

## Specifying mappings

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

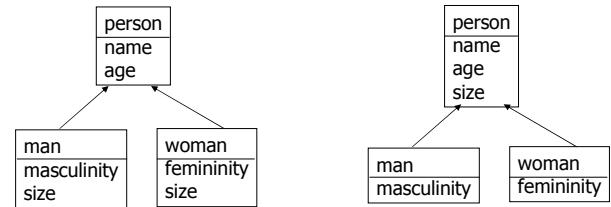
COMPSCI 732 FC §5. Specifying mappings

## 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)
).
```

COMPSCI 732 FC §5. Specifying mappings

## Inheritance example



COMPSCI 732 FC §5. Specifying mappings

## Differing conceptions example

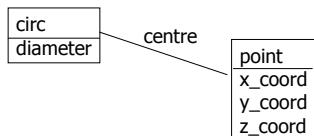


```
inter_view(idm, integrated, view1, read_write, complete).
inter_class([point],[point],
equivalences( r * cos(theta) = x_coord,
r * sin(theta) = y_coord,
r = sqrt(sqr(x_coord) + sqr(y_coord)),
theta = tan_1(y_coord / x_coord))
).
```

COMPSCI 732 FC §5. Specifying mappings

## Structure difference example

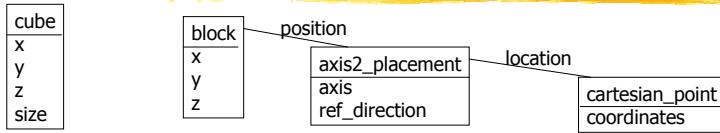
circle
radius
centre_x
centre_y
centre_z



```
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
).
```

COMPSCI 732 FC §5. Specifying mappings

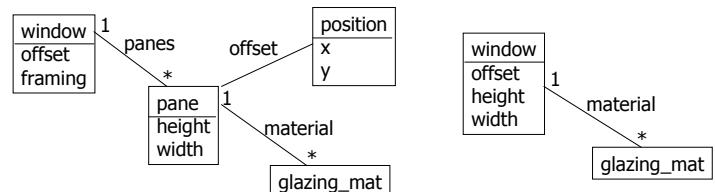
## Collapsing structure example



```
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)
).
```

COMPSCI 732 FC §5. Specifying mappings

## 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
).
```

COMPSCI 732 FC §5. Specifying mappings

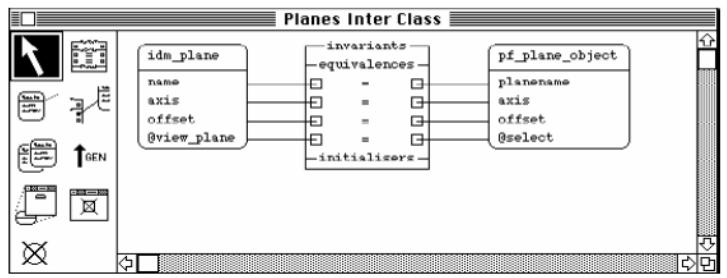
## Why do we need a graphical formalism?

- 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

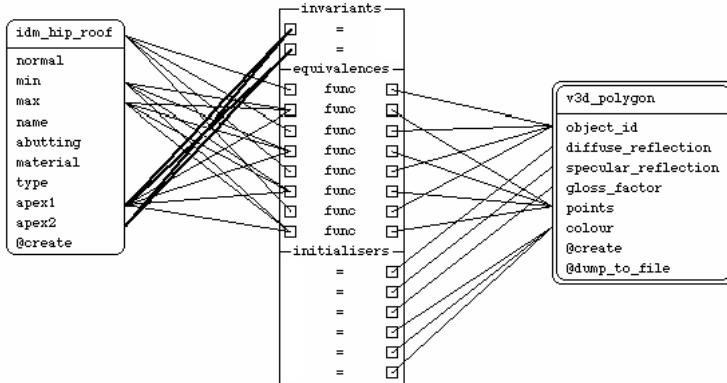
COMPSCI 732 FC §5. Specifying mappings

## VML-G

- Provides visualisation of main components within a VML mapping specification
  - Classes
  - inter\_class specification



## VML-G example



COMPSCI 732 FC §5. Specifying mappings

## 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

COMPSCI 732 FC §5. Specifying mappings

## 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

COMPSCI 732 FC §5. Specifying mappings

## Forms-based mapper (XML)

The screenshot shows the DataMapper application interface. On the left, there's a toolbar with various icons for file operations, mapping, and help. Below the toolbar, a navigation bar has tabs for 'Source Data', 'Target Data', and 'Mapping'. Under 'Source Data', there's a tree view of 'Operations' (with 3 items) and a list of 'Supplier' objects. One object is selected, showing its details: 'Name' (TotalData Ltd), 'Address' (223 Great South Road, Suburb/Province: Auckland, Country: New Zealand), and 'Phone' (064-9543432). On the right, under 'Target Data', there's a tree view of 'Operations' (with 1 item) and a list of 'Customer' objects. One object is selected, showing its details: 'Name' (TotalData Ltd), 'Address' (223 Great South Road, Suburb/Province: Auckland, Country: New Zealand), and 'Phone' (064-9543432). A central workspace shows the mapping between these two objects. A yellow box highlights the 'Operations' tree on the left.

## Drag and drop mapping

### One-to-one mapping

(1) Select TCName field in source form

(2) Drag TCName to Name field in target form

(3) Formula shows on target form after mouse released

(4) Sample mapping result shows on target field after mouse is clicked again

### One-to-many, function-based mapping

(5) Clicked TCName on Original String field

(6) The formula above is on the Target field

COMPSCI 732 FC §5. Specifying mappings

## Drag and drop mapping

The screenshot shows the DataMapper application interface. On the left, there's a toolbar with various icons for file operations, mapping, and help. Below the toolbar, a navigation bar has tabs for 'Data', 'Operations', and 'Mappings'. Under 'Data', there's a tree view of 'Operations' (with 1 item) and a list of 'Customer' objects. One object is selected, showing its details: 'Name' (TotalData Ltd), 'Address' (223 Great South Road, Suburb/Province: Auckland, Country: New Zealand), and 'Phone' (064-9543432). On the right, there's a tree view of 'Operations' (with 1 item) and a list of 'Customer' objects. One object is selected, showing its details: 'Name' (TotalData Ltd), 'Address' (223 Great South Road, Suburb/Province: Auckland, Country: New Zealand), and 'Phone' (064-9543432). A central workspace shows the mapping between these two objects. A yellow box highlights the 'Operations' tree on the left. Below the workspace, there are four numbered steps: (4) Formula shows on Date field in target field, (5) Enter Return and get result, (6) Visualising temporary elements, and (7) Many-to-one attribute mapping.

## 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

COMPSCI 732 FC §5. Specifying mappings