A Visual Language and Environment for EDI Message Translation

John Grundy[1], Rick Mugridge[1], John Hosking[1] and Paul Kendall[2]

[1]Department of Computer Science, University of Auckland
Private Bag 92019, Auckland, New Zealand
{john-g,rick,john}@cs.auckland.ac.nz

[2]Orion Systems Ltd
Mt Eden, Auckland, New Zealand
paul@orion.co.nz

Abstract

Electronic Data Interchange (EDI) systems are used in many domains to support inter-organisational information exchange. These systems require complex message translation, where data must be transformed from one EDI message format into another. We describe a visual language and support environment which greatly simplify the task of the systems integrator by using a domain-specific visual language to express translations.

1. Introduction

Electronic Data Interchange (EDI)-based systems exchange messages that codify organisational information [1, 3]. For example, in Health Informatics a treatment provider's Information System might describe a patient, hospital visit information, patient treatment and treatment costs [2]. It provides a set of EDI messages that are used to add, update and query this information. A health insurer or funding organisation requires this information to record the treatments, costs and reimbursements, but it uses a different set of EDI messages and information structures. In order to support information exchange between, the provider must supply the insurer/funder with its expected message format. Similarly, data sent back to the provider from the funder must be appropriately converted. Writing code to do such transformations is time-consuming, error-prone and difficult to maintain. We have developed a visual language environment to automate such EDI (and XML) message translation.

2. Message Mapping Specification

Our visual mapping tool provides a domain-specific visual language for representing message mappings. This includes hierarchical source and target message structure representation and mappings between source and target segments, records, fields and collections. A source message schema is placed on the left, target on the right, using rectangles ("nodes") to represent each message, segment, record and field item in the message "schema". Child nodes are linked to their owning parents. Ovals are mappings between nodes in a central hierarchy.

Figure 1 (1) shows example message schema and mapping specifications. The "main" mapping node (a) groups all mappings from one EDI message to another. The first child mapping node (b) groups mappings specifying how to copy PatientMessage information to a PVisitMessage's patient fields. The first node (c) specifies that MedRecNumField is copied from IDField, and is a bi-directional mapping (<->) i.e. can be applied the same when mapping in either direction. The PIDField value is defaulted (auto-generated) by an external function call (d). The PNameField value is a merge of the PatientNameRecord's LNameField and FnameField values (e). The DateRecord fields are merged into one DateOfBirthField value by a local mapping function call (f). Another reusable mapping function call says what to do when translating in the reverse direction (g), in this case the DateOfBirthField will have to be parsed and split to obtain the separated DOB record's field values. Figure 1 (2) shows a reusable date mapping.

More complex message transformations involve transformations of groups of records into similar or different structures and the use of conditional logic. Figure 1 (3) shows mapping the first source Physician record into the target AttendingDoctor record. The node (c) has value "[0]", indexing the first source PhysicianRecord. Figure 1 (4) shows selective mappings, in this case different target costing information calculated.

3. Mapping Visualisation

Our visual mapping tool can be used to visualise in-progress message mappings and to step through mappings as they are applied. Figure 2 shows it in use during message mapping debugging.
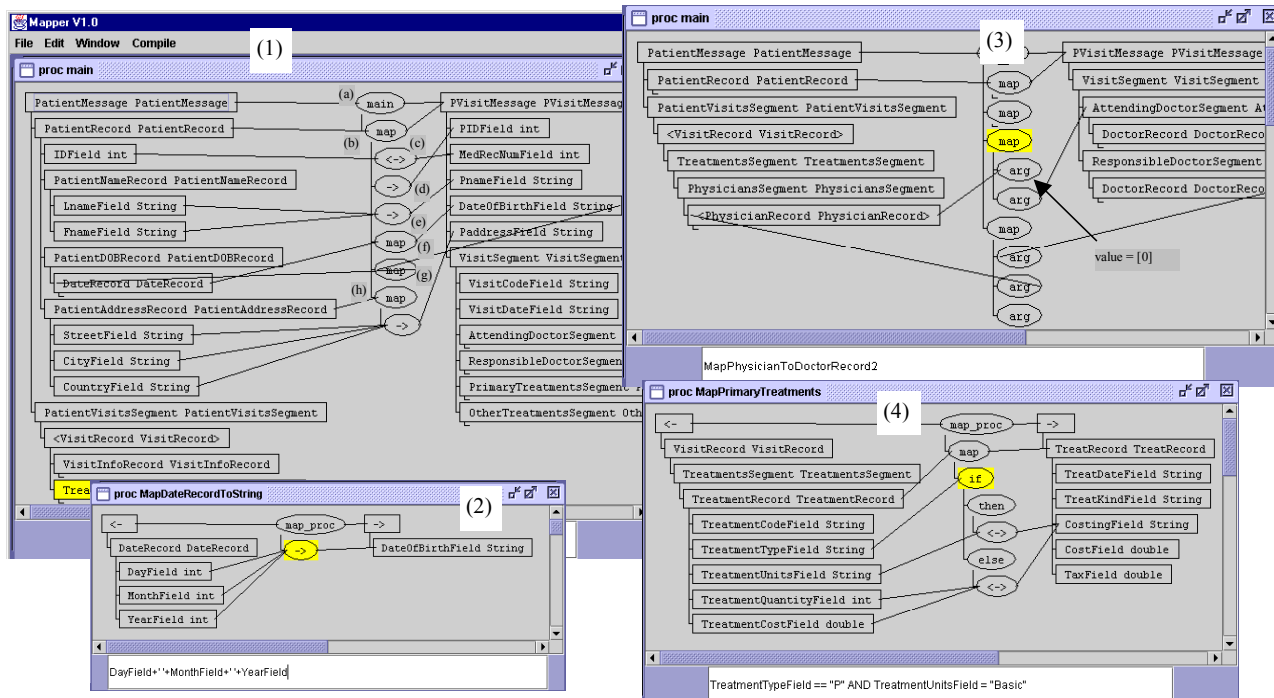
Figure 1. Some simple (1, 2) and complex (3, 4) EDI message mapping examples.
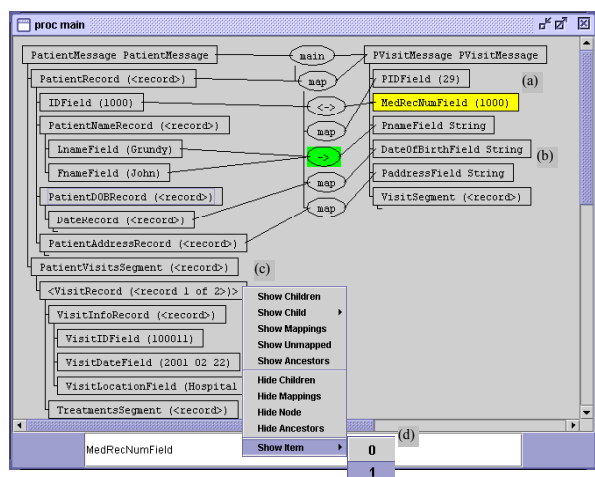

Figure 2. Run-time visualising partially mapped data.

The mapping is shown part-way through, with some target fields have values e.g. patient information (a) while others have not been assigned values yet (b). Groups of records can be displayed (c), one group item at a time by the use of pop-up menus associated with schema nodes (d). An example is shown of the user viewing the first source VisitRecord.

4. Summary

We have developed a message translation system for EDI-based and XML-based applications, which has been applied to and shown its worth in the health informatics domain. For example, a hand-coded UB92 to 837 health EDI message mapping program took over 3 months to build, whereas a visual mapper-specified and implemented equivalent took less than a week. Key contributions are its intuitive visual representation of inter-schema message mappings, full integration with a powerful textual mapping language, an efficient mapping engine and ability to dynamically visualise, using the visual mapping views, running message mappings. A commercial product developed from our proof-of-concept system by Orion Systems Ltd (www.orion.co.nz) and wide interest in this product is being shown in the health systems integration industry.

Acknowledgments

References

[1] Emmelhainz, M.A. *Electronic Data Interchang*, Van Nostrand Rein-hold; New York; 1990.

[2] McLure, M.L., Moynihan, J.J. Organizing for EDI (healthcare industry). *Healthcare Financial Management*, vol.49, no.1, Jan. 1995, pp.90-93.

[3] Swatman, P.M.C., Swatman, P.A., Fowler, D.C. A model of EDI integration and strategic business reengineering. *Journal of Strategic Information Systems*, vol.3, no.1, March, 1994, pp.41-60.