

# Counterpoint: Towards a Proof-Support Tool for VDM

Ken Pierce

Newcastle University, UK

# Motivation (1)

```
from a: AbWorld
1   x: ConWorld                               con-assign(h1)
2   x.conauth: PurseId-set                    conauth-form(1)
3   x.conpurses: PurseId  $\xrightarrow{m}$  ConPurse    conpurses-form(1)
4   dom x.conpurses: PurseId-set              dom-form(3)
5   from name: PurseId; name  $\in$  dom x.conpurses
5.1  x.conpurses(name): ConPurse              at-form(5.h1, 3, 5.h2)
5.2  x.conpurses(name).bal:  $\mathbb{N}$               ConPurse-bal-form(5.1)
5.3  x.conpurses(name).exlog: TD-set          ConPurse-exlog-form(5.1)
5.4  sumval(x.conpurses(name).exlog):  $\mathbb{N}$       sumval-form(5.3)
infer mk-AbPurse(x.conpurses(name).bal,
                sumval(x.conpurses(name).exlog)): AbPurse    mk-AbPurse-form(5.2, 5.4)
6   {name  $\mapsto$ 
    mk-AbPurse(x.conpurses(name).bal, sumval(x.conpurses(name).exlog)) |
    name  $\in$  dom x.conpurses}: PurseId  $\xrightarrow{m}$  AbPurse    map-comp-form-left-set(4, 5)
7   inv-ConWorld(x.conauth, x.conpurses)      inv-ConWorld-I(1)
8   dom x.conpurses  $\subseteq$  x.conauth          unfolding(7)
9   dom {name  $\mapsto$ 
    mk-AbPurse(x.conpurses(name).bal, sumval(x.conpurses(name).exlog)) |
    name  $\in$  dom x.conpurses} = dom x.conpurses    dom-defin-map-comp-left-set(4, 5)
10  dom {name  $\mapsto$ 
    mk-AbPurse(x.conpurses(name).bal, sumval(x.conpurses(name).exlog)) |
    name  $\in$  dom x.conpurses}: PurseId          dom-form(6)
```

# Motivation (2)

```

from a: AbWorld \begin{proof}
1   x: ConWorld \From a: AbWorld \\\
2   x.conauth : \setof{PurseId} \by conauth-form(1)\\\
3   x.conpurses : \mapof{PurseId}{ConPurse} \by conpurses-form(1)\\\
4   dom x : \setof{PurseId} \by dom-form(3)\\\
5   \From name:PurseId; name \in \dom x.conpurses\
5.1 \& x.conpurses(name) : ConPurse \by at-form(5.h1, 3, 5.h2)\\\
5.2 \& x.conpurses(name).bal : \Nat \by ConPurse-bal-form(5.1)\\\
5.3 \& x.conpurses(name).exlog : \setof{ID} \by ConPurse-exlog-form(5.1)\\\
5.4 \& \sumval(x.conpurses(name).exlog):\Nat \by sumval-form(5.3)\\\
   \Infer mk-AbPurse(x.conpurses(name).bal, \
   \& \quad \quad \quad \sumval(x.conpurses(name).exlog)):AbPurse \by
infer r   mk-AbPurse-form(5.2, 5.4)\\\
6   \set{name \mapsto \
   \& \quad mk-AbPurse(x.conpurses(name).bal,
   \quad \sumval(x.conpurses(name).exlog)) | \
   \& \quad name \in \dom x.conpurses} : \mapof{PurseId}{AbPurse} \by
   map-comp-form-left-set(4, 5)\\\
7   \& \inv-ConWorld(x.conauth, x.conpurses) \by \inv-ConWorld-I(1)\\\
8   \& \dom x.conpurses \subseq x.conauth \by unfolding(7)\\\
9   \& \dom \set{name \mapsto \
   \& \quad mk-AbPurse(x.conpurses(name).bal,
   \quad \sumval(x.conpurses(name).exlog)) | \
   \& \quad name \in \dom x.conpurses} = \dom x.conpurses \by
   dom-defn-map-comp-left-set(4, 5)\\\
10  \& \dom \set{name \mapsto \
   \& \quad mk-AbPurse(x.conpurses(name).bal,
   \quad \sumval(x.conpurses(name).exlog)) | \
   \& \quad name \in \dom x.conpurses} : PurseId \by dom-form(6)\\\

```

# Counterpoint

- Overture extension
  - *Proof* perspective
- Proof obligation manager
  - on top of the current PO generator
  - incl. refinement / reification proofs
    - manage multiple specifications and relations
- Proof artifacts
- Visual mnemonics

# Proof Artifacts: Automated Proof

- Supported by plug-ins for external provers
- Define extension points
  - for look & feel, AST access
- Feedback in VDM syntax preferable
- Brute force
  - plus automated retry upon specification change

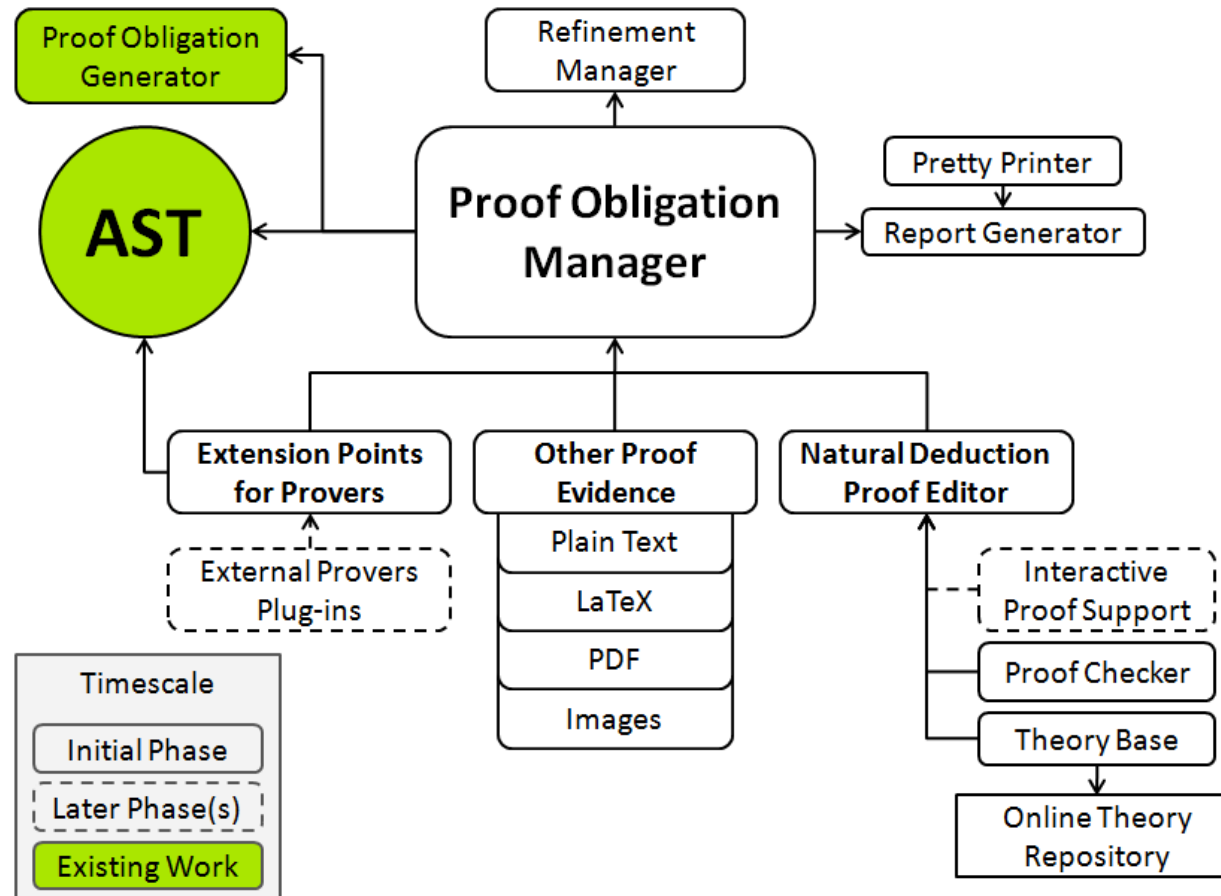
# Proof Artifacts: Natural Deduction

- Proof editor
  - file format
  - automated line numbering, syntax highlighting
- Proof checker
  - plus automated rechecking upon specification change
- Directory of theorems
  - plus online repository
- User-guided proof
  - e.g. like mural, AI4FM

# Proof Artifacts: Other Evidence

- E.g. proof by inspection, semi-formal proofs, structured arguments
- Also older proofs in other forms (e.g. scans)
- Plain text, LaTeX, PDF, PNG/JPEG etc.
- “Blue” mnemonic
  - human-checked
  - can warn of changed specification

# Proposed Components





# Mockup

The screenshot shows the VDM - Mondex/retr-form - Overture Tools IDE. The interface includes a menu bar (File, Edit, Navigate, Search, Project, Run, Window, Help), a toolbar with icons for file operations and debugging, and a main workspace divided into three panes.

**Proof Explorer:** A tree view on the left showing the project structure. The 'Mondex' folder contains several files with status icons (green checkmarks for success, red crosses for failure):

- bal-purses-equal (checked)
- con-assign (checked)
- ConWorld-AbWorld-form (checked)
- diff-{a}-C (checked)
- dom-purses-= (failed)
- retr-adequate (checked)
- retr-form (failed)
- sumval-defn1 (checked)
- sumval-form (checked)
- TransferLost-dom (failed)

**Code Editor:** The main pane shows a proof script for `*retr-form`. The script starts with `from mk-ConWorld(auth, cp) : ConWorld` and contains five numbered lines:

```
1 inv-ConWorld(auth, cp)           by inv-ConWorld-I(h1)
2 dom cp subset auth                by unfolding(1)
3 mk-ConWorld(auth, cp).cp:
  map PurseId to ConPurse           by conpurses-form(h1)
4 mk-ConWorld(auth, cp).cp = cp     by conpurses-defn(h1)
5 cp: map PurseId to ConPurse       by ==-type-inherit-right(3,4)
```

Below these lines, there is a red error icon and the text: `infer retr(mk-ConWorld(auth, cp)) : AbWorld by`, where `by` is underlined in red.

**Theory Base:** The bottom pane shows a table with the following content:

Theory Base	
dom-defn	
dom-defn-{ -> }	dom { -> } = {}
dom-defn-addm	dom addm(a  -> b, m) = addm(a, dom m)

# Roadmap

- Requirements document
  - based on community feedback
- Initial proof perspective
  - including ‘other evidence’ artifact support
- Natural deduction file format and editor
- Pilot study for external prover e.g. [Ver&10]

# Thanks for Listening

- Any questions?