An approach to unit testing

Rod Furey – Statistics Netherlands
Requested at the IBUC in Den Haag

- "Wouldn't it be nice if it did unit testing as well?"
- "It" in this case is the route checker POC program.
Background

– The route checker POC works as follows:
  - read the compiled datamodel (.bmx)
  - extract the RoutePaths
  - display the fields of the route paths on the screen
  - extract the fields
  - display the fields in a grid
  - allow values to be specified in the grid
  - assign the values to a datarecord
  - execute the rules on the datarecord
  - march the route and draw the route on the screen
Approach to Unit Testing of Blocks

– The compiled datamodel (.bmix) also contains the statements of the datamodel
– By identifying a block, it is possible to extract the contents of the block from the .bmix
– For the Unit Testing POC, a block was extracted and a datamodel was constructed around it which could then be edited, compiled, tested etc.
Source and generated

```
BLOCK bAgeRelated

FIELDS
  Junior : (Yes, No)
  Adult : (Yes, No)
  Senior : (Yes, No)
  Lollipop "Would you like a Lollipop?"
    : (Yes,No)

DOB : datatype
Age : integer

RULES
DOB
  Age := Year(SYSDATE) - Year(DOB)
  IF Age < 17 THEN
    Junior := Yes
    Adult := No
    Senior := No
    Lollipop
  ELSEIF Age > 65 THEN
    Junior := No
    Adult := No
    Senior := Yes
  ELSE
    Junior := No
    Adult := Yes
    Senior := No
  ENDFIELD
ENDBLOCK
```

```
Datamodel Athletics

Fields
  Junior : (Yes, No)
  Adult : (Yes, No)
  Senior : (Yes, No)
  Lollipop "Would you like a Lollipop?" : (Yes, No)

DOB : DATETYPE
Age : INTEGER

Rules

DOB.Ask
  Age := Year(SysDate) - Year(DOB)
  if Age < 17 then
    Junior := 1
    Adult := 2
    Senior := 2
    Lollipop.Ask
  elseif Age > 65 then
    Junior := 2
    Adult := 2
    Senior := 1
  else
    Junior := 2
    Adult := 1
    Senior := 2
  endif
Endmodel
```
Where's the problem?

- The problem is that the IF statement checks if the age is under 17
- In the POC's Use Case, this age should be 18
- In the POC it is possible to edit the generated datamodel to:
  - correct the IF statement
  - save the generated datamodel
  - compile it
  - feed it into the route checker whereupon it can be tested with an appropriate test set
Automated testing

- It is possible to automate the extraction of the blocks in a datamodel
- Once all blocks have been extrapolated to their own unit tests, test data can be generated to exercise these unit tests
- The route can be saved after each test run and compared against expected results for compliance or non-compliance as the case may be
- (Generating the test data is beyond the scope of this paper)