

Generating EDI Message Translations from Visual Specifications



John Grundy, Rick Mugridge, John Hosking
University of Auckland, New Zealand

Paul Kendall
Orion Systems Ltd, Auckland, New Zealand

ASE 2001

Outline



- Motivation
- Our Approach
- Visual mapping specifications
- Mapping code generation and Engine
- Visualisation of in-progress mappings
- Experience
- Future Work
- Conclusions

Motivation



□ Message mapping:

- EDI systems
- XML-based systems
- Message-oriented Middleware systems

□ Mapping messages is HARD!

□ Approaches:

- Hard-coded in C++, Java etc
- Scripting e.g. XSLT
- Mapping tools e.g. MQ Integrator™, BizTalk™

Health XML 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 Go Links
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 Go Links
- <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 Mugridge</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

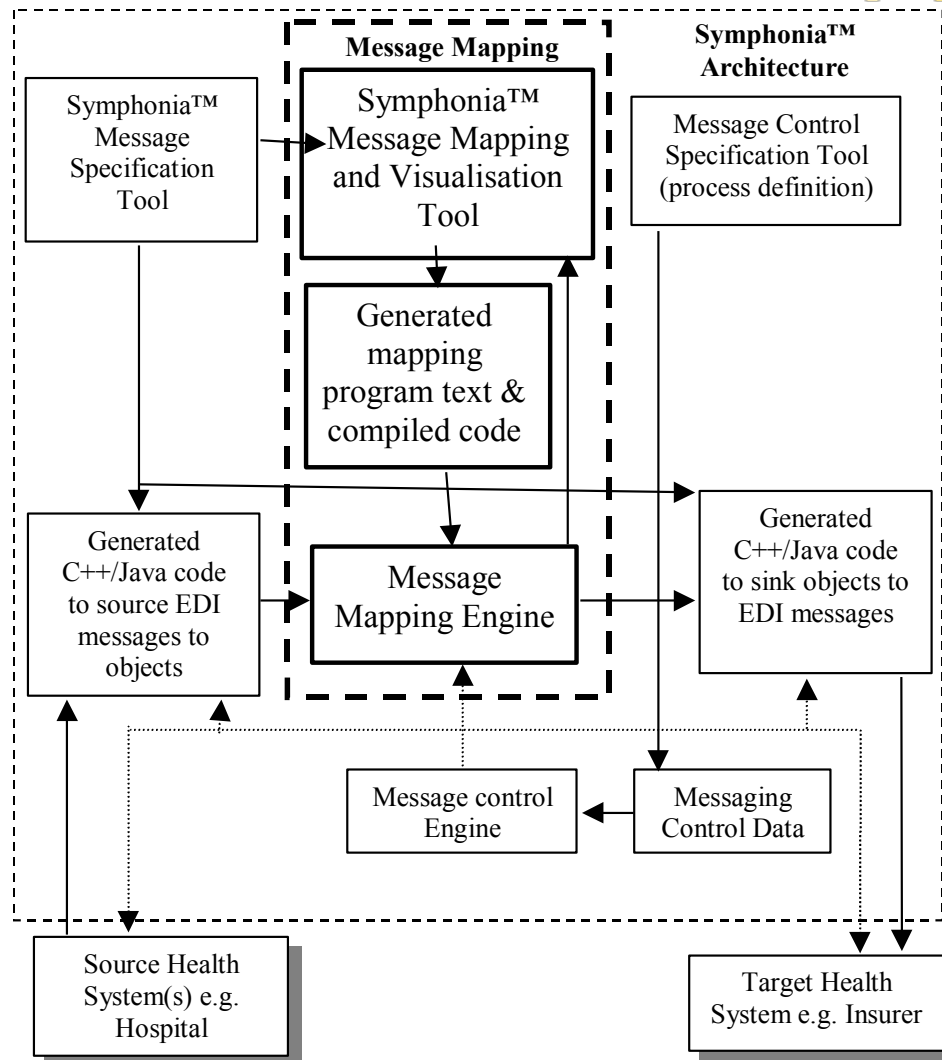
M:M records

M:1 field

M:N record selection

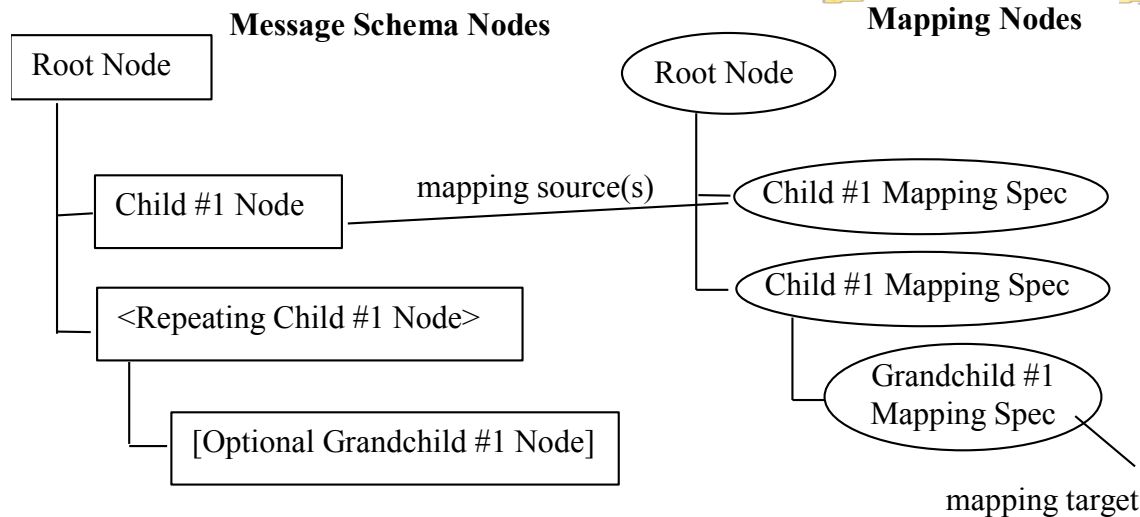


Our Approach



- ❑ Tool to source/sink EDI messages (generates C++/Java code)
- ❑ Tool to co-ordinate message exchange (via workflow descriptions)
- ❑ **Tool to specify visually message mappings - generates mapping engine specification**
- ❑ Mapping engine to perform complex message mappings

Mapping Language

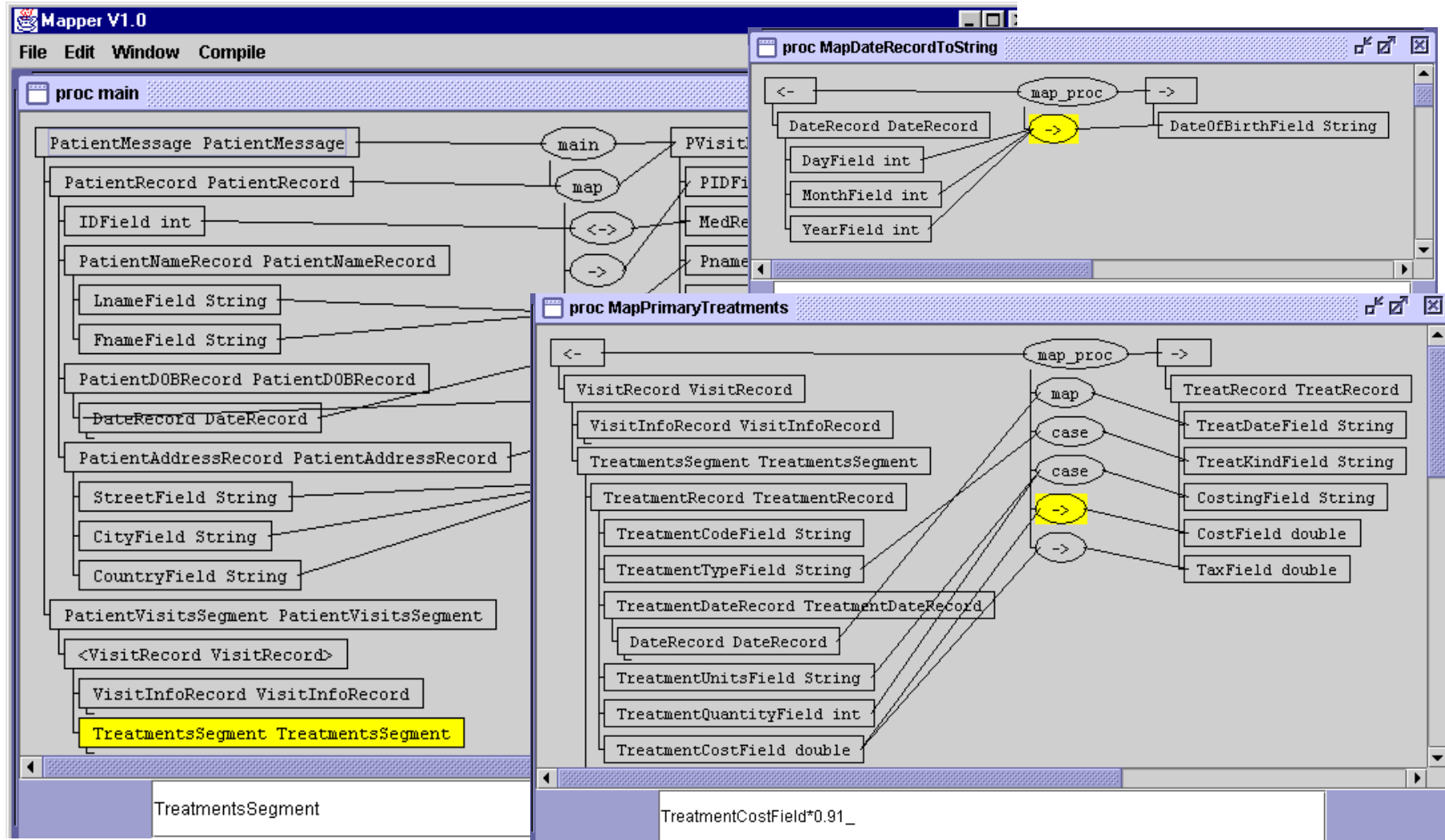


Mapping Node Types

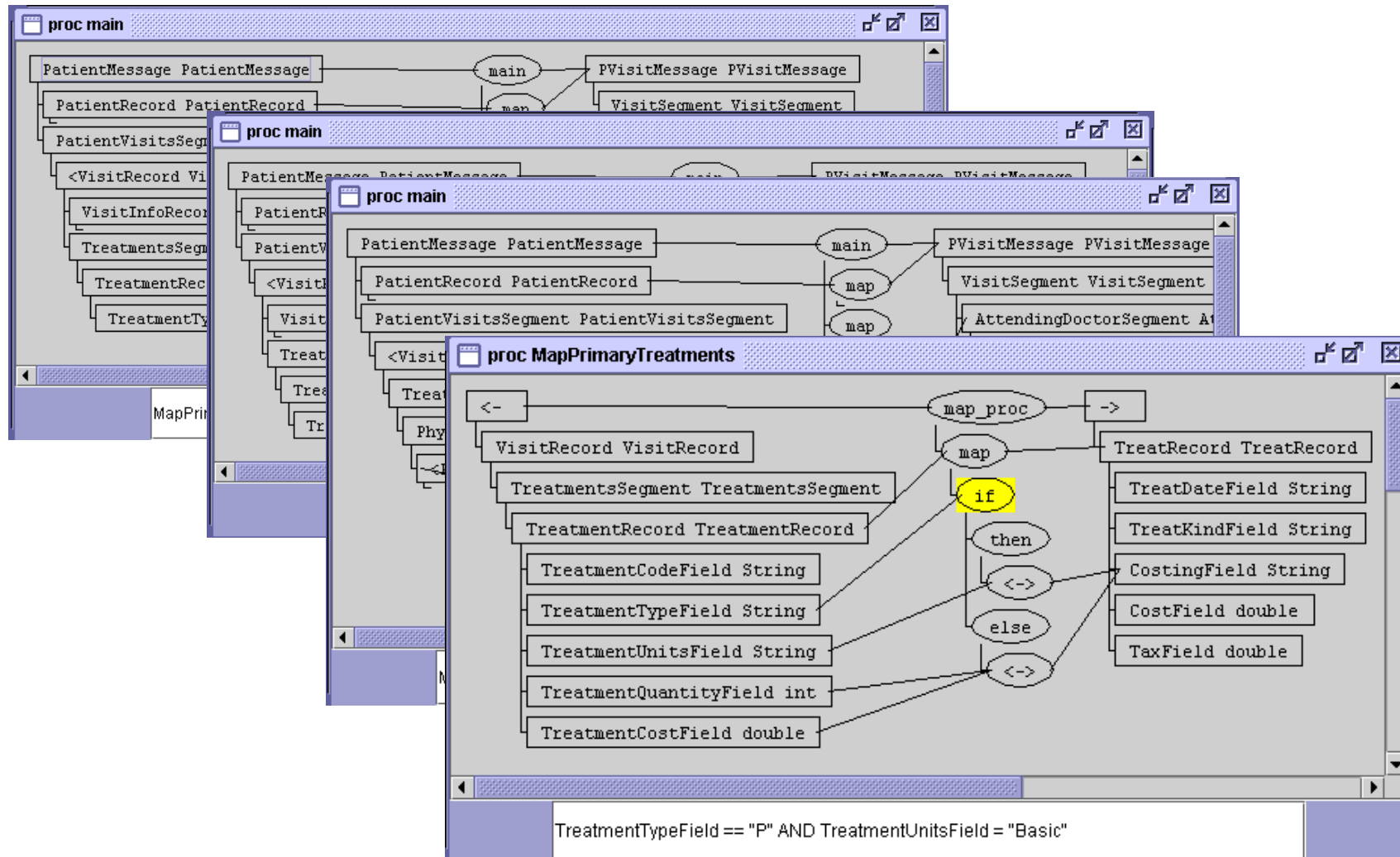
main	Root mapping function	<->	Bi-directional copy
map	Mapping function/group	->	Uni-directional formula
map_proc	Functional definition	if/then/else	Conditional mapping
arg	Function argument formula	case	Table look-up (formula)
select	Function argument filter		

- ❑ Schema (message) and mapping nodes
- ❑ Schema = messages, segments, records, fields, repeating/optional elements etc
- ❑ Hierarchical organisation
- ❑ Mappings run over source message and generate target message elements

Simple Example in Mapper Tool



Complex Example...

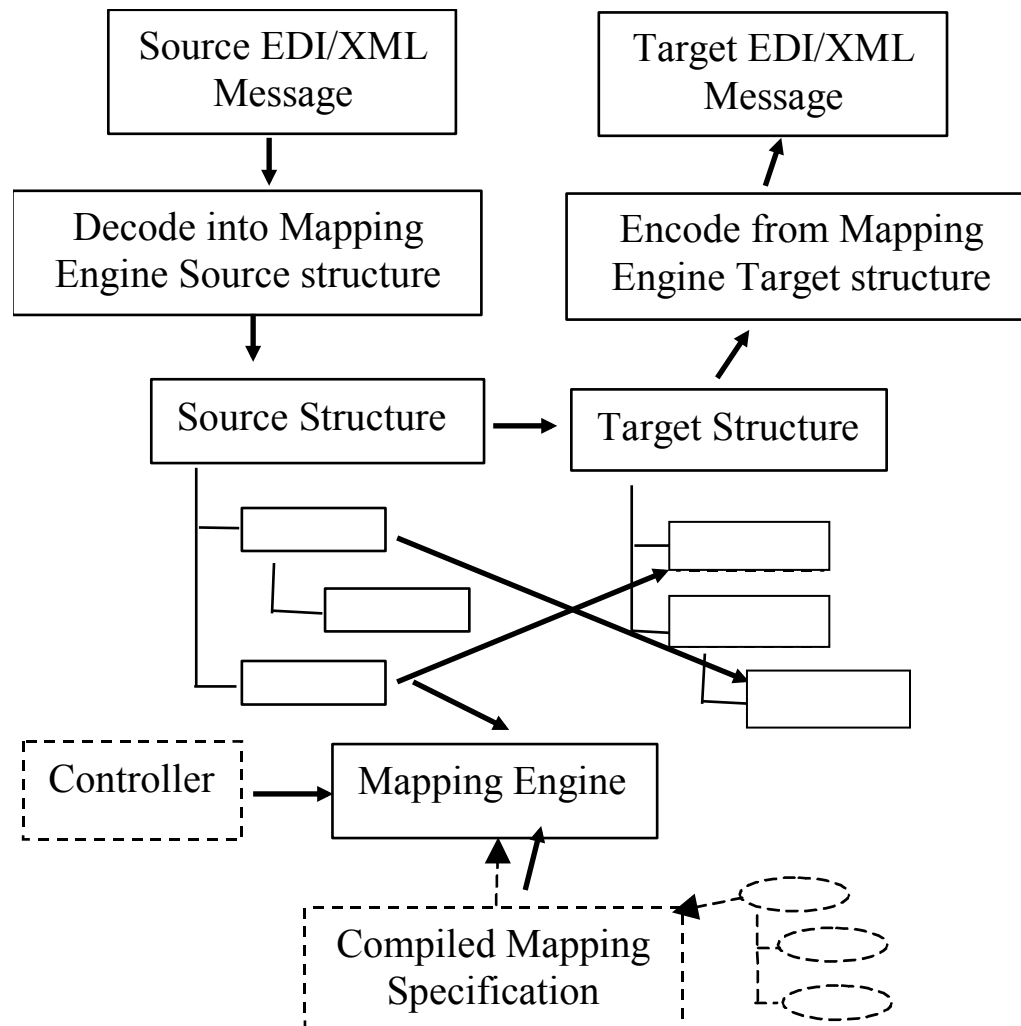


Mapping Language

```
type PatientMessage = struct {
  PatientRecord PatientRecord;
  PatientVisitsSegment PatientVisitsSegment;
};
type PatientRecord = struct {
  int IDField;
  PatientNameRecord PatientNameRecord;
  PatientDOBRecord PatientDOBRecord;
  PatientAddressRecord PatientAddressRecord;
};
...
map main(<- PatientMessage PatientMessage,
-> PVisitMessage PVisitMessage)
{
  PatientMessage.PatientRecord.IDField <->
    PVisitMessage.MedRecNumField;
  ExternalGeneratePatientID(, PVisitMessage.PIDField);
  Concat (PatientMessage.PatientRecord.PatientNameRecord.LnameField,
    ' ', ...);
  DOBRecordToDateOfBirth (PatientMessage.PatientRecord.
    PatientDOBRecord, PVisitMessage.DateOfBirthField);
  MapTreatmentRecordToTreatRecord(select (I from
    in.PatientVisitsSegment.VisitRecord.TreatmentsSegment.
    TreatmentRecord[*] where I.TreatmentSegment ... ));
...
}
```

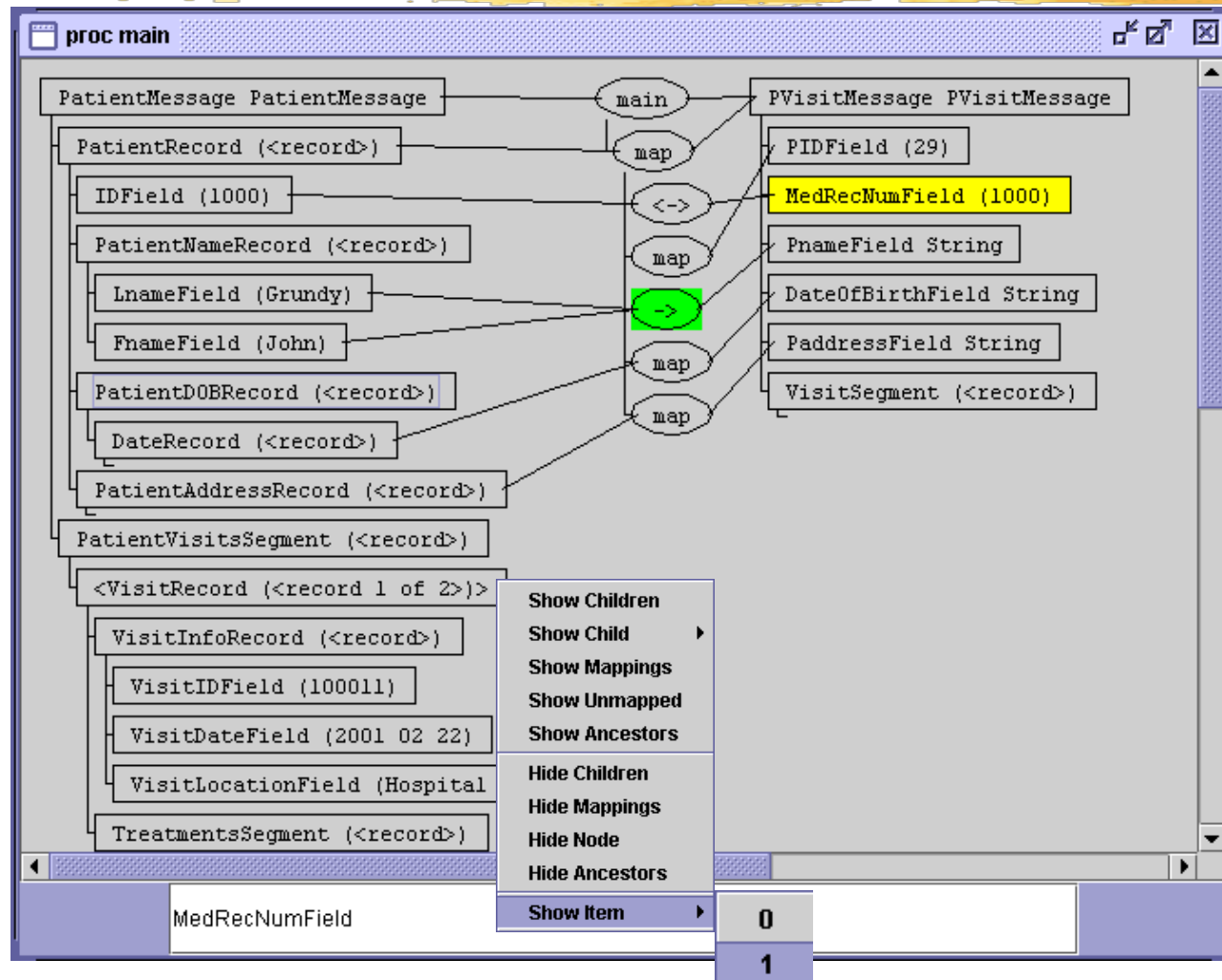
- ❑ Visual specifications + formulae used to generate a domain-specific textual language
- ❑ This includes record manipulation constructs, functional language constructs, source/target message expressions etc
- ❑ User DOES NOT edit this code - compiled and used by mapping engine...

Mapping Engine



- ❑ Message co-ordinator (controller) invokes mapping engine when needs to translate EDI (or XML) messages
- ❑ Compiled message mapping specification used to transform source->target message
- ❑ New message converted into target domain & used by controller...

Run-time Visualisation...



Experience...



- ❑ Prototype = Java 1.2/JAXP/Orion EDI APIs
- ❑ Used to specify bunch of different health EDI and business EDI and XML message mappings (HL7, UB92, 837a, EDIFACT, SOAP(ebXML), ...)
- ❑ Example: 3 month Java UB92<->837a mapper vs. 3 day mapping system specified map between these
- ❑ Good performance - ~30,000 complex messages/min...
- ❑ Usability evaluation of visual tool - good feedback on usefulness of visual language + tool
- ❑ Orion Systems Ltd developed commercial version of IDE/text language/engine... (using C++, MFC)

Future Work/Conclusions



- ❑ Richer visual presentation of structures/mappings
- ❑ Message->database; database->message mappings
- ❑ Further visual metaphors for non-programmers

- ❑ Can automatically map complex EDI and XML messages using high-level visual specifications
- ❑ Visual specification useful at specification time and to debug mappings (dynamic visualisation)
- ❑ Commercialisation of basic research successful

References



- 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.
- Grundy, J.C., Mugridge, W.B., Hosking, J.G. and Kendal, P., A Visual Language and Environment for EDI Message Translation, In Proceedings of Human-Centric Computing 2001, IEEE CS Press.
- Grundy, J.C., Mugridge, W.B. and Hosking, J.G. Constructing component-based software engineering environments: issues and experiences, Information and Software Technology Vol 42, No. 2, Special Issue on Constructing Software Engineering Tools, Elsevier Science Publishers.
- Grundy, J.C. Construction of an Integrated and Extensible Software Architecture Modelling Environment, In Proceedings of the 2nd International Symposium on Constructing Software Engineering Tools (CoSET'2000), Limerick, Ireland, pp. 51-61.
- Grundy, J.C. Distributed Component Engineering using a Decentralised, Internet-based Environment, In Proceedings of the 3rd ICSE Workshop on Software Engineering over the Internet, ICSE 2000 Workshop, Limerick, Ireland, June 6 2000, pp. 20-29.
- Grundy, J.C. Visual specification and monitoring of software agents in decentralised process-centred environments, International Journal on Software Engineering and Knowledge Engineering, Vol. 9, No. 4., August 1999, World Scientific Publishing Company, pp. 425-444.
- Grundy, J.C., Mugridge, W.B., Hosking J.G. Supporting Large-scale End-user specification of workflows, work coordination and tool integration, Journal of End-User Computing, Vol. 10, No. 2, May 1998, Idea Group Publishing, pp. 39-49.
- Grundy, J.C. and Hosking, J.G. Serendipity: integrated environment support for process modelling, enactment and work coordination, Automated Software Engineering: Special Issue on Process Technology, Vol. 5, No. 1, January 1998, Kluwer Academic Publishers, pp. 27-60.