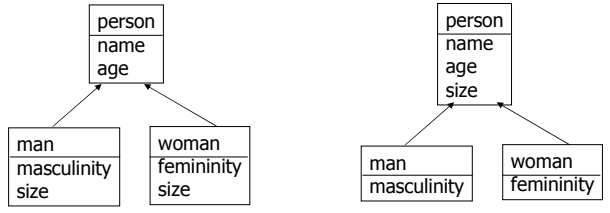
Specifying mappings

- Examples using VML
- Graphical formalisms for mapping languages
  - VML-G
  - Forms-based mapper

Inheritance example



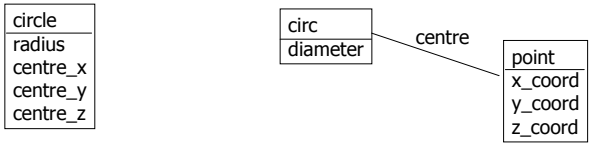
VML for inheritance

```
inter_view(idm, integrated, view1, read_write, complete).
inter_class([person],[person],
  equivalences(  name = name,
                 age = age)
).
inter_class([man],[man],
  inherits(inter_class([person],[person])),
  equivalences(  size = size,
                 masculinity = masculinity)
).
inter_class([woman],[woman],
  inherits(inter_class([person],[person])),
  equivalences(  size = size,
                 femininity = femininity)
).
```


Differing conceptions example

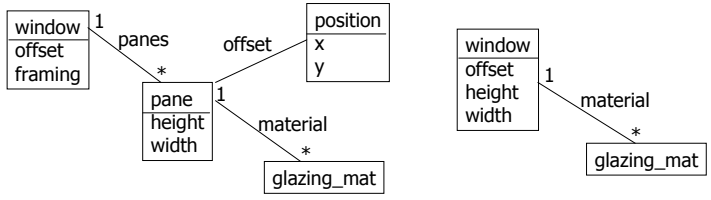

```
inter_view(idm, integrated, view1, read_write, complete).
inter_class([point],[point],
  equivalences(  r * cos(theta) = x_coord,
                 r * sin(theta) = y_coord,
                 r = sqrt(sqr(x_coord) + sqr(y_coord)),
                 theta = tan_1(y_coord / x_coord))
).
```

Structure difference example



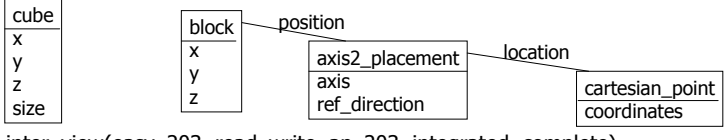

```
inter_view(idm, integrated, view1, read_write, complete).
inter_class([circle],[circ],
  equivalences(  radius * 2 = diameter,
                 centre_x = centre=>x_coord,
                 centre_y = centre=>y_coord,
                 centre_z = centre=>z_coord)
).
```


One way mapping example




```
inter_view(idm, integrated, view1, read_only, complete).
inter_class([window],[window],
  equivalences(    offset = offset,
                   panes[1]=>material = material,
  maximum(panes=>(offset=>y + height))- minimum(panes=>offset=>y) = height,
  maximum(panes=>(offset=>x + width))- minimum(panes=>offset=>x) = width)
).
```


Collapsing structure example



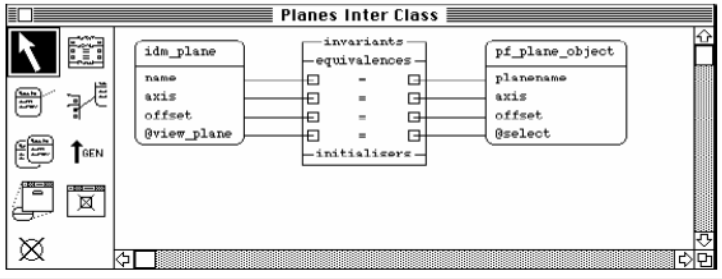

```
inter_view(easy_203, read_write, ap_203, integrated, complete).
inter_class([cube],[block],
  invariants(      block.x = block.y,
                   block.y = block.z),
  equivalences(  size = x,
                   x = position=>location=>coordinates[1],
                   y = position=>location=>coordinates[2],
                   z = position=>location=>coordinates[3]),
  initialisers(    [0,0,1] = position=>axis=>vector,
                   [0,0,1] = position=>ref_direction=>vector)
).
```


Why do we need a graphical formalism?

- Easier to comprehend the mapping?
  - Diagrams can be read by non-experts
- Easier to check that everything is mapped?
  - All attributes for classes from both schemas
  - All combinations of invariants for the same classes
  - All classes in a schema?
- Faster to specify the mappings?
  - Support for mapping equation syntax
- Graphical mapping specification tool can provide validation support and schema management support

VML-G

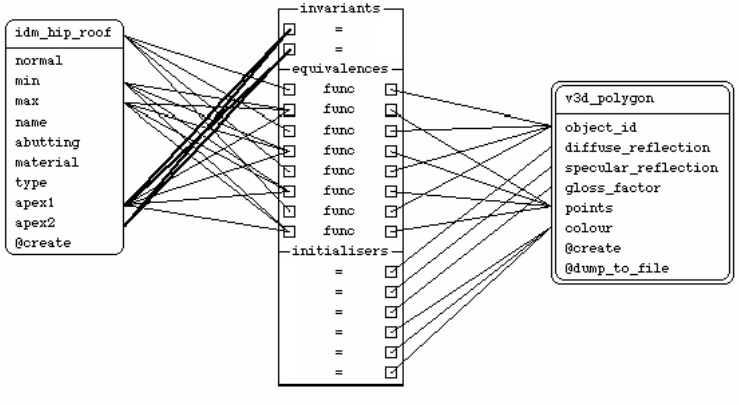
- Provides visualisation of main components within a VML mapping specification
  - Classes
  - inter_class specification



VML-G approach

- Icons for classes
  - View all attributes and methods (or subsets)
- Icon for an inter_class
  - Break up invariants, equivalences, initialisers
  - Equation type is denoted by a symbol
  - Full specification of each equation is viewed in a textbox
  - Full text of the inter_class specification can be viewed in a text window
- Wiring approach to join attributes and classes to equations in the inter_class

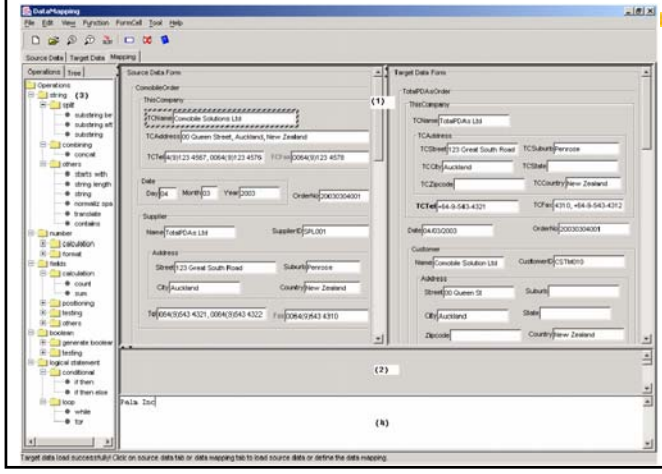
VML-G example



Forms-based mapper

- VML and VML-G are aimed at analyst programmers
- In the real world a business analyst often knows the correspondences between information in different business systems
  - How to get this information out of them?
    - Business people deal with forms containing information
    - Provide a forms-based view of the two representations
    - Allow the business analyst to connect form components together to specify the mapping
    - This specification can be refined by programmers later

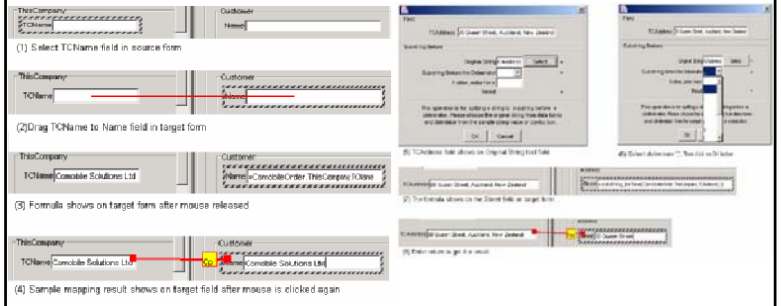
Forms-based mapper (XML)



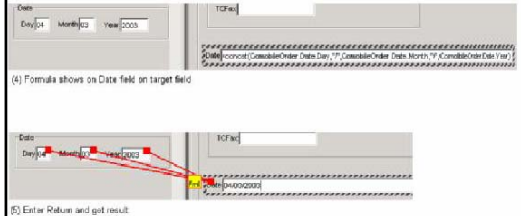
Drag and drop mapping

One-to-one mapping

One-to-many, function-based mapping

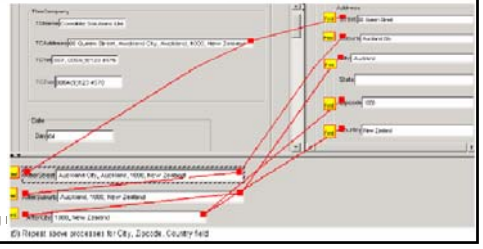


Drag and drop mapping



Many-to-one attribute mapping

Visualising temporary elements



Graphical formalisms

- Adjunct to the textual mapping notation
  - Provide a more comprehensible view of a mapping
  - Enable greater checking of mapping specifications
  - Tailorable to specifier specific notations