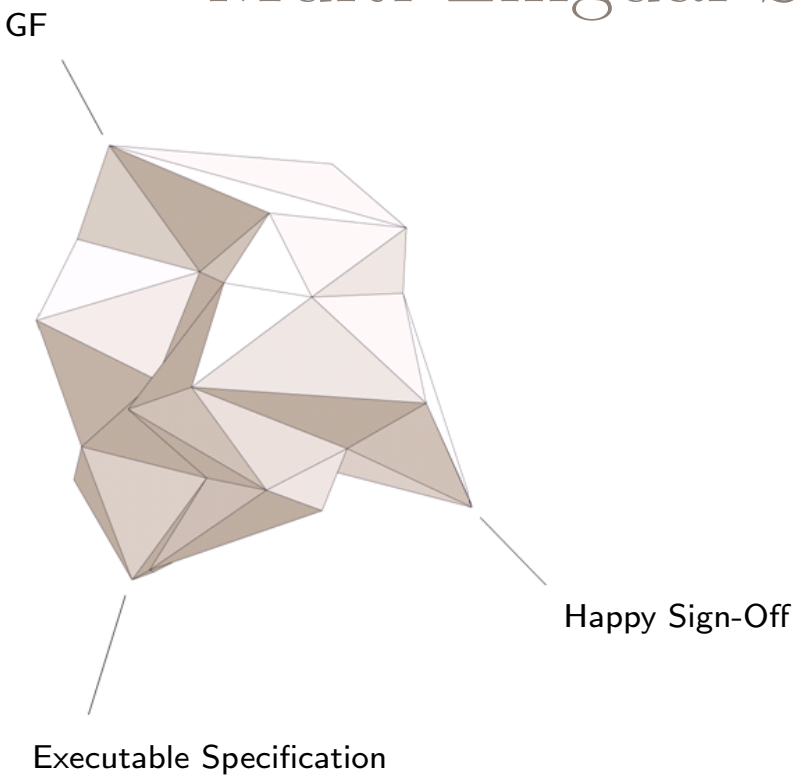
Multi-Lingual Software Specifications

Angela Wallenburg

17 August 2017



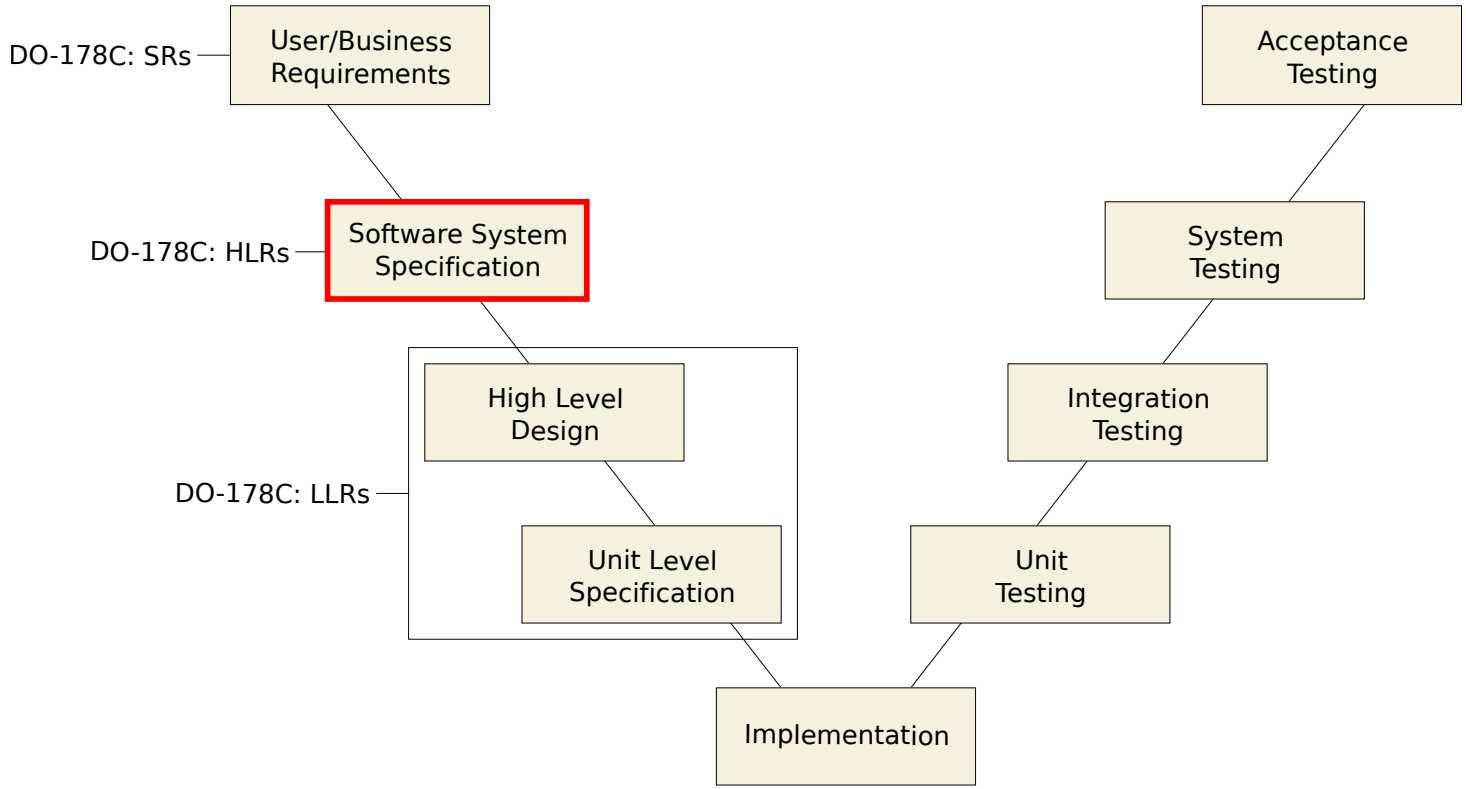
Our World - Critical Software

No bugs please!

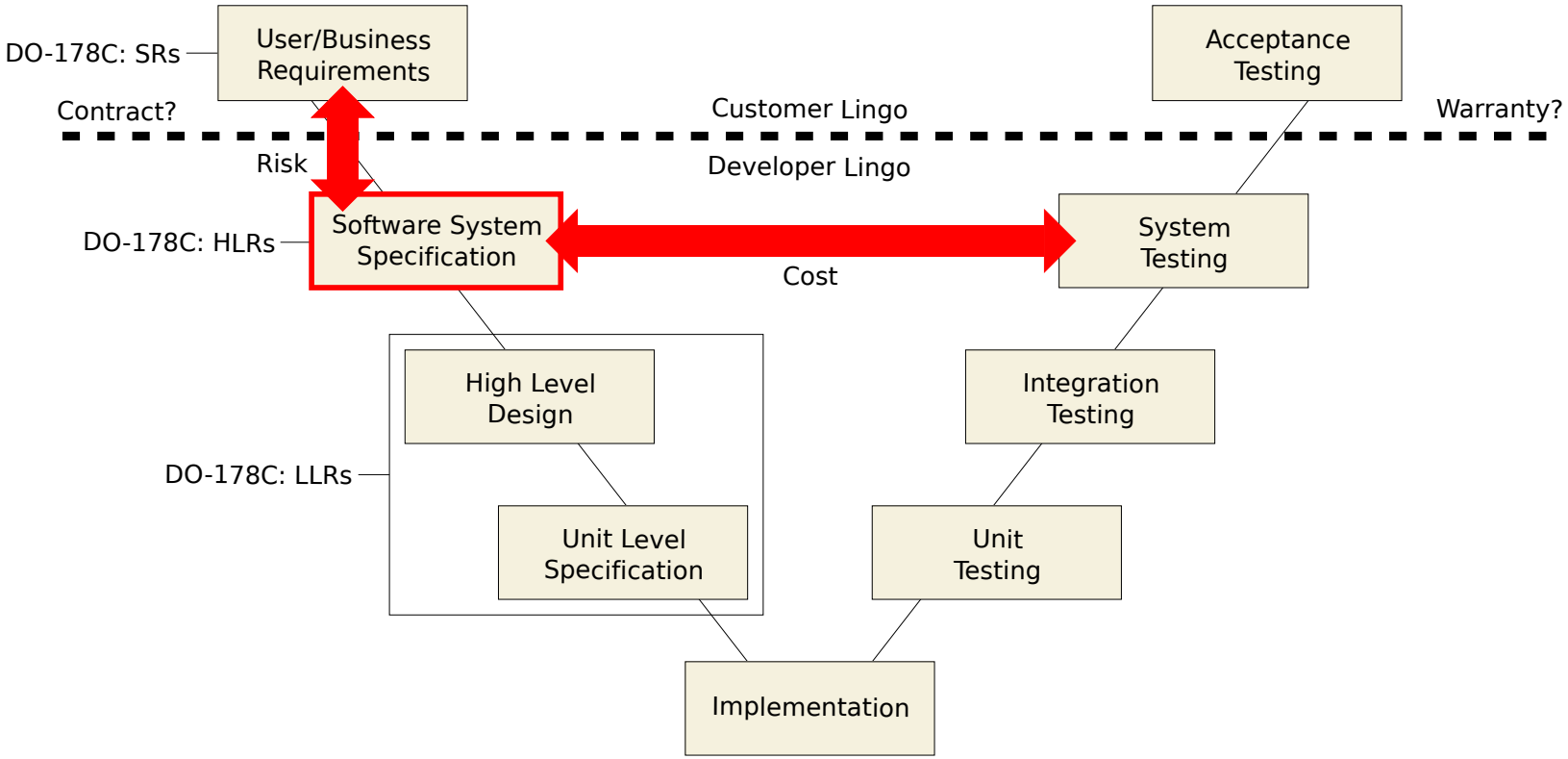
SECT-AIR Goal

"To deliver a step-change improvement in the *affordability* of aerospace software. This is required to secure and develop the UK as a world leader in critical and complex systems development and enable UK aerospace to build new products."

Specification Improvement Motivation

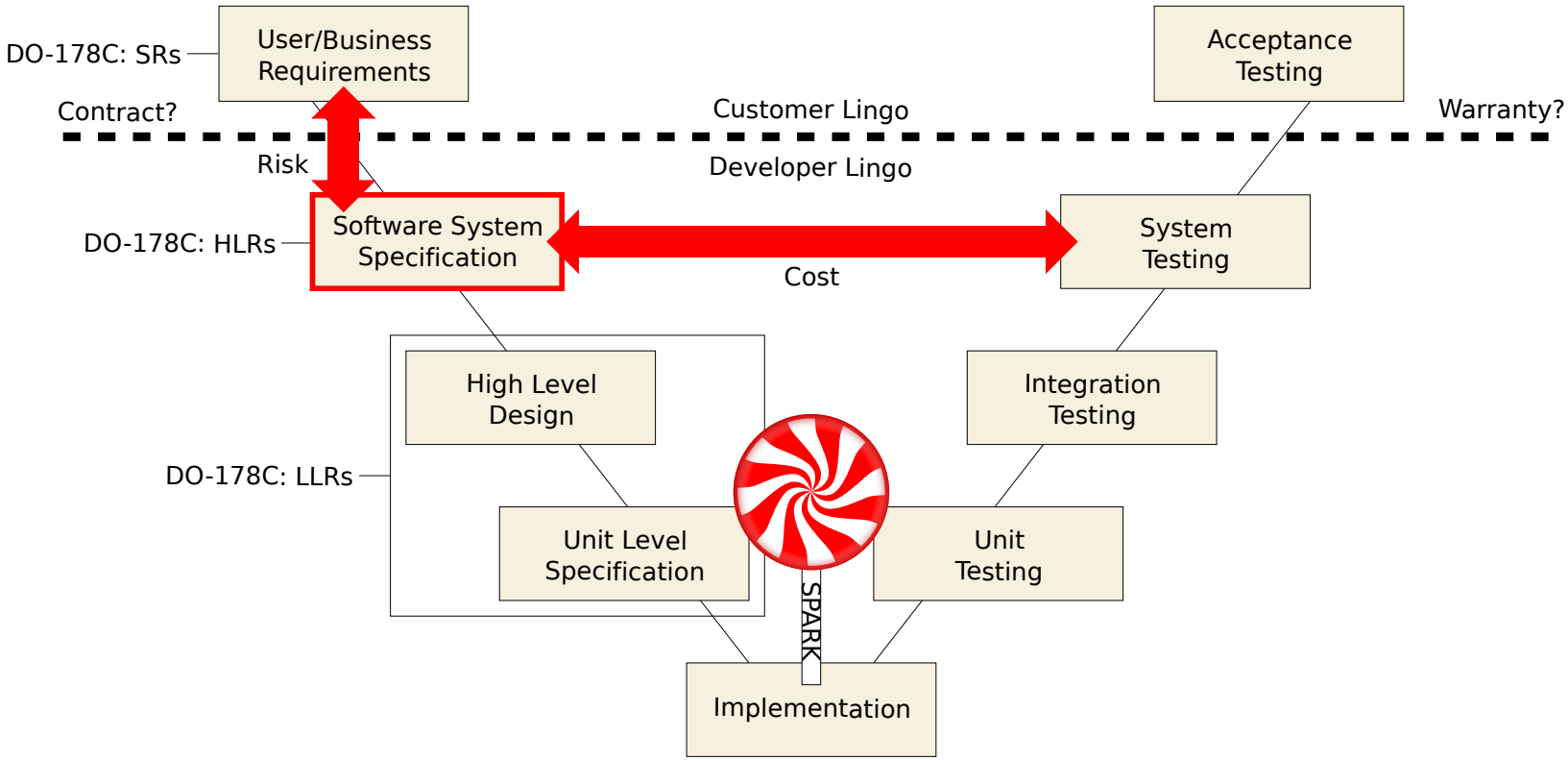


Specification Improvement Motivation



ALTRAN

Specification Improvement Motivation

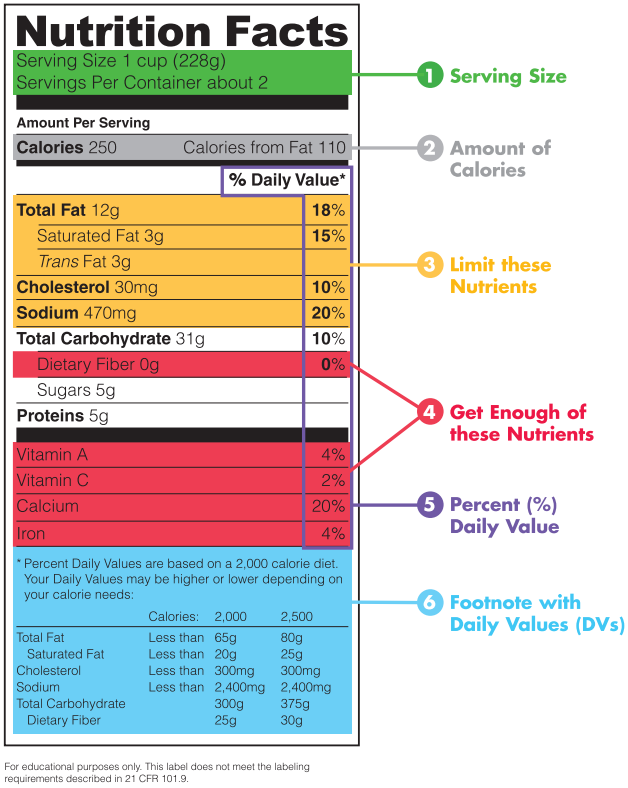


http://www.spark-2014.org/

Which Specification Technology to Choose?

Wait... let us consider semantics first!

Ontology – *Classes* of Options



Focusing on *semantics*.

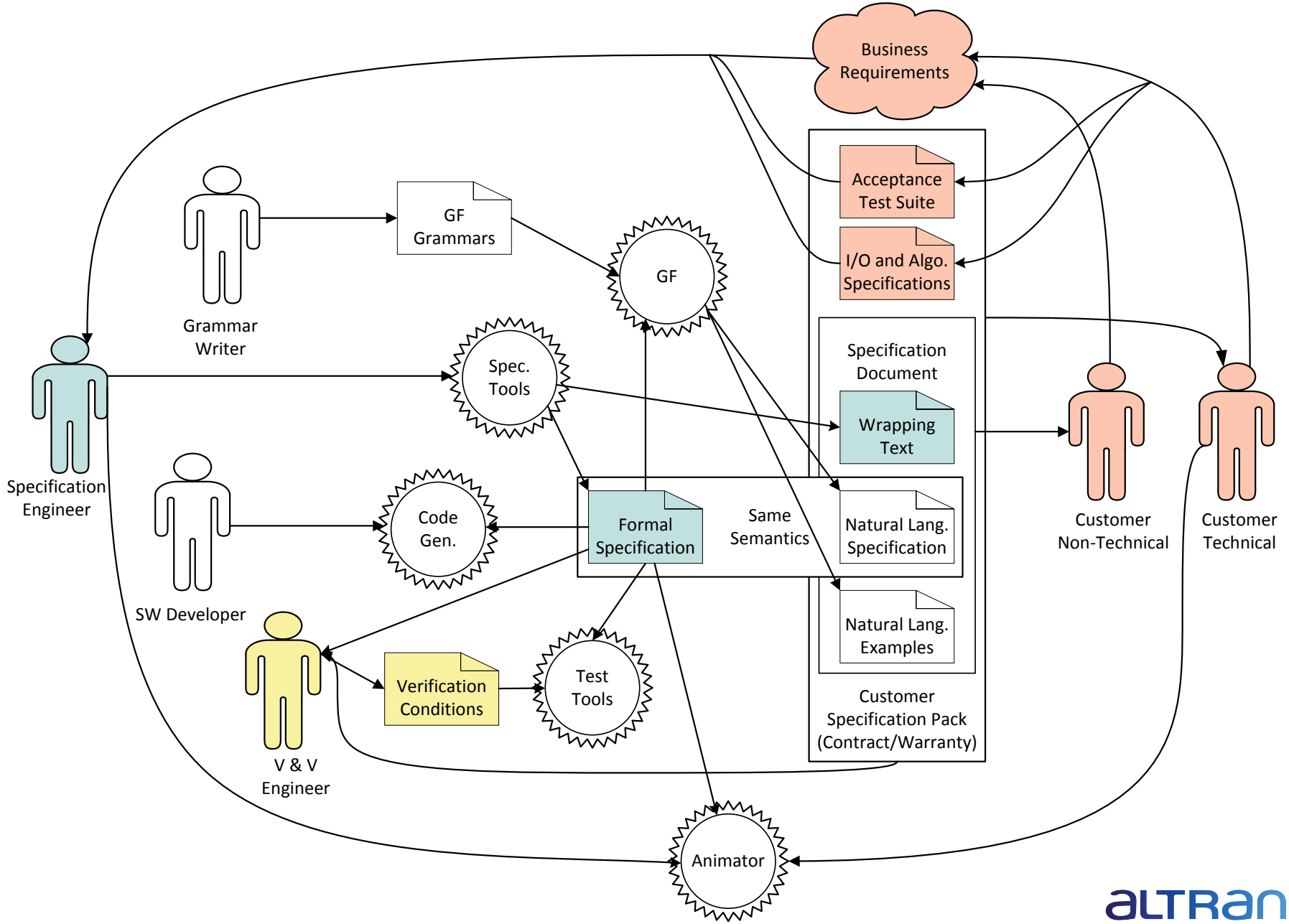
- Domain
- Expressiveness
- Executable or not
- Non-determinism
  - Ex: throttle is in range $t_{min} \ldots t_{max}$
- Abstraction mechanisms
- Validation possibilities
- Tool processing possibilities

Classes of specification languages:

- Reactive control
- Set theory based
- Behavioural interface specification

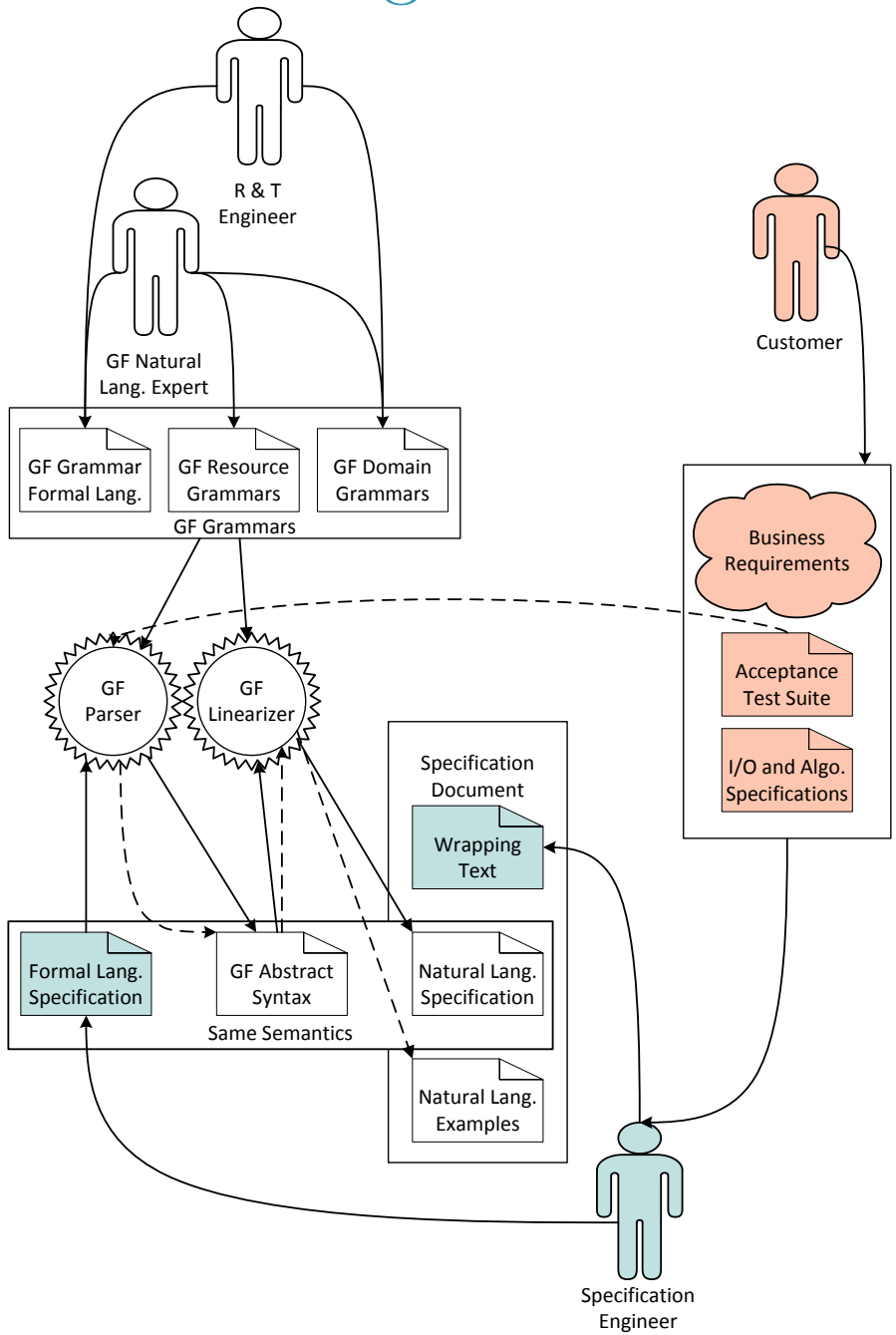
ALTRAN

Specification Solution Design



ALTRAN

Specification Solution Design - GF



ALTRAN

13 Years Ago in Another Verification Tool Building Team...

An example OCL specification for verification of a JavaCard program in the KeY proof system (Johannison 2005):

```
context OwnerPIN
def: let tryCounter = self.triesLeft->at(1)

context OwnerPIN::check(pin: Sequence(Integer),
         offset: Integer, length: Integer): Boolean
post: self.tryCounter = 0 implies result = false
post: (self.tryCounter > 0 and pin <> null and offset >= 0 and length >= 0
         and offset+length <= pin->size()
         and Util.arrayCompare(self.pin, 0, pin, offset, length) = 0
      ) implies (result = true and self.isValidated() and tryCounter = maxTries)
post: (self.tryCounter > 0 and not (pin <> null and offset >= 0 and length >= 0
            and offset+length <= pin->size()
            and Util.arrayCompare(self.pin, 0, pin, offset, length) = 0)
      ) implies (not self.isValidated() and self.tryCounter = tryCounter@pre-1 and
          (( not excThrown(java::lang::Exception) and result = false)
             or excThrown(java::lang::NullPointerException)
             or excThrown(java::lang::ArrayIndexOutOfBoundsException)))
```

Fig. 1. OCL specification from the Java Card API

First Translation Attempt

Translation to English of the OCL specification (Johannison 2005):

In Fig. 2 we show the translation of the OCL specification produced by an earlier version of our system. The English text is basically correct, but it is clumsy and very hard to read.

for the class OwnerPIN introduce the following definition : the tryCounter is defined as the element at index 1 of the triesLeft of the ownerPIN for the operation check (pin : Seq(Integer) , offset : Integer , length : Integer) : Boolean of the class javacard::framework::OwnerPIN the following holds : the following postconditions should hold : (*) if the tryCounter of the ownerPIN is equal to 0 , the result is equal to false (*) if the tryCounter of the ownerPIN is greater than 0 and pin is not equal to null and offset is at least 0 and length is at least 0 and offset plus length is at most the size of pin and the query arrayCompare (the pin of the ownerPIN , 0 , pin , offset , length) to Util is equal to 0 , the result is equal to true and the query isValidated ( ) holds for the ownerPIN and the tryCounter of the ownerPIN is equal to the maxTries of the ownerPIN (*) if the tryCounter of the ownerPIN is greater than 0 and it is not the case that pin is not equal to null and offset is at least 0 and length is at least 0 and offset plus length is at most the size of pin and the query arrayCompare (the pin of the ownerPIN , 0 , pin , offset , length) to Util is equal to 0 , it is not the case that the query isValidated ( ) holds for the ownerPIN and the tryCounter of the ownerPIN is equal to the tryCounter of the ownerPIN at the beginning of the Operation minus 1 and it is not the case that an exception is thrown and the result is equal to false or a nullPointerException is thrown or an arrayIndexOutofBoundsException is thrown

Fig. 2. Translation of OCL specification (before)

Improved Translation

for the class **OwnerPIN** introduce the following definition :

- the try counter is defined as the element at index 1 of the `triesLeft` attribute

for the operation **check (pin : Sequence(Integer) , offset : Integer , length : Integer) : Boolean** of the class **javacard::framework::OwnerPIN** ,
the following post-conditions should hold :

- if the try counter is equal to 0 then this implies that the result is equal to false
- if the following conditions are true
  - the try counter is greater than 0
  - *pin* is not equal to null
  - *offset* is at least 0
  - *length* is at least 0
  - *offset* plus *length* is at most the size of *pin*
  - the query `arrayCompare ( the pin , 0 , pin , offset , length )`[1] on `Util` is equal to 0

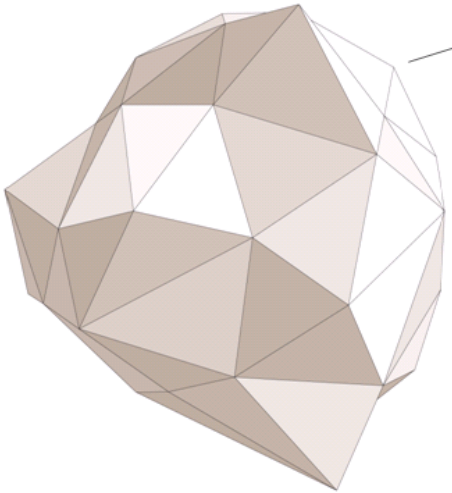
  then this implies that the following conditions are true
  - the result is equal to true
  - this owner PIN is validated
  - the try counter is equal to the maximum number of tries
- if the try counter is greater than 0 and at least one of the following conditions is not true
  - *pin* is not equal to null
  - *offset* is at least 0
  - *length* is at least 0
  - *offset* plus *length* is at most the size of *pin*
  - the query `arrayCompare ( the pin , 0 , pin , offset , length )`[2] on `Util` is equal to 0

  then this implies that the following conditions are true
  - this owner PIN is not validated
  - the try counter is equal to the previous value of the try counter minus 1
  - at least one of the following conditions is true
    - an exception is not thrown and the result is equal to false
    - a null pointer exception is thrown
    - an array index out of bounds exception is thrown

Work in Progress

- Demonstrator of specification framework on existing piece of Z spec. including V&V test monitor running and GF grammars:
  1. Abstract grammar covering non-trivial Z example
  2. Concrete grammar for "Fuzzlisp"
  3. Concrete grammar for zed style LaTeX
  4. Concrete grammar for "SpecSPARK"
  5. Concrete grammar for English (jointly with Digital Grammars)
- Scaling up.

INNOVATION MAKERS



ALTRAN