A data mapping specification environment using a concrete business form-based metaphor

THE UNIVERSITY OF AUCKLAND
www.auckland.ac.nz

Yongqiang Li, John Grundy, Robert Amor and John Hosking

Department of Computer Science

University of Auckland, New Zealand

Outline

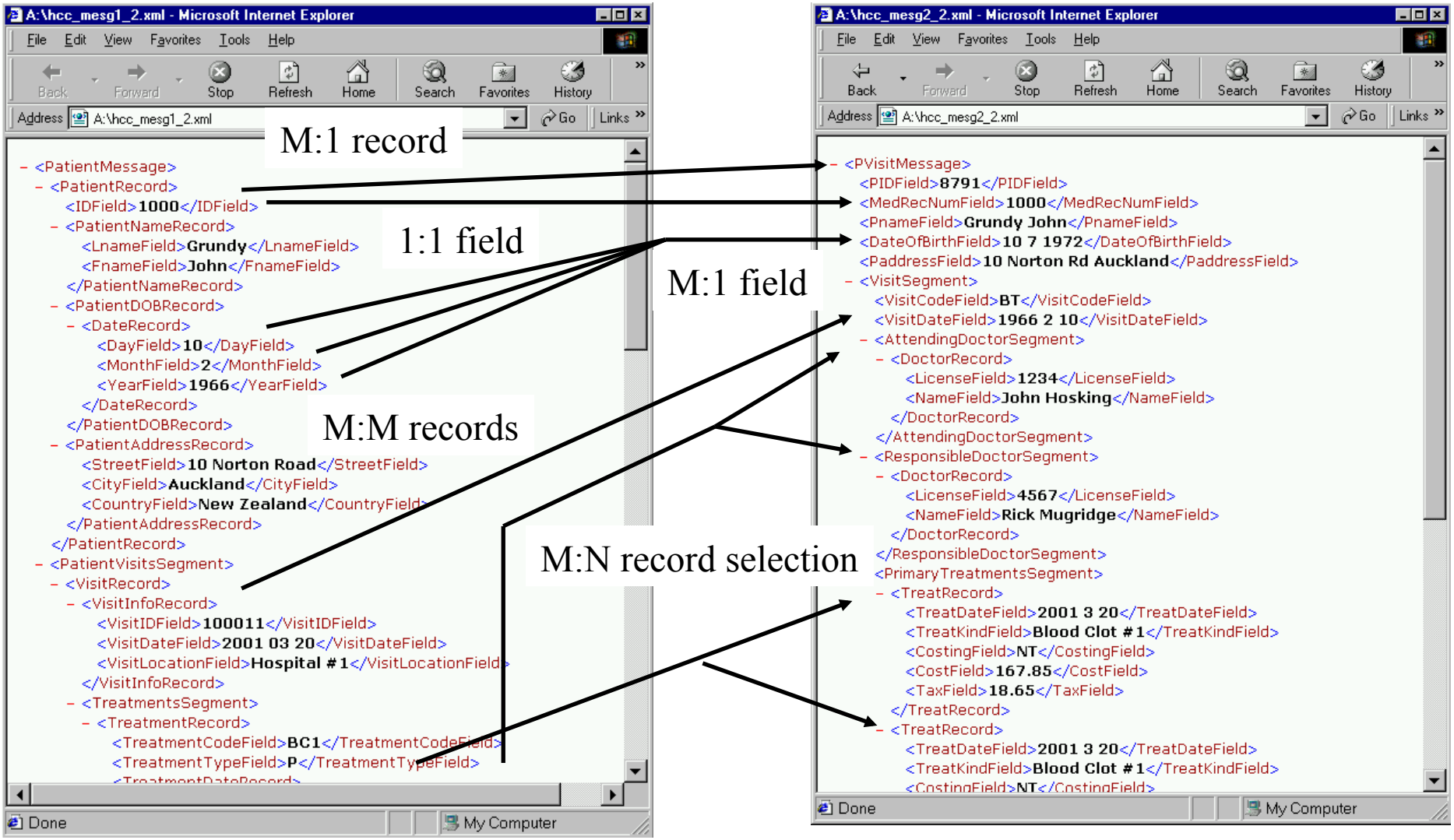
THE UNIVERSITY OF AUCKLAND
www.auckland.ac.nz

- Motivation for our work
- Need for data mapping systems
- Existing approaches to data mapping
- Our approach: "form-based copying"
- Overview of prototype environment
- Examples of specifying mappings
- Data transformation code generation
- Future work
- Summary

Motivation

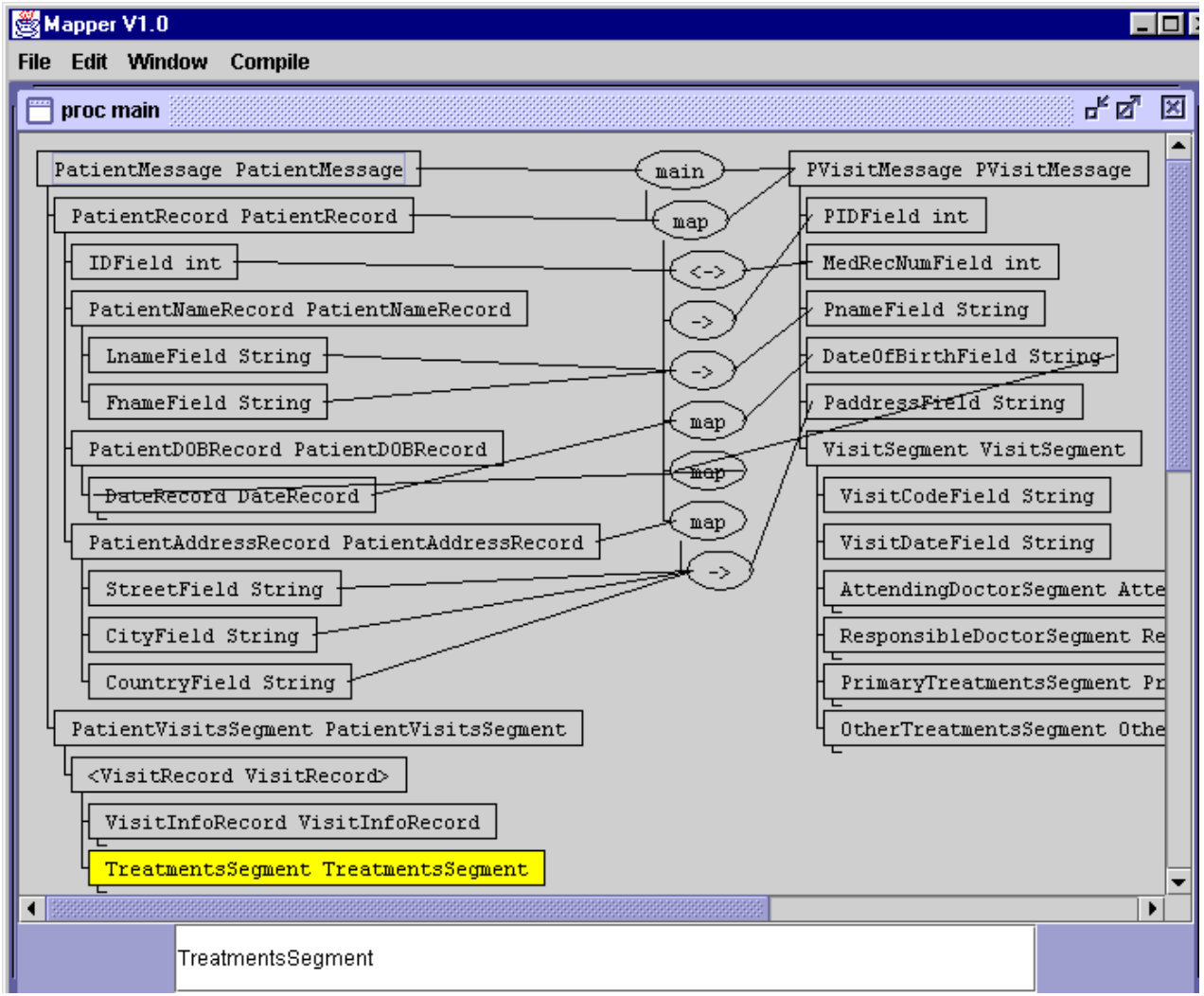
- **Data transformation required in many problem domains:**
  - Message-based systems integration
  - Database integration
  - ERP system integration
  - Document exchange
- **Existing approaches to specifying data mappings:**
  - Programming, scripting (textual)
  - Tree-based visual linking
  - Custom "visual language" (programmatic)

Data Mapping Example



Existing Tool Example

THE UNIVERSITY OF AUCKLAND
www.auckland.ac.nz

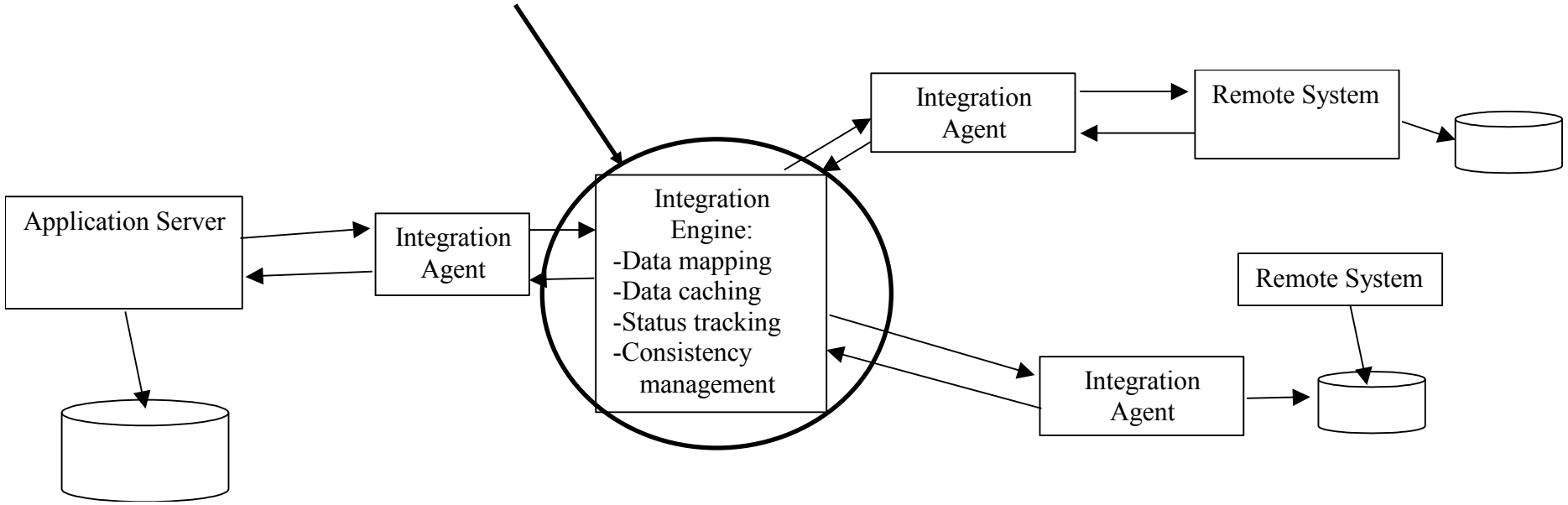


- Common approach to specifying mappings
- Tree based schema and mapping functions
- Programmers are users
- Complex mapping language underneath

Context for Our Work

THE UNIVERSITY OF AUCKLAND
www.auckland.ac.nz

- Enterprise systems integration
- Need to be able to specify complex data transformations
- Want END USERS to be able to specify complex data mappings…



Our Approach

THE UNIVERSITY OF AUCKLAND
www.auckland.ac.nz

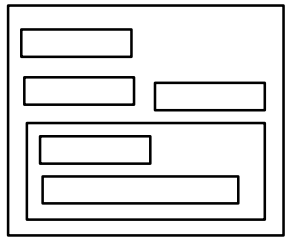
- Aim: to support end-user specification of complex mappings
- Users = "business analysts"
- Want to generate mapping implementations
- Most businesses manually implement such data exchange manually via "copying" from one business form to another (hard-copy or on a computer)
- We wanted to support this "form-copying" metaphor in an end-user oriented mapping specification tool…

Process

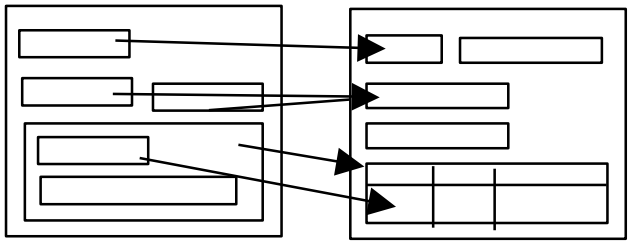
1. Analyst imports meta-data from source and target enterprise systems

Meta-data e.g. XML DTDs

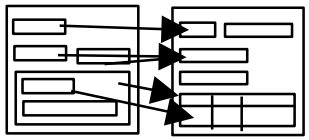
2. Default business form layouts generated. Analyst can rearrange layout to better-reflect actual business forms.



3. Analyst specifies 1:1, 1:n, m:1 group and field correspondences i.e. specifies how to "copy" data from one form to the other

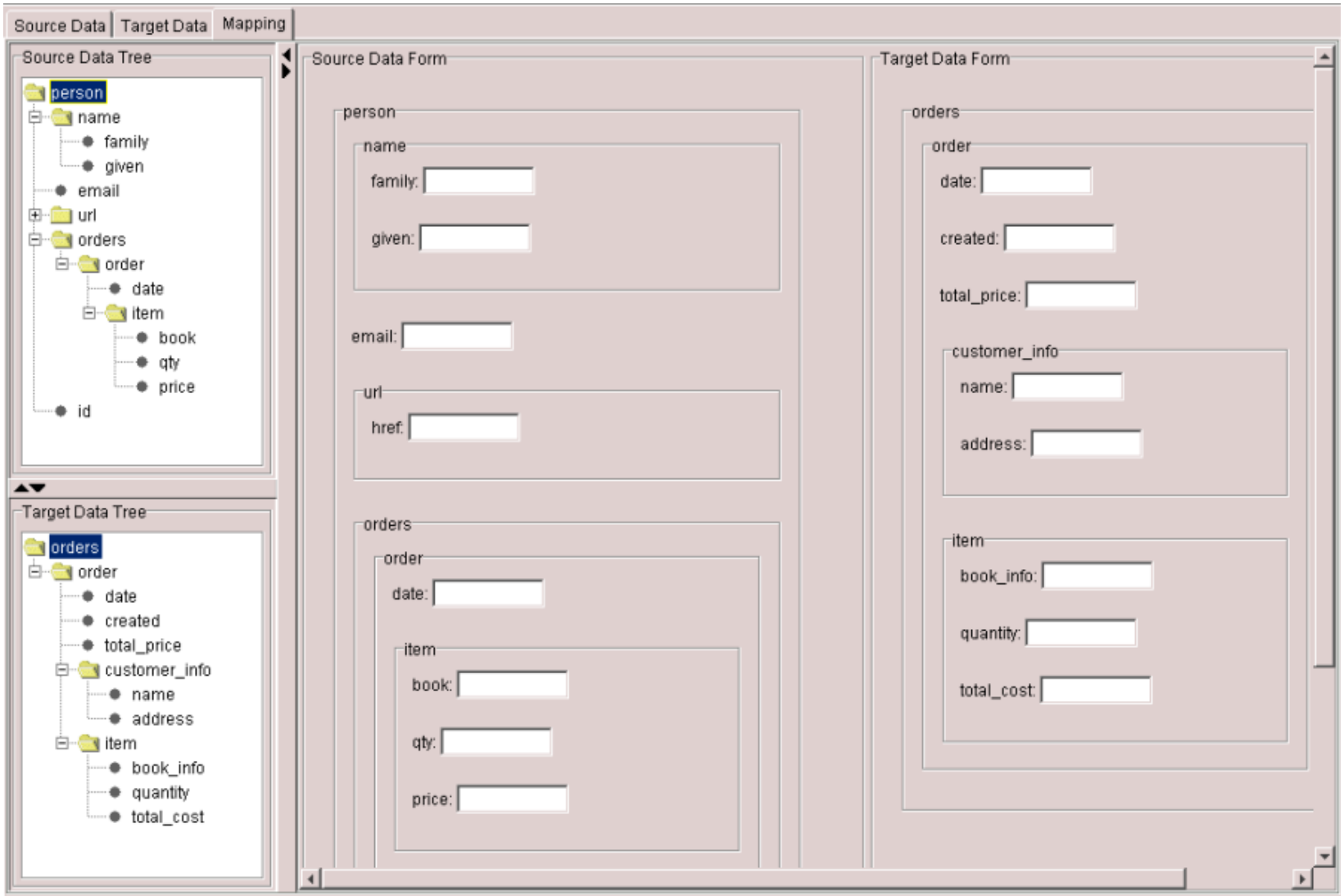


4. Data transformation implementation generated from specification

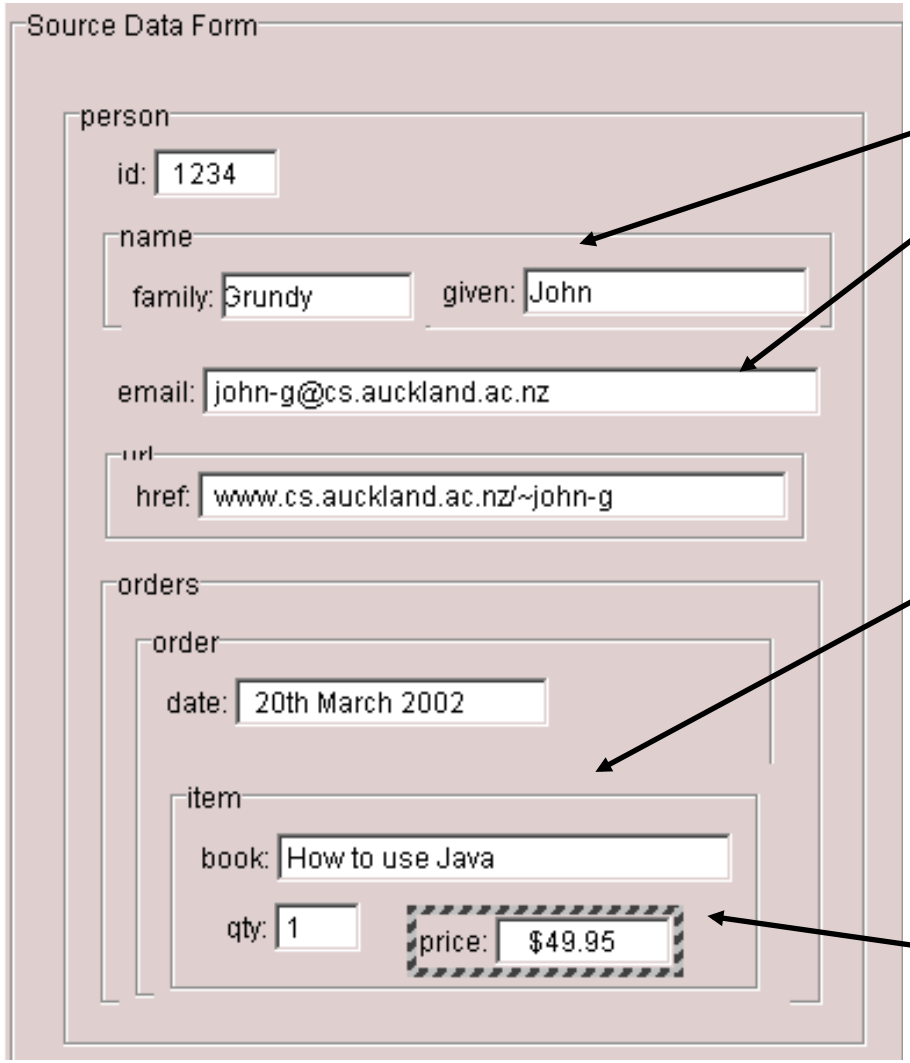


```
<xsl…>
  <xsl:apply-templates…>
…
</xsl:…>
```

Example of Form-based Visualisation of Documents



Rearranging Form Layout

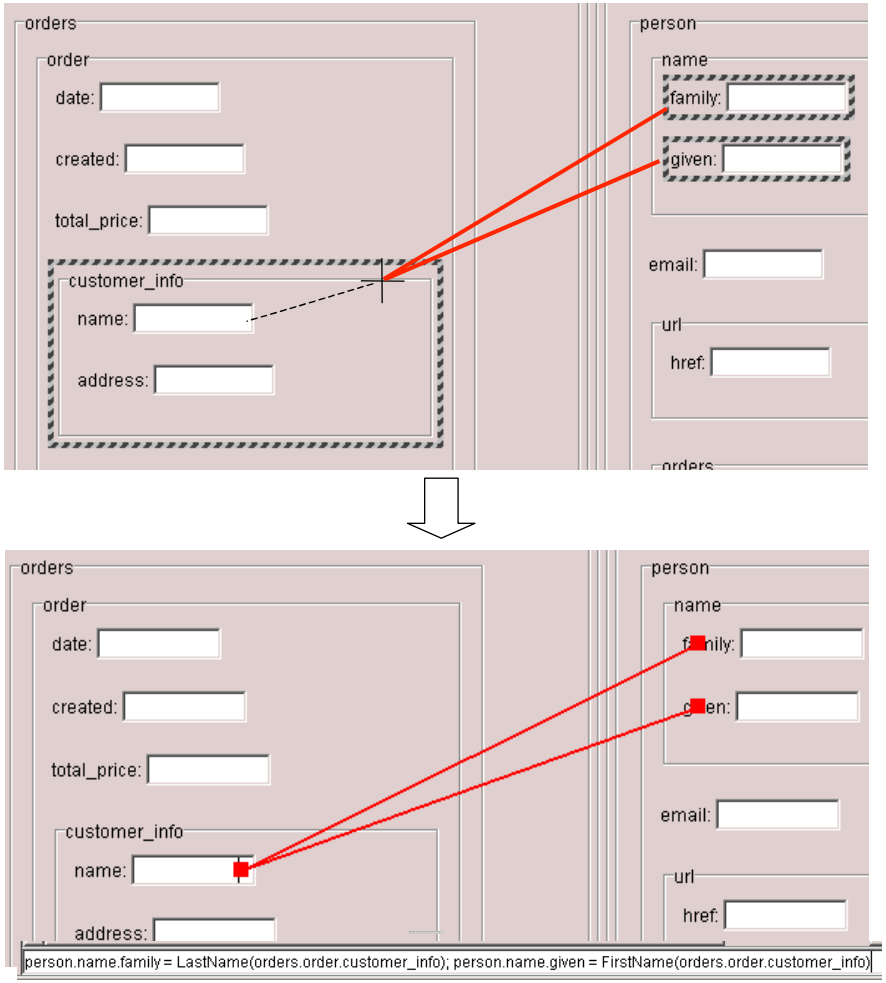


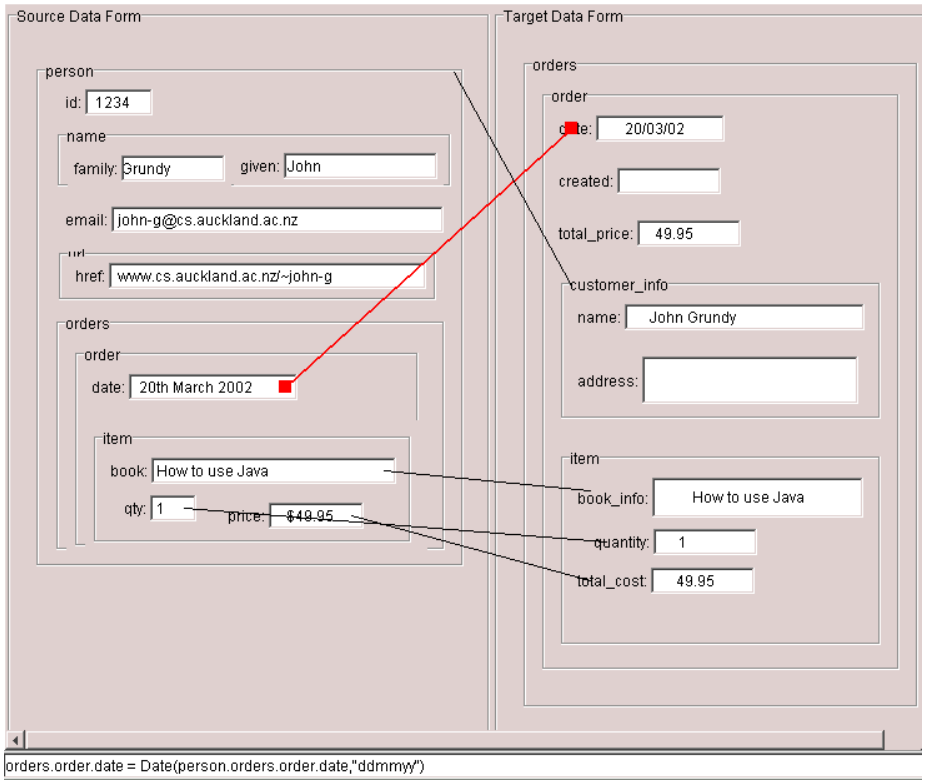
Resize

Add sub-structure

Rearrange layout

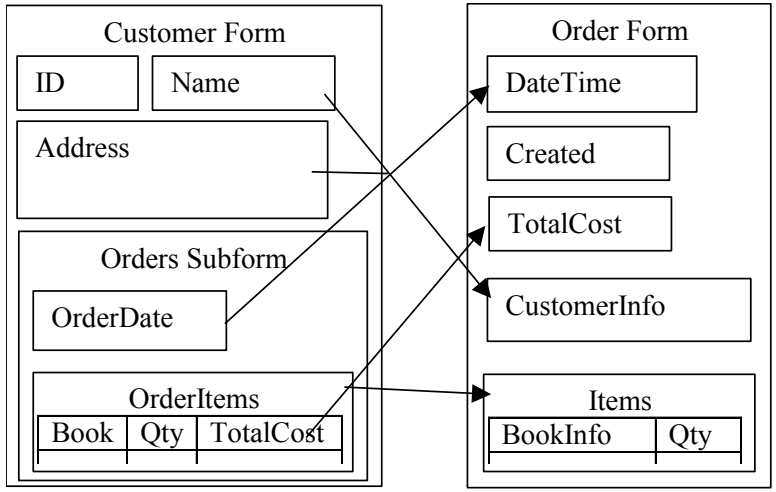
Specifying Mappings





THE UNIVERSITY OF AUCKLAND
www.auckland.ac.nz

Code Generation…



Order:
1
2
3
4

XSLT transformation script generation

```
<xsl:template match="/">
 <Order>
  <Number>…</Number>
  <DateTime><xsl:value-of select="/Order[1]/Order/Date"/>
    </DateTime>
  <Created>
    <xsl:value-of select="date:to-string(date:new())"/>
  </Created>
  <TotalCost><xsl:value-of
    select="sum(//OrderItem/TotalCost)"/> </TotalCost>
  <xsl:variable name="customer_id" select=
    "/Order/OrderItem[1]/CustomerSID"/>
  <CustomerInfo>
    <xsl:apply-templates select="//Customer [@id =
    $customer_id]"/>
  </CustomerInfo>
  <Items>
   <xsl:apply-templates select="//OrderItem"/>
  </Items>
 </Order>
</xsl:template>
…
```

Future Work

THE UNIVERSITY OF AUCKLAND
www.auckland.ac.nz

- **Lots of kinds of mappings – need to distinguish e.g.**
  - Record -> record
  - Field -> multiple fields
  - Fields -> one field
  - Records -> select to one record
  - Multiple records -> multiple records
- **Programming by example specification of these plus field splitting/merging**
- **"Sub-form"-based mappings for "functions"**
- **Visual formulae for target field/record values**

Summary

THE UNIVERSITY OF AUCKLAND
www.auckland.ac.nz

- Data mapping systems required in many problem domains
- Current approaches very programmatic
- Want to support business analysts specifying complex data mappings
- Used form-based copying metaphor within a prototype environment to do this
- Automatic mapping code generation
- Extending in various ways to improve usability/visualisation of complex mappings

References

THE UNIVERSITY OF AUCKLAND
www.auckland.ac.nz

- 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.Huh, J., Grundy, J.C., Hosking, J.G., Li, N., Amor, R., Integrated data mapping for a software meta-tool, In Proceedings of the 2009 Australian Software Engineering Conference, Gold Coast, Australia, April 2009, 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,
- Bossung, S., Stoeckle, H., Grundy, J.C., Amor, R. and Hosking, J.G. Automated Data Mapping Specification via Schema Heuristics and User Interaction, In Proceedings of the 2004 IEEE International Conference on Automated Software Engineering, Linz, Austria, September 20-24, IEEE CS Press, pp. 208-217.
- Mugridge, W., Grundy, J.C.., Hosking, J., Bryant, D., Supporting information mapping in Health Informatics via integrated message transformation, In Proceedings of the 2002 Health Informatics New Zealand Conference, 8-10 August 2002, Auckland, New Zealand.
- 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.
- Amor, R., Augenbroe, G., Hosking, J.G., Rombouts, W., Grundy, J.C., Directions in Modelling Environments, Automation in Construction, Vol. 4 (1995), Elsevier Science Publishers, 173-187.