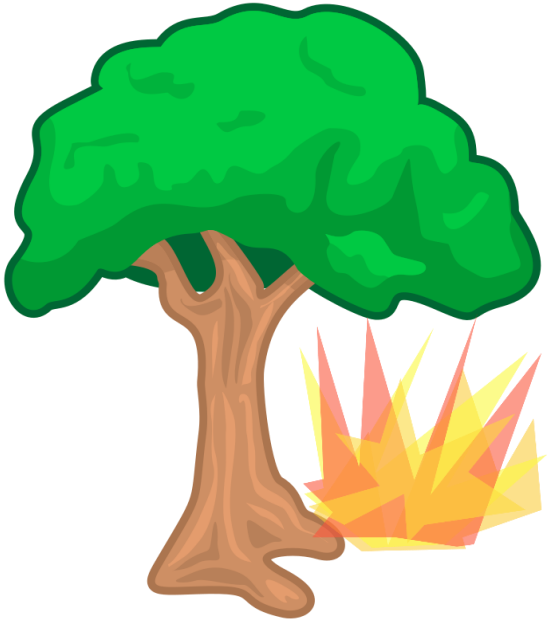
Technische Universität Braunschweig

Context-Sensitive Software Product Lines

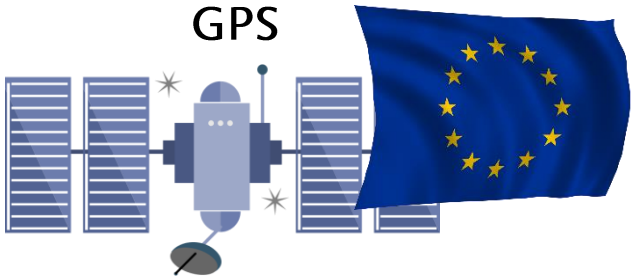
Michael Nieke

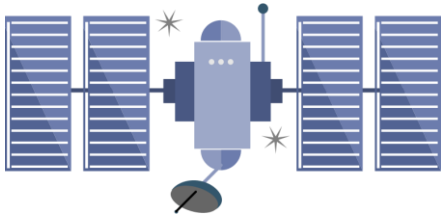
September 1, 2017

Running Example
In-Car Emergency Call Systems



Running Example
In-Car Emergency Call Systems – Different Countries



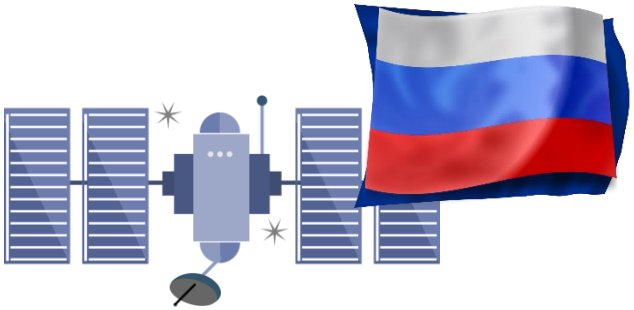


eCall

ERA-Glonass

→ Spatial variability

Running Example
In-Car Emergency Call Systems – Context-Awareness



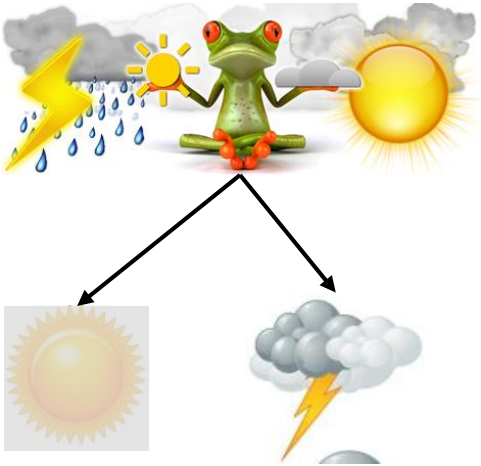
eCall

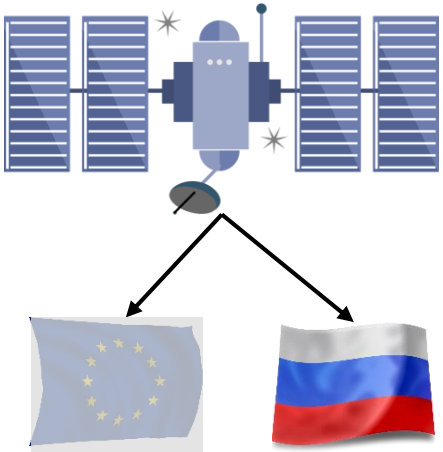
Glonass

→ Contextual variability

Context-Sensitive SPLs
Capturing Context

- Context of systems needs to be captured
- Context may consist of various *context information*
- 3 types of context information with domains:
  - Boolean
  - Integer
  - Enum
  - (String)





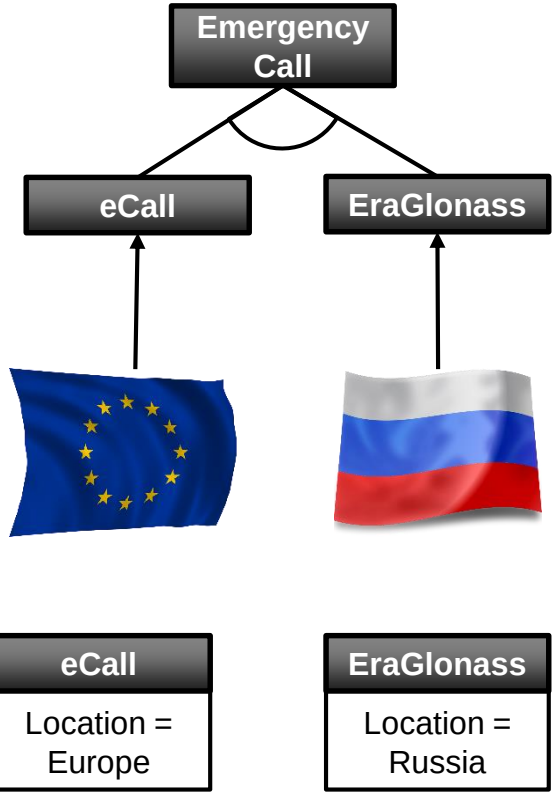
- A context is defined by values for context information

Context-Sensitive SPLs
Context Sensitivity

- Systems need to adapt based on the context
  - For SPLs, this means feature selection needs to change

Multiple approaches:

1. Validity formulas (VF):
   - Enrich features with propositional formulas
   - A feature is selectable only if the VF evaluates to *true*
   - VF can relate context, attributes, and features



Context-Sensitive SPLs
Context Sensitivity

- Systems need to adapt based on the context
  - For SPLs, this means feature selection needs to change

Multiple approaches:

2. Contextual cross-tree constraints:
   - Use context information in CTCs
   - Can require a feature to be seleted in a certain context

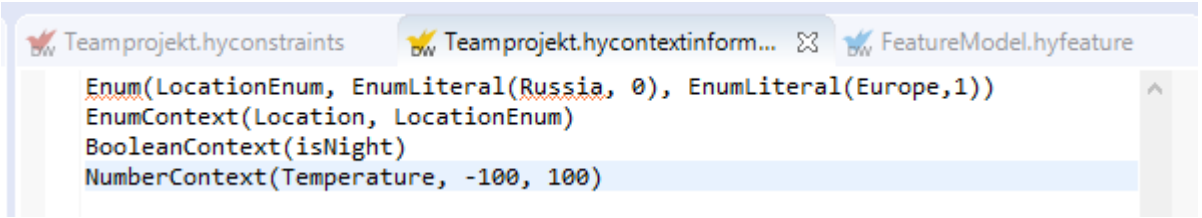
Location = Europe → eCall
Location = Russia → EraGlonass

Context-Sensitive SPLs
DarwinSPL – Context Information

Context information syntax:

1. Define enums and values: Enum(<EnumName>, EnumLiteral(<LiteralName>, <value>), EnumLiteral(<LiteralName>, <value>), …)

2. Define context information:
   - Boolean: BooleanContext(<Name>)
   - Enum: EnumContext(<Name>, <EnumName>)
   - Integer: NumberContext(<Name>, <min>, <max>)



Context-Sensitive SPLs
DarwinSPL –Contextual Constraints & Validity Formulas

Contextual Constraints:

- Integrated in „normal“ constraint editor
- Contextual information identified using „context:“ keyword
- Enums identified using „enum:“ keyword → enum literal value: „enum:<enumName>.<literalName>

Teamprojekt.hyconstraints | Teamprojekt.hycontextinform... | FeatureModel.hyfeature

```
EraGlonass -> Glonass
eCall -> Positioning
context:Location = enum:LocationEnum.Europe -> eCall
context:Location = enum:LocationEnum.Russia -> EraGlonass
```

Syntax Validity Formulas:

- <FeatureName> : <Validity Formula>
- The validity formula itself is written as constraints

Teamprojekt.hy... | Teamprojekt.hy... | FeatureModel.hy... | Teamprojekt.hy...

```
eCall: context:Location = enum:LocationEnum.Europe
EraGlonass: context:Location = enum:LocationEnum.Russia
```

Demo

DarwinSPL

https://github.com/HyVar/DarwinSPL

Task

- Create contextual information „Location“ with „USA“, „Europe“ and „else“ as values
- In the USA, only „SimpleEncryption“ is allowed to be used, as the CIA wouldn‘t be able to crack the „StrongEncryption“
- In Europe, the „StrongEncryption“ is mandatory
- In the rest of the world, no specific encryption type is required