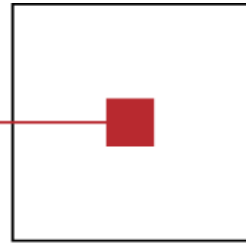




s c c h

software competence center
hagenberg



Improving the Understandability of Formal Specifications

An Experience Report

**Felix Kossak, Atif Mashkooor,
Verena Geist, Christa Illibauer**

+43 7236 3343 811
felix.kossak@scch.at
www.scch.at

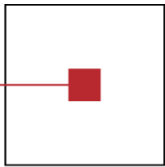
SCCH is an initiative of



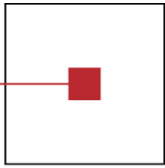
JOHANNES KEPLER
UNIVERSITY LINZ | JKU

SCCH is located in

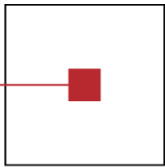
softwarepark 
hagenberg



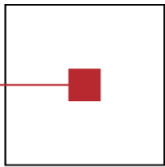
- Motivation
- Specific Suggestions
- Conclusion



- Rigorous methods improve software quality
- But they are hard to sell
 - Stakeholders of a specification: domain experts, managers, lawyers, developers, ...
 - A specification is part of a contract
 - The first project phase is slower
 - It takes longer until something “can be seen”
- Key Problems:
 - Notation
 - Style

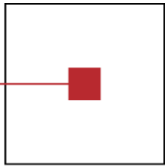


- Notation: Abstract State Machines
- Style:
 - Top-down
 - Identifiers
 - Bracketing
 - Keywords
 - Structure of expressions
 - Set expressions
- Flexibility

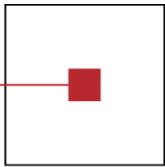


```
rule ConsumeOneToken(in, i) =  
  choose t in toksInSFForInst(in, i) do  
    remove t from toksInSF(in)
```

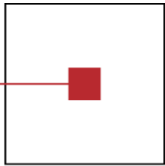
```
rule ConsumeOneToken(incomingSequenceFlow,  
  instance) =  
  choose token in tokensInSequenceFlowForInstance(  
    incomingSequenceFlow, instance) do  
    remove token from tokensInSequenceFlow(  
      incomingSequenceFlow)
```



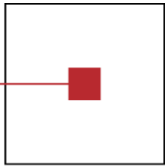
```
rule FlowNodeBehaviour(flowNode) =  
  if eventCondition(flowNode)  
    and controlCondition(flowNode)  
    and dataCondition(flowNode)  
    and resourceCondition(flowNode) then  
  DataOperation(flowNode)  
  ControlOperation(flowNode)  
  EventOperation(flowNode)  
  ResourceOperation(flowNode)
```



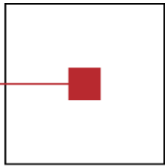
```
rule FlowNodeBehaviour(flowNode) =  
  if eventCondition(flowNode)  
    and controlCondition(flowNode)  
    and dataCondition(flowNode)  
    and resourceCondition(flowNode) then  
  parallelblock  
    DataOperation(flowNode)  
    ControlOperation(flowNode)  
    EventOperation(flowNode)  
    ResourceOperation(flowNode)  
  endparallelblock
```



```
rule FlowNodeBehaviour(flowNode) =  
  if eventCondition(flowNode)  
    and controlCondition(flowNode)  
    and dataCondition(flowNode)  
    and resourceCondition(flowNode) then  
  do in parallel  
    DataOperation(flowNode)  
    ControlOperation(flowNode)  
    EventOperation(flowNode)  
    ResourceOperation(flowNode)
```

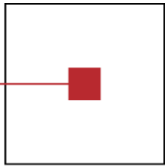
- **forall** token **do** instanceOfToken(token) := instance
forall token instanceOfToken(token) = instance
- **foreach** token **do** ...
foreach token **holds** ...
- **forsome** token **holds** ...
- **do** completionQuantity(flowNode) **times**
ProduceToken(...)



$\{ \text{node} \mid \text{node} \in \text{eventGateTargetNodes}(\dots) \text{ and } \dots \}$

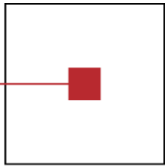
**The set containing each node for which holds
node is in eventGateTargetNodes(...) and ...**

?



```
{ instanceOfToken(token) |  
  forsome sequenceFlow ε incomingSequenceFlows(  
    flowNode) holds  
  token ε sequenceFlow }
```

```
{ instance |  
  forsome token holds  
  forsome sequenceFlow ε incomingSequenceFlows(  
    flowNode) holds  
  token ε sequenceFlow and  
  instanceOfToken(token) = instance  
}
```



- Rigorous methods can and should be used in all kinds of software projects
- Rigorous methods can and should be made more generally understandable
- Rigorous methods can be introduced “gently”

