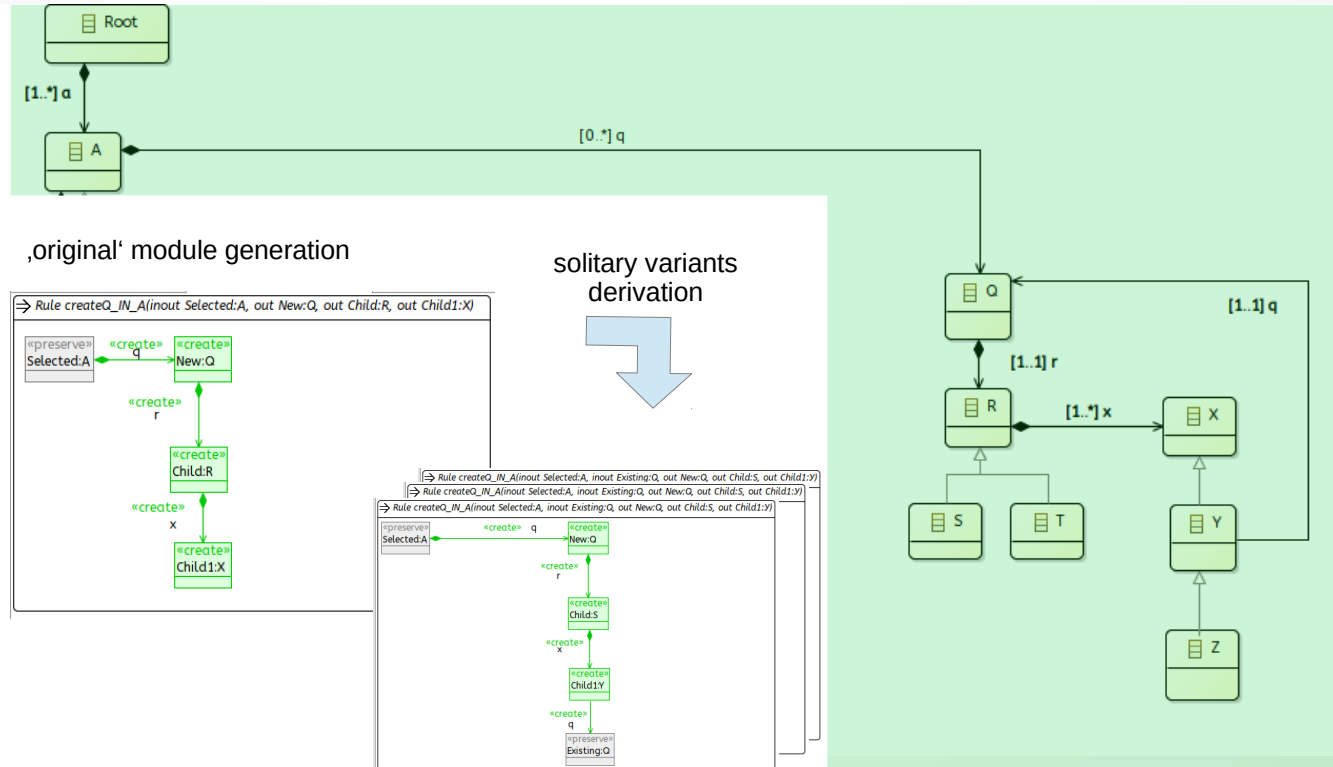
Consolidated Variant Generation with SERGe

Create_Q_In_A(q)



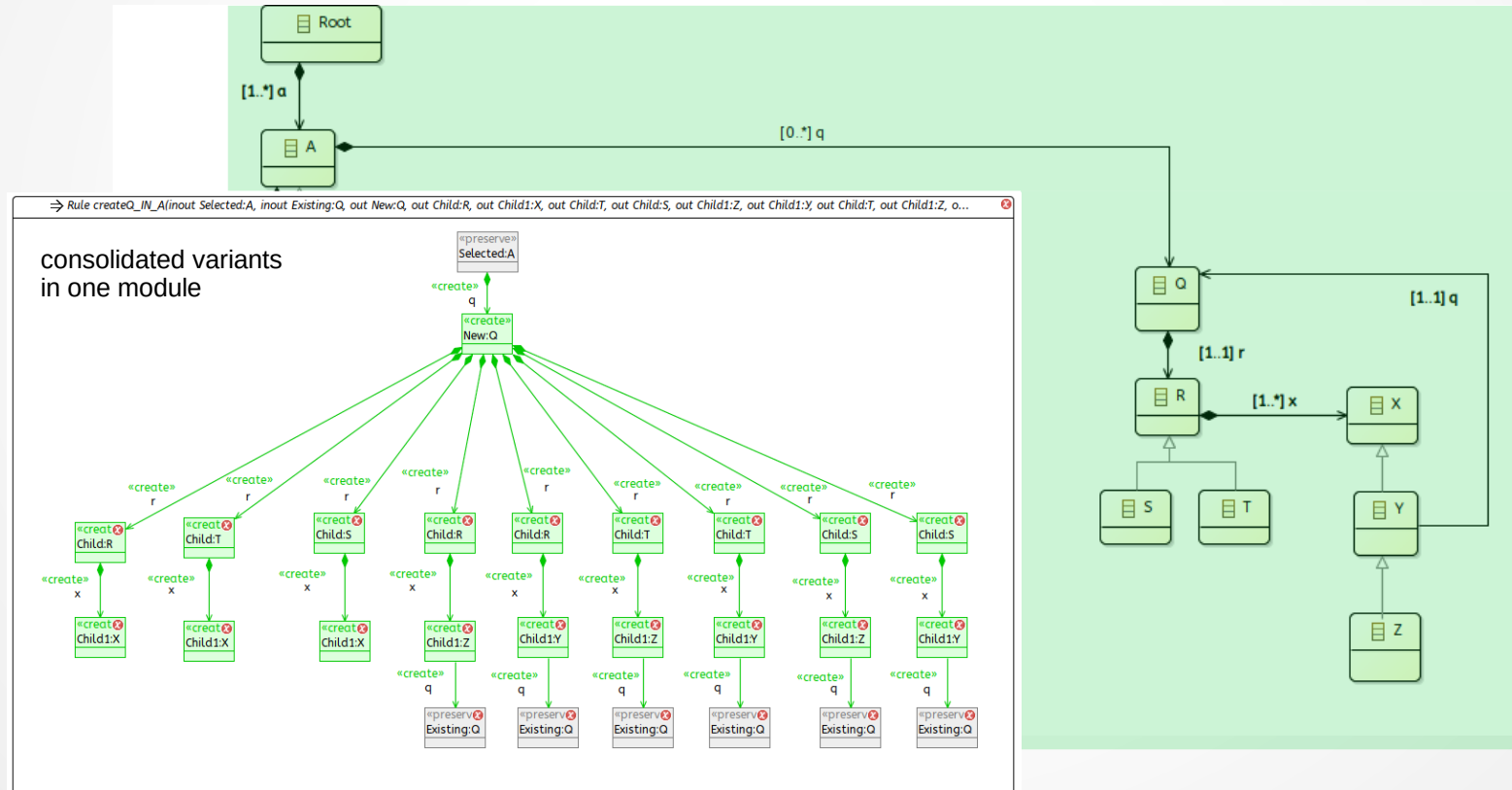
Consolidated Variant Generation with SERGe

Create_Q_In_A(q)



Consolidated Variant Generation with SERGe

Create_Q_In_A(q)



Consolidated Variant Generation with SERGe

Resource Set

- platform:/resource/TEST/rules/CREATE/CREATE_Q_IN_A_(q)_execute.henshin
 - Module CREATE_Q_IN_A_(q)
 - Annotation allVariants
 - ⇒ Rule createQ_IN_A(inout Selected, inout Existing, out New, out Child, out Ch
 - ✖ PriorityUnit mainUnit(selectedEObject, Existing, Existing, Existing, Existing, E

Problems Target Platform State Console Git Staging Call Hierarchy

Property	Value
Key	allVariants
Value	R X,T X,S X,R Z Q,R Y Q,T Z Q,T Y Q,S Z Q,S Y Q

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <henshin:Module xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ec
3 <annotations xmi:id="_EeHiwdhbEei9g9a0YuhJZw" key="allVariants" value="R X,T X,S X,R Z Q,R Y Q,T Z Q,T Y Q,S Z Q,S Y Q"/>
4 <imports href="http://www.sidiff.org/editrulegenerator/test/variantmodel#"/>
5 <units xsi:type="henshin:Rule" xmi:id="_EeHiwthbEei9g9a0YuhJZw" name="createQ_IN_A" description="creates one Q in the context: A"
6 <parameters xmi:id="FeHiw9hhFeiv9a0Yuh17w" name="Selected" kind="INOUT">
```

Consolidated Variant Generation with SERGe

Resource Set

- platform:/resource/TEST/rules/CREATE/CREATE_Q_IN_A_(q)_execute.henshin
 - Module CREATE_Q_IN_A_(q)
 - Annotation allVariants
 - ⇒ Rule createQ_IN_A(*inout Selected, inout Existing, out New, out Child, out Chil*)
 - (...)Parameters
 - L Graph Lhs
 - R Graph Rhs
 - Node Selected:A
 - Node New:Q
 - Node Child:R
 - Node Child1:X
 - Node Child:T
 - Annotation presenceCondition
 - Node Child1:X
 - Node Child:S
 - Node Child1:X
 - Node Child:R
 - Node Child1:Z

Problems Target Platform State Console Git Staging Call Hierarchy

Property	Value
Key	presenceCondition
Value	T X

```
<nodes xmi:id=" EeHi4dhbEei yg9a0YuhJZw" name="Child" incoming=" EeHjCthbEei yg9a0YuhJZw" outgoing="_EeHjC9hbEei yg9a0YuhJZw">  
  <annotations xmi:id=" EeHi4thbEei yg9a0YuhJZw" key="presenceCondition" value="T X"/>  
  <type href="http://www.sidiff.org/editrulegenerator/test/variantmodel#//T"/>  
</nodes>
```

Consolidated Variant Generation with SERGe

Restrictions:

- Only CREATE rules are considered
- rules containing abstract <<create>> elements not dismissed yet
- no element type exclusion implemented (as with everything on the xml configuration)

Consolidated Variant Generation with SERGe

How it works?

Basically:

- 1) Sequences of all variants are calculated for one module
- 2) Nodes/Edges are replicated and extended and recurved for every sequence

Consolidated Variant Generation with SERGe

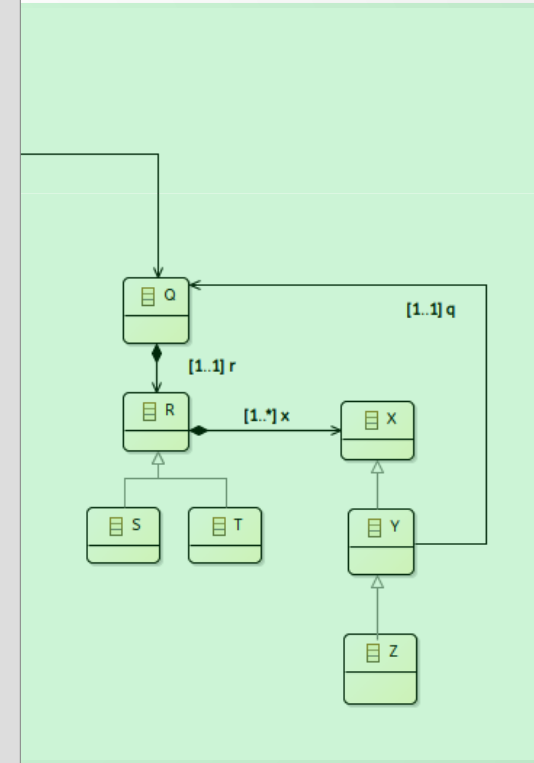
- STEP 1 :: Init sequence:
R X // the core (Root A Q) is further on omitted
- STEP 2 ::
2.1 Consider first „top most, lefthand side“ supertype replacement option
[R] X

2.2 Create additional sequences for all possible sub types by copy
=> R X
 S X
 T X
- STEP 3 ::
3.1 Consider next, „top most, lefthand side“ super type replacement option:
R [X]

3.2 Create additional sequences for all possible sub types by copy
=> R X
 S X
 T X
 R Y {Q} //further sub sequences added also; i.e. {Q}
 R Z {Q} //further sub sequences added also; i.e. {Q}

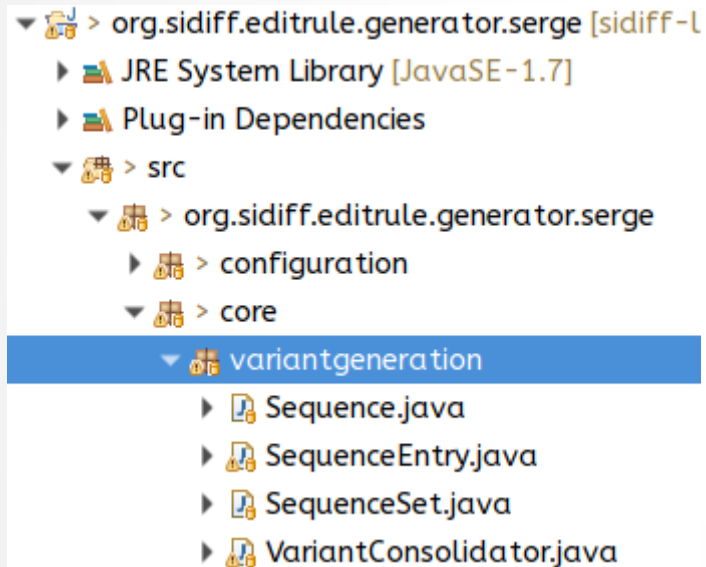
3.3 Create additional sequences for all possible sub types in former sequences
=> R X
 S X
 T X
 R Y {Q}
 R Z {Q}
 S Y {Q}
 S Z {Q}
 T Y {Q}
 T Z {Q}
- STEP 4 :: Consider next unvisited sequence in the previous list, go back to step 2 until no sub type replacement option is left

sequence generation algorithm



Consolidated Variant Generation with SERGe

Algorithmic part can be found in:



Consolidated Variant Generation with SERGe

Default: consolidated variant generation is disabled

To enable it:

```
Org.sidiff.editrule.generator.serge > Configuration > set enable_consolidatedvariants = true;
```

Everything is merged into master branch by now