

Using Concrete Visual Notations as First Class Citizens for Model Transformation Specification

Iman Avazpour
John Grundy

SWIN
BUR
* NE *

SWINBURNE
UNIVERSITY OF
TECHNOLOGY

Swinburne
▶ think forward



Outline

- Why CONVERt
- Approach
- Basic examples
- Case study – Minard's Map
- Future work

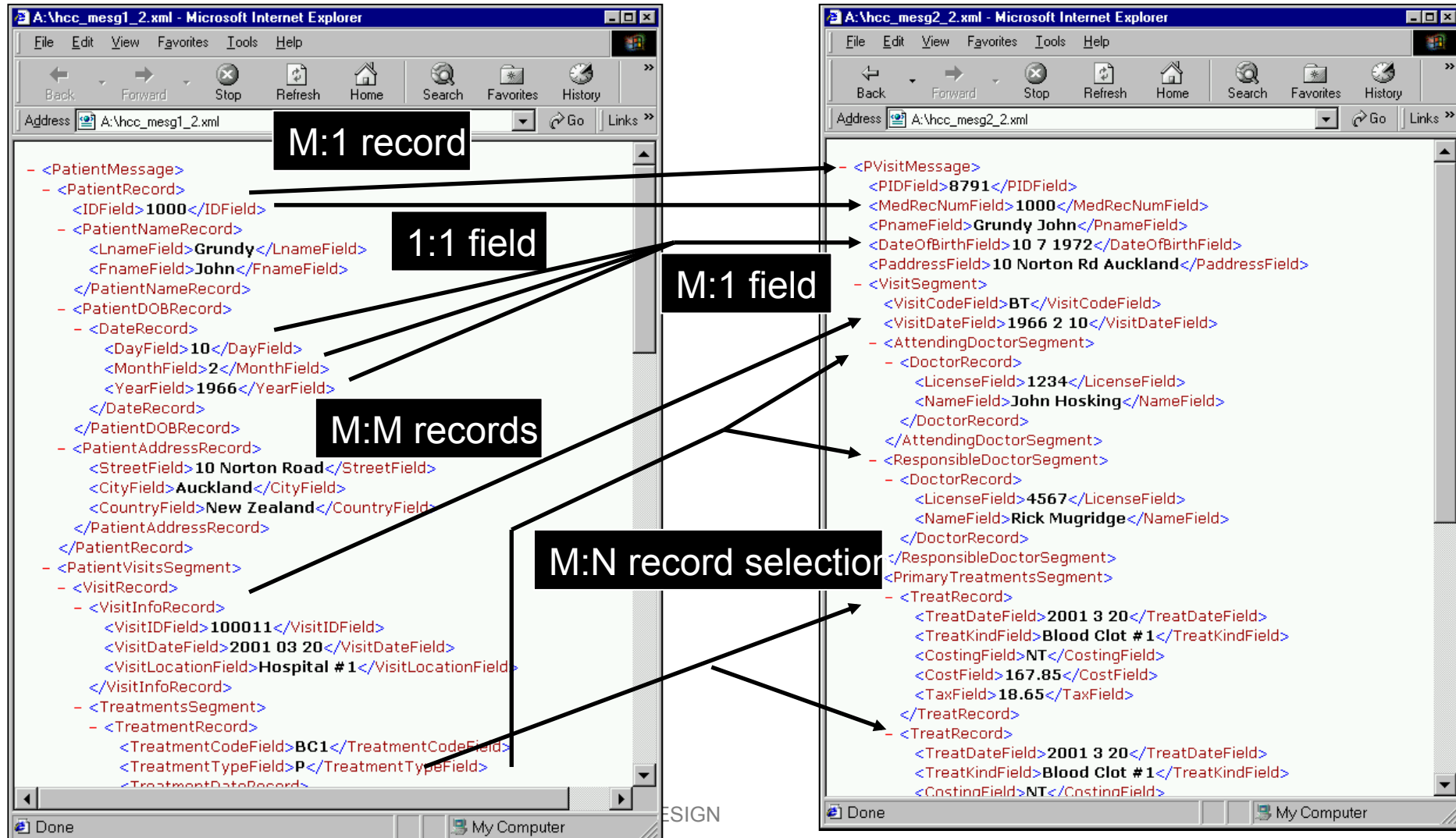


Motivation

- Complex data mapping is hard
 - Programming languages & APIs
 - Scripting languages e.g. QVT, ATL, XSLT
 - Abstraction visual mappers
 - Concrete, domain-specific visual mappers
- Wanted to provide end-users with concrete, example-based data mapping tool
 - Specify own visualisations of complex data
 - Visualise source / target model data
 - Drag and drop between elements to specify mappings
 - Generate model mapping script / code (XSLT)

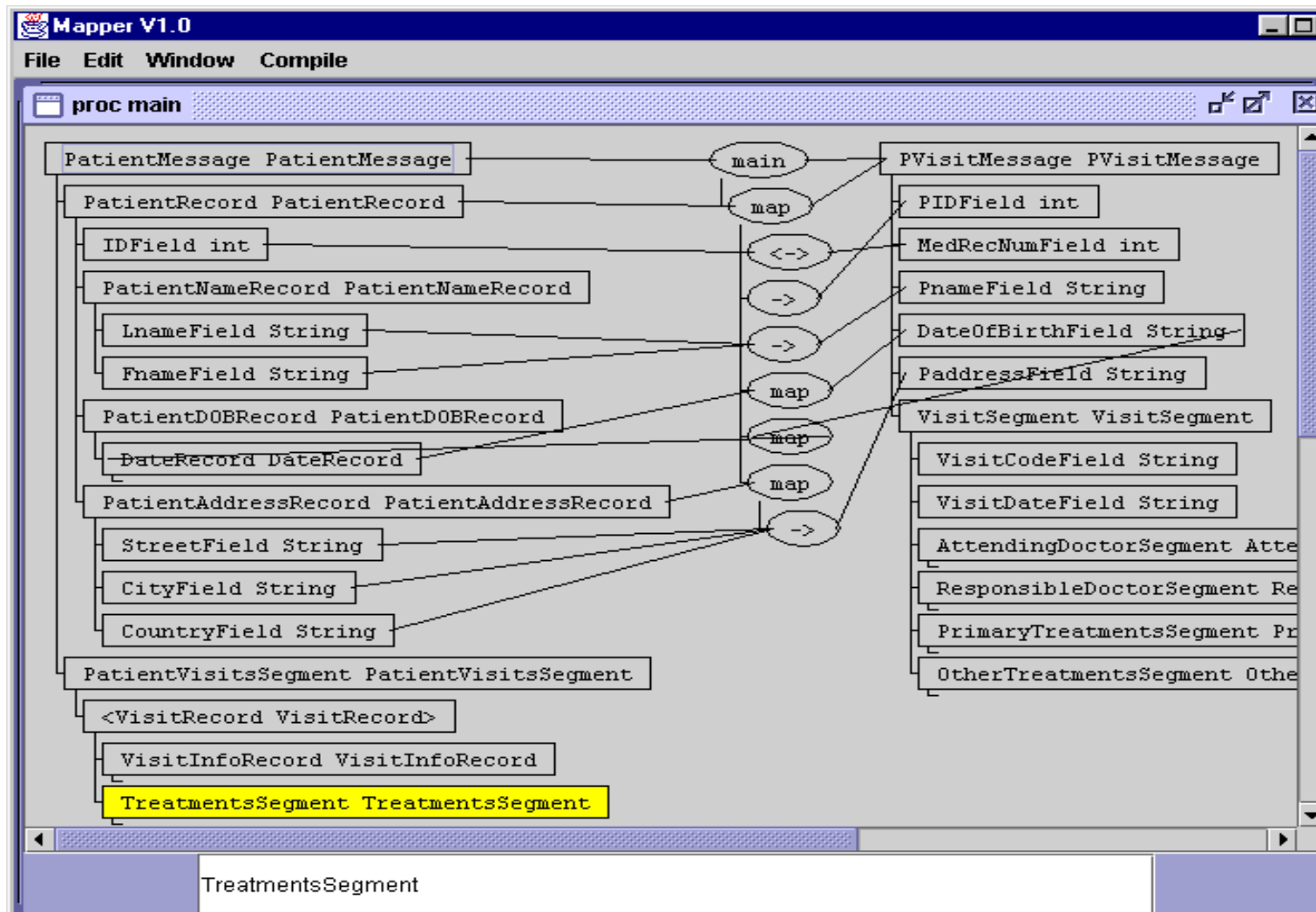


Example





c.f. Orion health message mapper (ASE 2001)



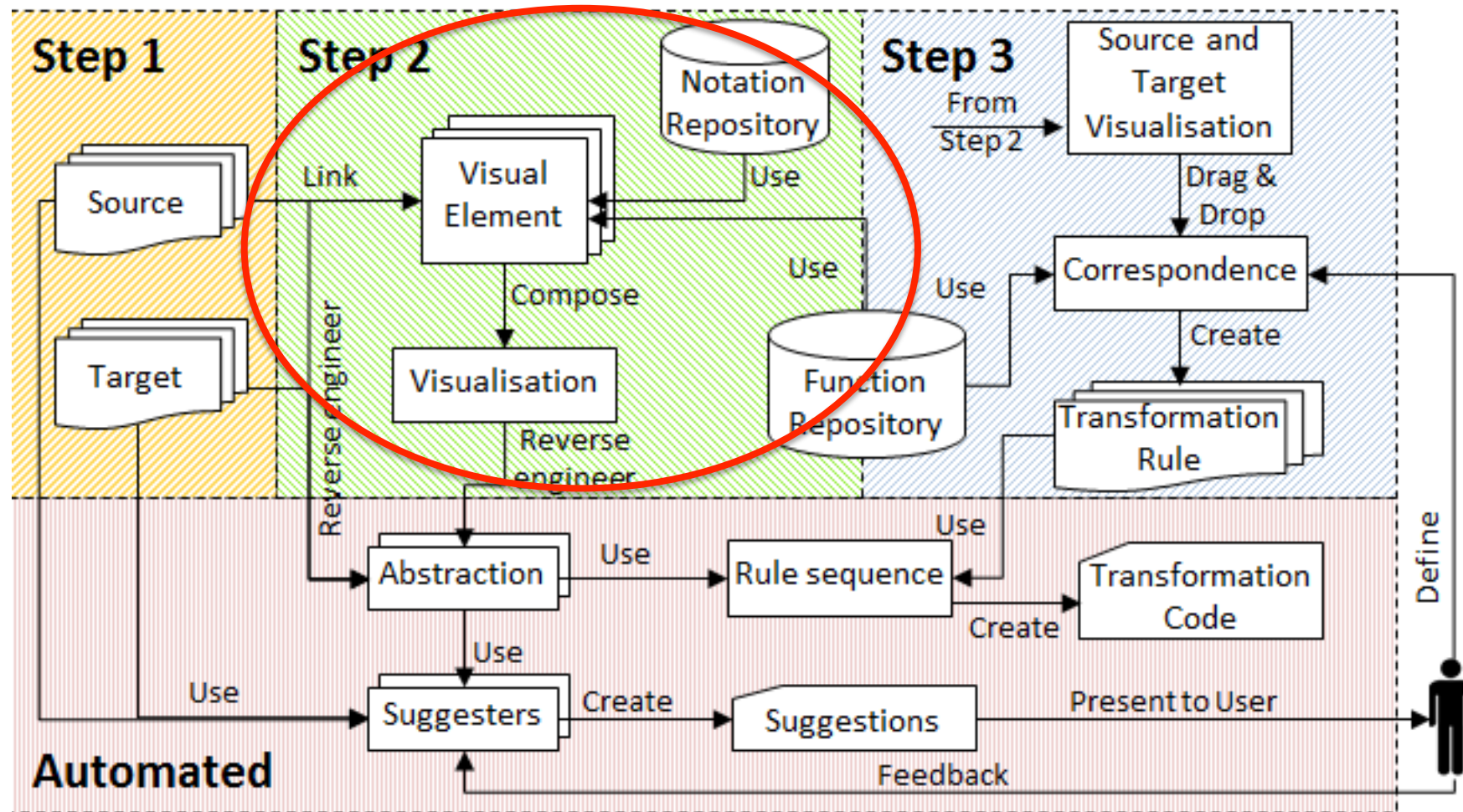


c.f. Form-based mapper (HCC 2002)

The diagram illustrates a form-based mapper with two panels: 'Source Data Form' and 'Target Data Form'. The 'Source Data Form' contains fields for a person (id: 1234, name: Grundy, John, email: john-g@cs.auckland.ac.nz, href: www.cs.auckland.ac.nz/~john-g) and an order (date: 20th March 2002, item: book: How to use Java, qty: 1, price: \$49.95). The 'Target Data Form' contains fields for an order (date: 20/03/02, created: , total_price: 49.95) and customer_info (name: John Grundy, address:) and an item (book_info: How to use Java, quantity: 1, total_cost: 49.95). A red arrow points from the 'date' field in the source order to the 'date' field in the target order. Black arrows point from the 'book' field in the source item to the 'book_info' field in the target item, and from the 'price' field in the source item to the 'total_cost' field in the target item. A code snippet at the bottom reads: `orders.order.date = Date(person.orders.order.date,"ddmmyy")`

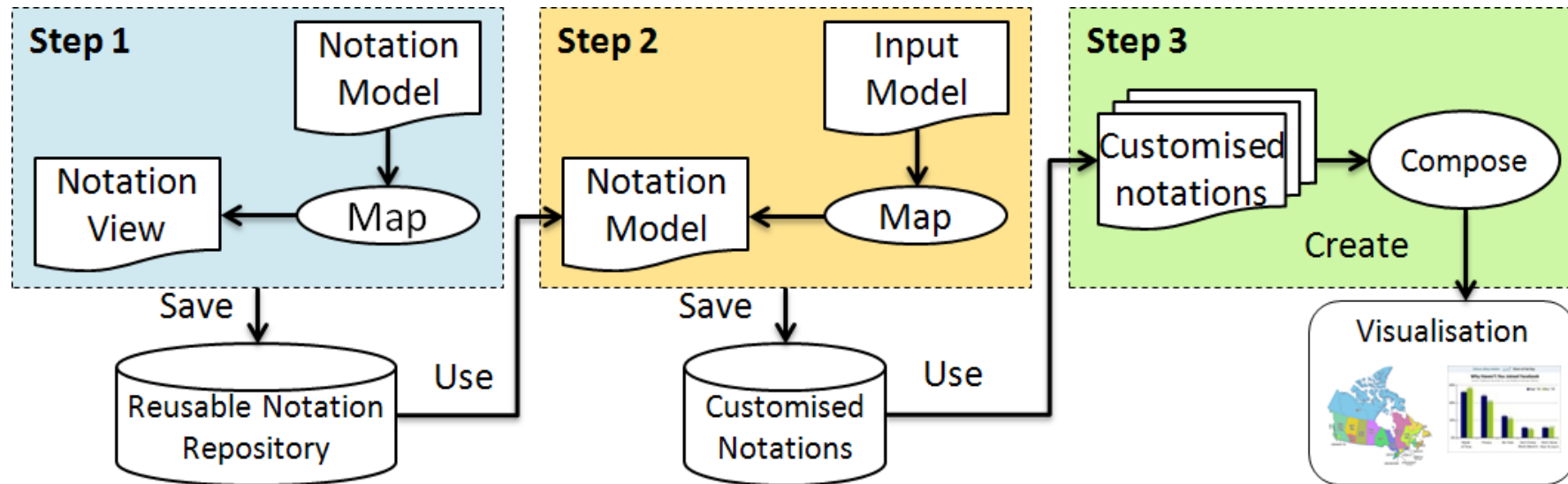


Our New Approach - CONVERt



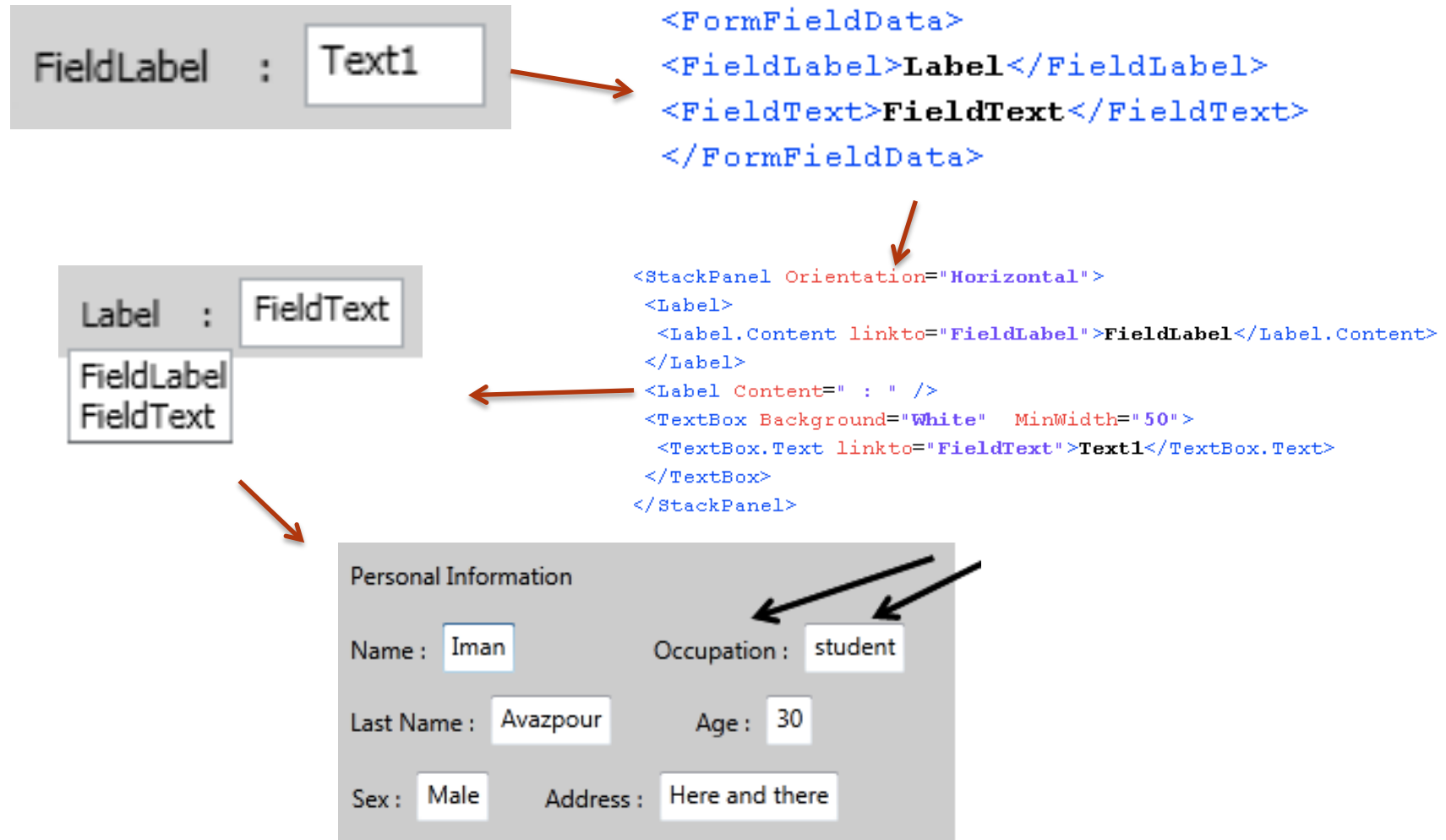


CONVERt – Specifying Concrete Model Visualisations



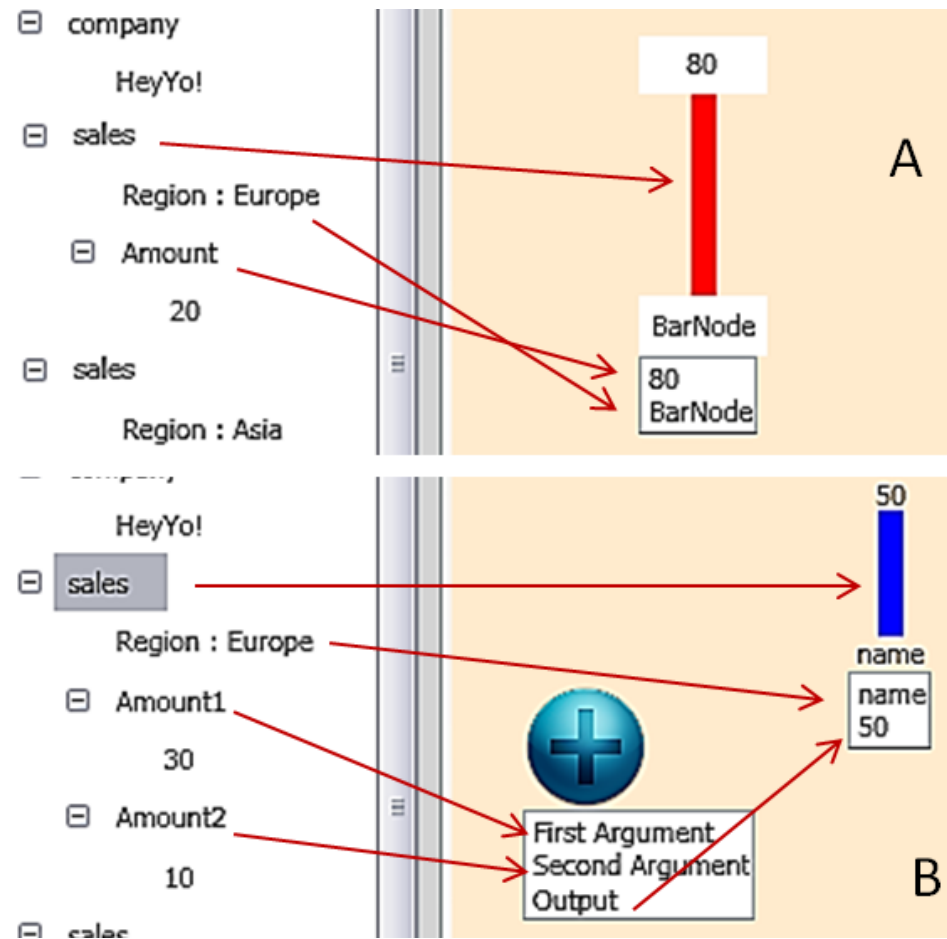


1. Specify Notational Elements



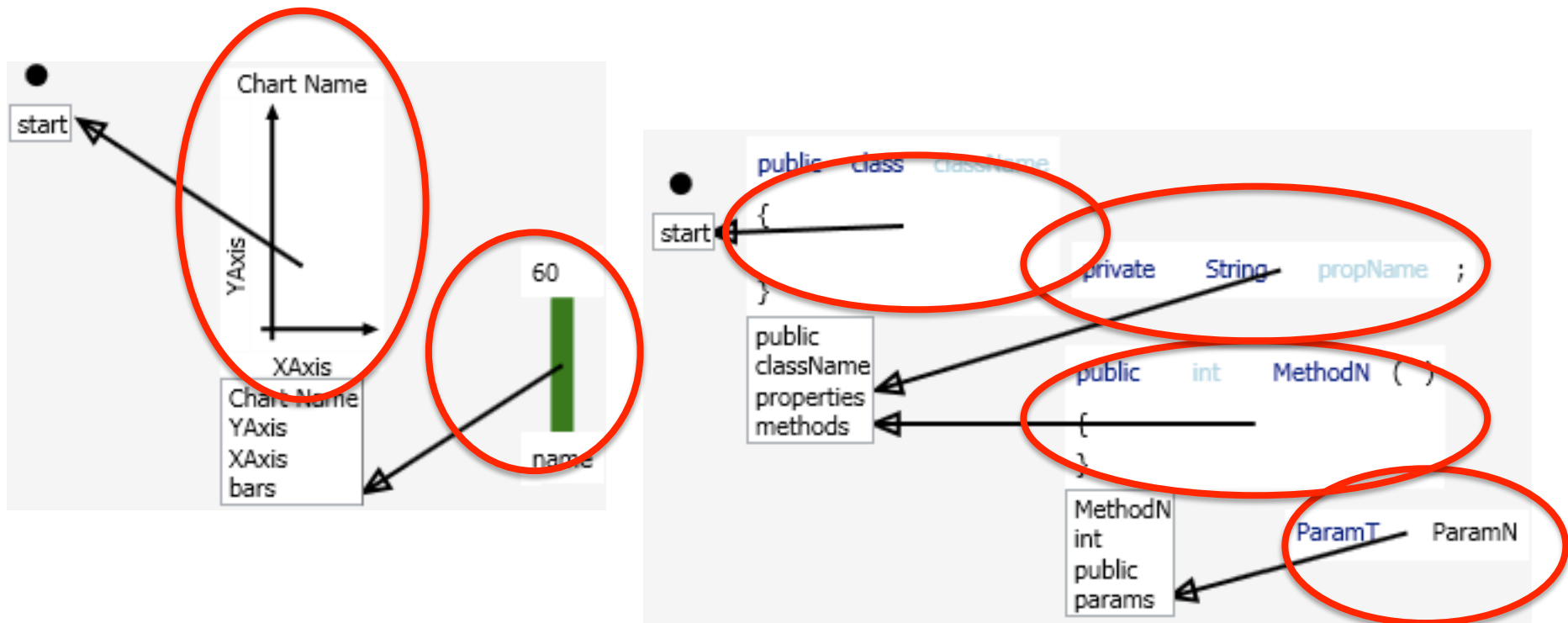


2. Map data fields to elements





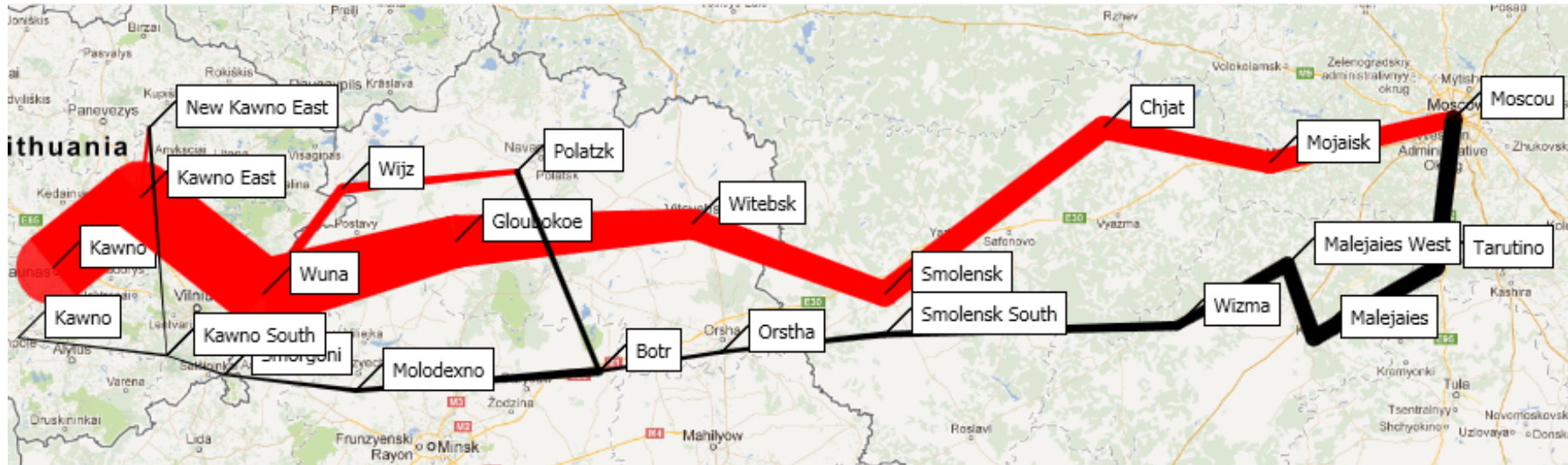
3. Compose basic notational elements





Case study – Minard’s Map (see the paper!)

Figurative Map of successive losses in men of the French army in Russian Campaign 1812 ~ 1813



<MapData>

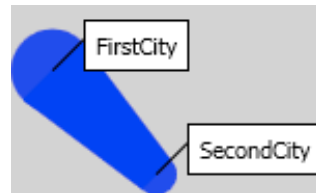
<Description>Discription goes here</Description>

<Movements>Troop Movements</Movements>

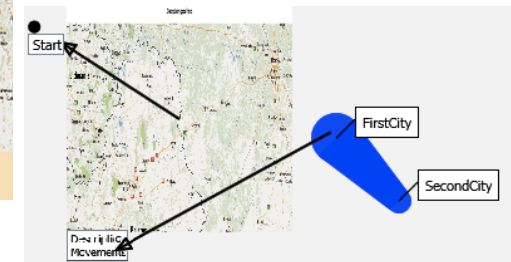
</MapData>

```

<StackPanel Orientation='Vertical' Height='290' Width='716'>
  <TextBlock Height='27' TextAlignment='Center' Background='White'>
    <TextBlock.Text linkto='Description'>Description of the map
  </TextBlock.Text>
</TextBlock>
<Canvas Height='263' Width='716'>
  <Canvas.Background>
    <ImageBrush ImageSource='NapoleonMap.bmp' />
  </Canvas.Background>
  <local:Minard Height='263' Width='716' callfor='Movements'>
  </local:Minard>
</Canvas>
</StackPanel>
    
```



Record	HeadCount	Lost	Status
	422000	0	Advancing





Mapping example: CAD building design data to Hierarchical org data

The screenshot shows a software interface with the following components:

- Source Visualisation:** A floor plan of a 'New Green Building' divided into three sections: 'Living Area' (containing 'Open Kitchen' and 'Kitchen'), 'Upper Rooms' (containing 'Room 2', 'Room 3', and 'Room 4', all labeled 'BedRoom'), and 'Third Floor Rooms' (containing 'Room 5' and 'Room 6', both labeled 'BedRoom'). A 'Toilet' is also shown in the Living Area. A 'Geometry' panel on the left lists 'Name' and 'Type'.
- Target Visualisation:** A hierarchical tree structure for 'CityCouncil'. It branches into 'Ground', 'First Floor', and 'Second Floor'. Under 'Ground' are 'Toilet' and 'G1'. Under 'First Floor' are 'Toilet', 'Kitchen', and 'S104'. Under 'Second Floor' are '201', 'Toilet', '202', '101', '203', '102', '204', and '103'. A 'Stock1' node is also present.
- Mapping Functions:** A panel with various icons for mapping operations, including a double-headed arrow and a plus sign. A label 'b' is placed below these icons.
- Mapping Rules:** A panel showing a rule that maps 'KKitchen' (from the 'Kitchen' in the source) to 'room1' (from the '201' in the target). A double-headed arrow indicates the bidirectional relationship. A label 'c' is placed below this rule.
- Visualisation Mapping / Rule designer:** A panel at the bottom with a list of mapping rules:
 - Map BuildingNode To BuidlingNode (with a green checkmark and a red X)
 - Map BuidlingNode/Name To BuidlingNode/Name (with a green checkmark and a red X)
 - Map BuidlingNode/Floors To BuidlingNode/Floors (with a green checkmark and a red X)
 A label 'a' is placed to the right of this panel.



Evaluation and Future Work

- Range of example models visualised and mapped
 - Business, Buildings and eHealth, software (MDE)
- Range of end users surveyed
 - 11 people - business charts; 12 people – MDE (UML +Java code); Results => its pretty good!!! 😊
- Key issues to improve:
 - Scaling (set of “Suggesters” provided)
 - More reusable functions, notational elements
 - By-example function specification & reuse
 - Other implementations e.g. ALT, JavaScript/HTML etc
 - Live, incremental visualisation; web-based GUI



Summary

- Support end users to **interactively specify rich, human-centric visualisations** of complex data using a visual, drag-and-drop, by-example approach
- Support end users to **generate reusable visualisation implementations** from these high-level specifications
- Allow end users to **reuse their generated, reusable model visualisations** to visualise two (or more) complex data sets (i.e. example models)
- Support end users to **specify model element mappings** between these data sets **via drag-and-drop between their concrete visualisation elements**
- **Generates complex, reusable model transformation implementations** from these visually specified mappings

Questions?

CONVERt Videos & Web site:

<http://www.youtube.com/watch?v=RExa0MT-zqU>

<https://sites.google.com/site/iavazpour/tools-manuals>



Swinburne
▶ think forward



References

Avazpour, I., Grundy, J.C., Grunske, L., Tool Support for Automatic Model Transformation Specification Using Concrete Visualisations, 2013 IEEE/ACM International Conference on Automated Software Engineering, Palo Alto, CA, USA, 11-15 Nov 2013, IEE CPS

Avazpour, I., Grundy, J.C., Using Concrete Visual Notations as First Class Citizens for Model Transformation Specification, 2013 IEEE Symposium on Visual Languages and Human-Centric Computing, San Jose, CA, USA, Sept 15-19 2013, IEEE CPS.

Avazpour, I. and Grundy, J.C. CONVErT: A Framework for Complex Model Visualisation and Transformation, 2012 IEEE International Symposium on Visual Languages and Human-Centric Computing, Innsbruck, Austria, Sept 30-Oct 4 2012, IEEE CS Press.

Grundy, J.C, Hosking, J.G., Amor, R., Mugridge, W.B., Li, M. Domain-specific visual languages for specifying and generating data mapping system, Journal of Visual Languages and Computing, vol. 15, no. 3-4, June-August 2004, Elsevier, pp 243-263,

Li, Y., Grundy, J.C., Amor, R. and Hosking, J.G. A data mapping specification environment using a concrete business form-based metaphor, In Proceedings of the 2002 International Conference on Human-Centric Computing, IEEE CS Press

Grundy, J.C., Mugridge, W.B., Hosking, J.G. and Kendal, P. Generating EDI Message Translations from Visual Specifications, In Proceedings of the 16th International Conference on Automated Software Engineering, San Diego, 26-29 Nov 2001, IEEE CS Press, pp. 35-42.