



THE UNIVERSITY OF AUCKLAND

[www.auckland.ac.nz](http://www.auckland.ac.nz)

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

**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](http://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



THE UNIVERSITY OF AUCKLAND

[www.auckland.ac.nz](http://www.auckland.ac.nz)

- **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" (programmatically)

# Data Mapping Example



```
A:\hcc_mesg1_2.xml - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History
Address A:\hcc_mesg1_2.xml
M:1 record
- <PatientMessage>
- <PatientRecord>
  <IDField>1000</IDField>
  - <PatientNameRecord>
    <LnameField>Grundy</LnameField>
    <FnameField>John</FnameField>
  </PatientNameRecord>
  - <PatientDOBRecord>
  - <DateRecord>
    <DayField>10</DayField>
    <MonthField>2</MonthField>
    <YearField>1966</YearField>
  </DateRecord>
  </PatientDOBRecord>
  - <PatientAddressRecord>
    <StreetField>10 Norton Road</StreetField>
    <CityField>Auckland</CityField>
    <CountryField>New Zealand</CountryField>
  </PatientAddressRecord>
  </PatientRecord>
- <PatientVisitsSegment>
- <VisitRecord>
  - <VisitInfoRecord>
    <VisitIDField>100011</VisitIDField>
    <VisitDateField>2001 03 20</VisitDateField>
    <VisitLocationField>Hospital # 1</VisitLocationField>
  </VisitInfoRecord>
  - <TreatmentSegment>
  - <TreatmentRecord>
    <TreatmentCodeField>BC1</TreatmentCodeField>
    <TreatmentTypeField>P</TreatmentTypeField>
    <TreatmentDateRecord>
```

```
A:\hcc_mesg2_2.xml - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History
Address A:\hcc_mesg2_2.xml
- <PVisitMessage>
  <PIDField>8791</PIDField>
  <MedRecNumField>1000</MedRecNumField>
  <PnameField>Grundy John</PnameField>
  <DateOfBirthField>10 7 1972</DateOfBirthField>
  <PaddressField>10 Norton Rd Auckland</PaddressField>
  - <VisitSegment>
    <VisitCodeField>BT</VisitCodeField>
    <VisitDateField>1966 2 10</VisitDateField>
  - <AttendingDoctorSegment>
    - <DoctorRecord>
      <LicenseField>1234</LicenseField>
      <NameField>John Hosking</NameField>
    </DoctorRecord>
  </AttendingDoctorSegment>
  - <ResponsibleDoctorSegment>
    - <DoctorRecord>
      <LicenseField>4567</LicenseField>
      <NameField>Rick Muiridge</NameField>
    </DoctorRecord>
  </ResponsibleDoctorSegment>
  </VisitSegment>
  </PVisitMessage>
- <PrimaryTreatmentSegment>
- <TreatmentRecord>
  <TreatDateField>2001 3 20</TreatDateField>
  <TreatKindField>Blood Clot # 1</TreatKindField>
  <CostingField>NT</CostingField>
  <CostField>167.85</CostField>
  <TaxField>18.65</TaxField>
</TreatmentRecord>
- <TreatmentRecord>
  <TreatDateField>2001 3 20</TreatDateField>
  <TreatKindField>Blood Clot # 1</TreatKindField>
  <CostingField>NT</CostingField>
```

M:1 record

1:1 field

M:1 field

M:M records

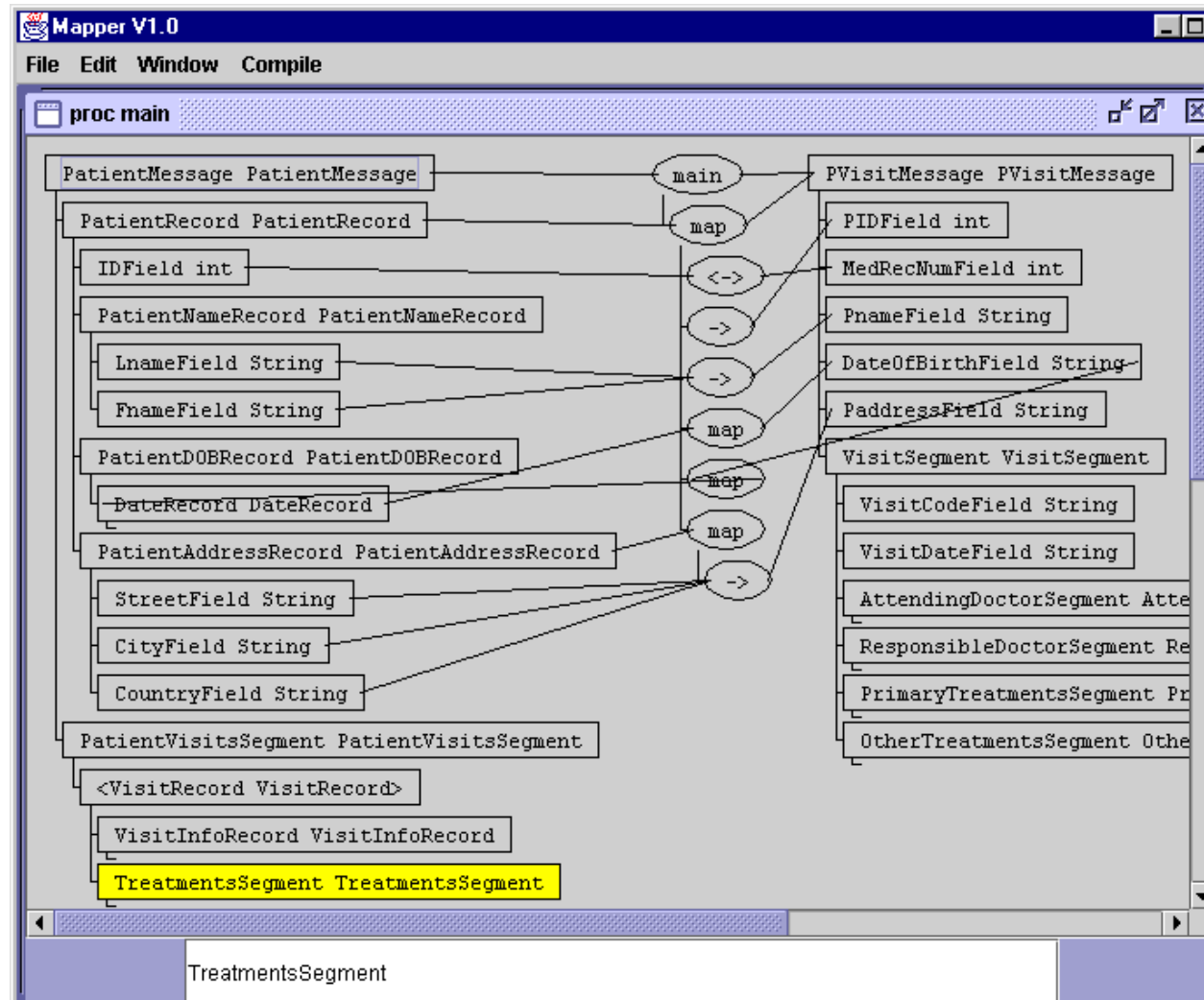
M:N record selection

# Existing Tool Example



THE UNIVERSITY OF AUCKLAND

[www.auckland.ac.nz](http://www.auckland.ac.nz)



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

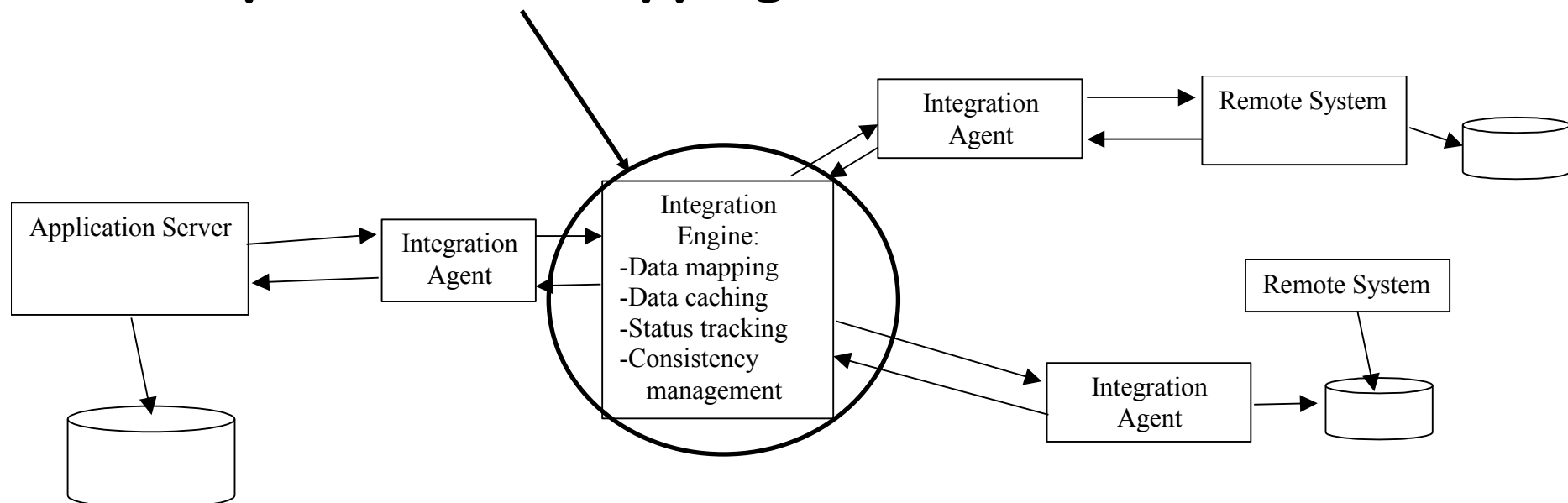
# Context for Our Work



THE UNIVERSITY OF AUCKLAND

[www.auckland.ac.nz](http://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](http://www.auckland.ac.nz)

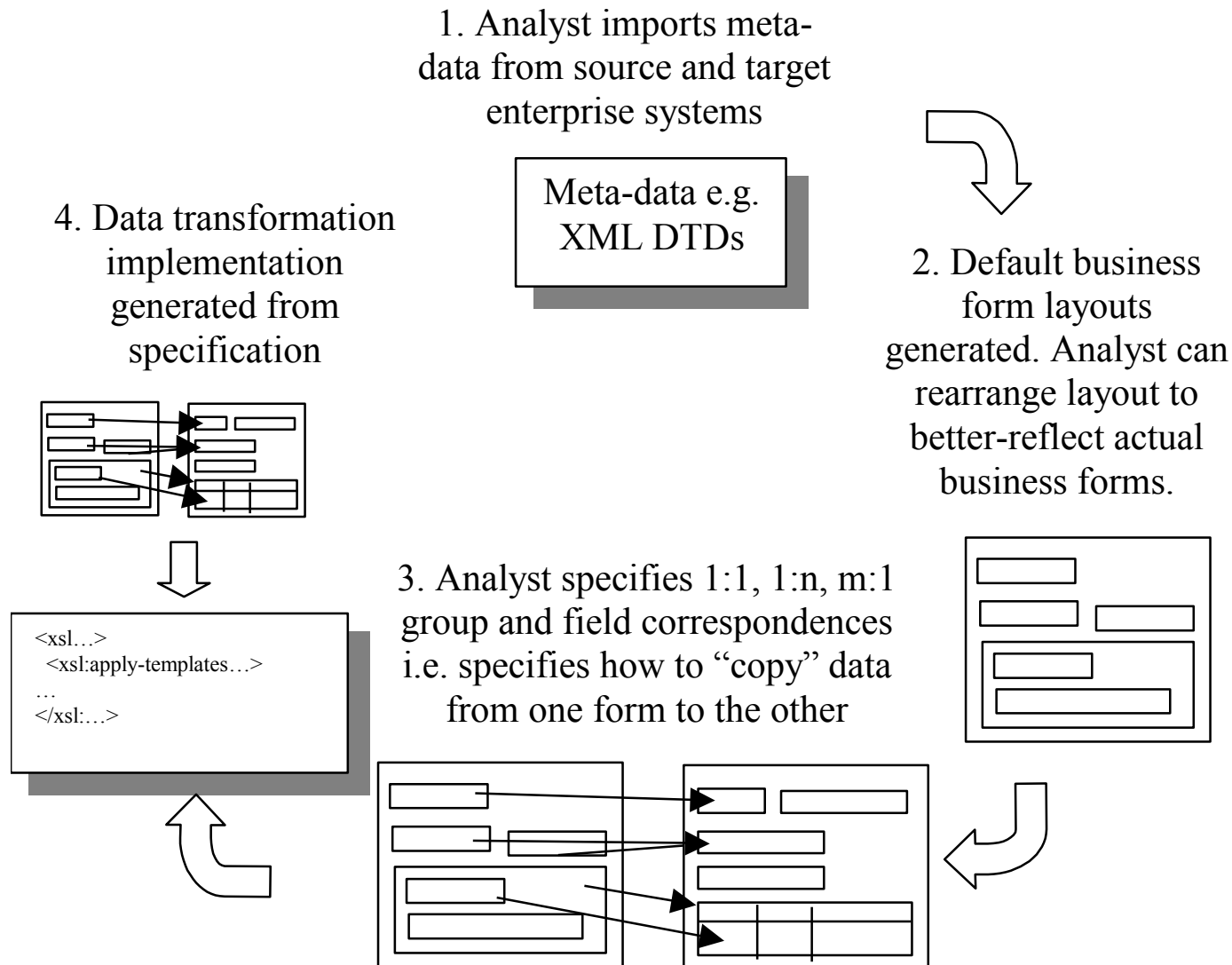
- Aim: to support end-user specification of complex mappings
- Users = “business analysts”
- Want to generate mapping implementations
- Most businesses manually implement such data exchange manually via “copying” from one business form to another (hard-copy or on a computer)
- We wanted to support this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

# Process



THE UNIVERSITY OF AUCKLAND

[www.auckland.ac.nz](http://www.auckland.ac.nz)





# Example of Form-based Visualisation of Documents



THE UNIVERSITY OF AUCKLAND  
[www.auckland.ac.nz](http://www.auckland.ac.nz)

The screenshot displays a software interface for mapping data between source and target documents. The interface is divided into several sections:

- Source Data Tree:** A hierarchical tree view showing the structure of the source document. The root is 'person', which contains 'name' (with sub-elements 'family' and 'given'), 'email', 'url', and 'orders'. The 'orders' element contains an 'order' element, which in turn contains 'date', 'item', and 'id'. The 'item' element contains 'book', 'qty', and 'price'.
- Source Data Form:** A visual representation of the source document's data. It shows a 'person' container with input fields for 'family:', 'given:', 'email:', and 'url:' (with a sub-field for 'href:'). Below this is an 'orders' container with an 'order' sub-container containing a 'date:' field, and an 'item' sub-container containing 'book:', 'qty:', and 'price:' fields.
- Target Data Tree:** A hierarchical tree view showing the structure of the target document. The root is 'orders', which contains 'order', 'customer\_info', and 'item'. The 'order' element contains 'date', 'created', and 'total\_price'. The 'customer\_info' element contains 'name' and 'address'. The 'item' element contains 'book\_info', 'quantity', and 'total\_cost'.
- Target Data Form:** A visual representation of the target document's data. It shows an 'orders' container with an 'order' sub-container containing 'date:', 'created:', and 'total\_price:' fields. Below this is a 'customer\_info' sub-container with 'name:' and 'address:' fields, and an 'item' sub-container with 'book\_info:', 'quantity:', and 'total\_cost:' fields.

# Rearranging Form Layout



THE UNIVERSITY OF AUCKLAND

[www.auckland.ac.nz](http://www.auckland.ac.nz)

Source Data Form

person

id:

name

family:  given:

email:

url

href:

orders

order

date:

item

book:

qty:  price:

Resize

Add sub-structure

Rearrange layout

# Specifying Mappings



THE UNIVERSITY OF AUCKLAND

www.auckland.ac.nz

orders

order

date:

created:

total\_price:

customer\_info

name:

address:

person

name

family:

given:

email:

url

href:



orders

order

date:

created:

total\_price:

customer\_info

name:

address:

person

name

family:

given:

email:

url

href:

```
person.name.family = LastName(orders.order.customer_info); person.name.given = FirstName(orders.order.customer_info)
```

Source Data Form

person

id: 1234

name

family: Grundy given: John

email: john-g@cs.auckland.ac.nz

url

href: www.cs.auckland.ac.nz/~john-g

orders

order

date: 20th March 2002

item

book: How to use Java

qty: 1 price: \$49.95

Target Data Form

orders

order

date: 20/03/02

created:

total\_price: 49.95

customer\_info

name: John Grundy

address:

item

book\_info: How to use Java

quantity: 1

total\_cost: 49.95

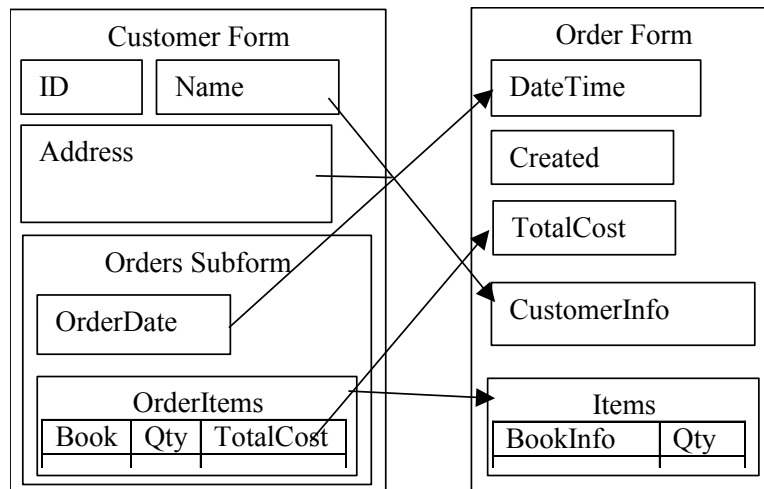
```
orders.order.date = Date(person.orders.order.date,"ddmmyy")
```

# Code Generation...



THE UNIVERSITY OF AUCKLAND

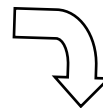
[www.auckland.ac.nz](http://www.auckland.ac.nz)



Order:

1

XSLT transformation  
script generation



2

3

4

```
<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](http://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](http://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](http://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.