Experience in developing interoperations among legacy information systems using partial reverse engineering

Michael Johnson
Department of Computing
Macquarie University
Sydney, Australia
Email: mike@ics.mq.edu.au

C.N.G. Dampney
School of Information Technology
University of Newcastle
Australia
Email: Kit.Dampney@newcastle.edu.au

Abstract

At ICSM01 Johnson and Rosebrugh proposed a new methodology, based on formal specification techniques and partial reverse engineering, for developing interoperations among legacy information systems. The present brief paper reports on progress to date in designing such interoperations for major health informatics information systems. We describe briefly the methodology, argue for the value of partial reverse engineering and the benefits of the formal specification approach (based on category theory) that we use, and evaluate the methodology in two case studies. The main new result is the following. The ICSM01 paper correctly predicted that what was called there "full-duplex interoperation" would often provide more interoperability than is in fact required. We have discovered that the provision of full-duplex interoperability is indeed usually unnecessary, and it is costly because of the need to modify the legacy system. We proposed the development of "half-duplex interoperation" to avoid these costs, and we have since developed the half-duplex techniques. Nevertheless, the half-duplex techniques can still require some legacy system modification. Following recent work on partial information, we are developing an extension to the view based interoperation technique that avoids legacy system modification in over 92% of the cases we have investigated.
Keywords: Category theory, databases, formal methods, reverse engineering, interoperations, semantic data modelling

1 Introduction

One of the major software maintenance problems confronted by medium to large enterprises is the need to develop interoperations between large legacy systems. Indeed, the problems are so great that very often the organisations choose to forgo the benefits of interoperations until the legacy systems are retired and replaced by new integrated systems. This is costly since the savings from successful interoperation can be great, and since the costs of customising new systems to match the organisation's processes, or of modifying organisational processes to match new systems, can be very great.

So, perhaps we should say instead that a major *opportunity* for the effective utilisation of software maintenance techniques lies in providing an effective methodology to develop interoperations between legacy systems. At ICSM'01 Johnson and Rosebrugh [10] proposed a methodology based on category theoretic specification techniques (see [8], [9] and for related work [1], [6], [7], [16], [17], [14]) which showed considerable promise for developing interoperations with only limited modifications of the legacy systems. This paper reports our experience to date in using the JR-methodology to design interoperations for large health informatics information systems.

The paper is arranged as follows. In Section 2 we briefly review the JR-methodology as presented in [10]. In Section 3 we explain what we mean by *partial reverse engineering* and give some indications as to why the category theoretic techniques are so effective in supporting the partial reverse engineering of legacy information systems. Section 4 describes *full-duplex* and *half-duplex* interoperations, and explains why half-duplex interoperations save significant development costs. Section 5 reports the results of the first of two case studies designed to evaluate the methodology and based on our consultancy work for a state Department of Health. Finally, Section 6 describes how new work on *partial information* can extend the JR-methodology and, in cases where the legacy systems support partial information, interoperations can usually be achieved with *no* legacy system modification. In the second case study we measured the frequency of this happy outcome and identified under what circumstances it failed.

2 The JR-Methodology

In short, the JR-methodology is as follows. Given two (or more) legacy information systems for which we wish to design interoperations, we use category theoretic specification techniques to develop logically data independent models of (parts of) the legacy information systems. These models are formal specifications of reverse engineered relevant subsystems. We then develop a single logically data independent model of data derivable from each of the reverse engineered subsystems. This model, $\mathbb{K}$, is the locus of interoperation. It is a single formal representation of those data from the original systems that will be involved in the interoperations. It remains to develop the interoperation code.

To develop the interoperation code we need to explore the *propagatability* of update information from each legacy system, "out" to the common submodel $\mathbb{K}$ and back "in" to the other information system(s). It turns out that the outward propagatability (from the legacy system to $\mathbb{K}$) is guaranteed, first by the construction of the reverse engineered model, and then for formal mathematical reasons from the reverse engineered model to $\mathbb{K}$. Inward propagatability on the other hand *cannot* be guaranteed. Instead, inward propagatability needs to be explored in two steps. First from $\mathbb{K}$ to the reverse engineered submodel we use mathematical methods (since both models are formally specified), and a catalogue of some of these methods has been published in [9]. Then ad hoc checking techniques are used to test inward propagatability from the reverse engineered submodel to the legacy system (in fact these checks are typically part of the reverse engineering process we use, and there is no need to implement code for this part of the inward propagation since the legacy system interface is used to carry it out).

More detail can be found in [10] and in [12]. The important things to note are that

1. Mathematical methods are used to determine when interoperations are possible
2. When interoperations are possible the legacy systems are accessed via their own extant interfaces and the mathematical techniques yield the interoperations code that is needed to link the systems
3. When interoperations are not possible, the mathematical techniques identify the locus of the problem, and more traditional software maintenance techniques can be used to modify one or more of the legacy systems to make interoperations possible.

The next two sections expand upon the partial reverse engineering techniques, and the generality of the interoperations provided, before we report the case study results.

3 Partial Reverse Engineering and Category Theory

Since "partial reverse engineering" plays an important role in the JR-methodology, and since reverse engineering is notoriously difficult, it is important that we explain in more detail what is required.

First, there is no need to reverse engineer significant systems. We use the term *partial* reverse engineering to emphasise that only very limited amounts of the legacy systems need to be analysed in detail. Generally a legacy system will have very significant functionality, but only a very small amount of the system will relate to the storage of data which needs to be linked with other systems — it is that small amount of the system that we need to analyse. We will refer to the relevant small part of the legacy system as the *relevant subsystem*.

Secondly, and quite importantly, we require only a logically data independent reverse engineered model — we seek a formal specification of the relevant subsystem which has approximately equivalent data expressivity, but we are not concerned to model the actual data structures used in the legacy system. Indeed, there is no need to understand the internal data structures because they will be manipulated by the legacy system's own interface. What is important is that we determine a logical model of them so that the manipulations that we require during interoperations are indeed among those available through the interface, and so that manipulations that may at various times take place within the relevant subsystem can be represented in our logical model, and can in turn be transmitted to other interoperating legacy systems.

Category theory ([2], [15]) is a branch of mathematics renowned for the simplicity of its axioms and the semantic expressiveness it provides. We have been using category theoretic data modelling over a number of years in large scale information system consultancies [18], [5], [4]. Some of the techniques that we use have been described in [8], [3] and [11]. The JR-methodology builds logical models of relevant subsystems by constructing category theoretic data models [10]. Interestingly the goal of modelling a relevant subsystem using category theoretic data models corresponds closely to our experience in developing information system specifications using category theory. In both cases we probe an extant system (the legacy information system, or the extant business system) to determine enough about the logical relationships among data items (but not necessarily the actual data structures used) to be able to design a formal specification.

One of the great advantages of a category theoretic data model is that it supports much greater semantic data modelling than more traditional methods. This is important in these applications because the mathematical methods used to test for interoperability [9] make use of the extra semantic structure (because they require functors which preserve finite limits and finite coproducts [15], [2]).

4 Full- and Half-Duplex Interoperations

The JR-methodology, as described in [10], proposed interoperations among equal cooperating information systems and sought to develop loci of interoperations, $\mathbb{K}$, which would always support interoperations in either direction. In analogy with data communications, such interoperations were called full-duplex.

Full-duplex interoperations are expensive because Proposition 14 of [9] shows that they depend upon *attribute harmonisation* — entities in $\mathbb{K}$ must include all of the attributes of the corresponding parts of both legacy systems and in turn for propagatability to succeed both legacy systems must include all of those attributes. In our experience this is rarely the case, and so we would expect full-duplex interoperability to usually require legacy system modification.

We have since developed a detailed notion of half-duplex interoperation in which updates can occur in either direction, but for a particular type of update at a particular entity it may only be possible to interoperate in one direction. One of the major reasons for carrying out the case study reported in the next section was to test whether indeed fully duplex interoperations did require legacy system modification, and to what extent half-duplex interoperation may suffice and may ameliorate the need for legacy system modification.

5 Application to Health Informatics

We evaluated the JR-methodology by exploring its application to the data models held by the NSW Department of Health. In brief our findings were that

- The size of the relevant subsystems is indeed small. The largest subsystem required for interoperability was only four entities. However, it should be noted the even single entity subsystems sometimes included many attributes since full-duplex interoperation requires attribute harmonisation.

- The partial reverse engineering process was in every case equivalent to a category theory modelling task — the usual elicitation tools were useful although we were exploring a legacy information system rather than a business process. Of course modelling is not a straightforward task, but after considerable category theoretic modelling experience it is not difficult, especially in contrast to more traditional reverse engineering.

- In none of the system pairs explored by us could full-duplex interoperation be achieved without legacy system modification.

- In over 60% of the system pairs half-duplex interoperations provided the required interoperability. Three quarters of those systems did still require legacy system modification, but in all of those cases only *one* of the two legacy system needed to be modified, a significant saving. The other two systems provided half-duplex interoperation without legacy system modification.

6 Partial Information and Interoperations

Although half-duplex interoperations demonstrated significant cost advantages, the fact that they still required so much legacy system modification led us to consider alternative approaches.

Recent work [13] has explored different approaches to the extension of category theoretic data modelling (the sketch data model) to incorporate partial information (attribute values which may be undefined or unavailable). We had hoped to explore the applicability of the "second technique" from [13], but it turns out that that technique has significant unexpected complications. Instead we are working with the third technique. The details of the technique are not important here, but our initial findings are well worth reporting.

- It appears that using the third technique to model possibly partial information is a natural extension of the JR-methodology. In particular, the mathematical methods required to test for propagatability are unaffected.

- When the legacy systems support partial information (for example, when they are databases that allow certain attributes to take the value `null`) there is generally no need for attribute harmonisation.

- Testing this extended methodology on thirteen system pairs we were able to design interoperations for all but one with no legacy system modification.

Of course it is interesting to note the circumstances that still require legacy system modification: In the one unsuccessful case we discovered that the attribute which would have been required to accept partial information was in fact a key attribute so the technique could not apply. Furthermore the key used was semantically meaningful and confidential, so even attribute harmonisation would have failed because the organisation would have refused to share the data.

7 Conclusion

Overall our results show that as expected full-duplex interoperations are very expensive. Half-duplex interoperations offer significant savings, but are still costly. In both cases the JR-methodology is effective and relatively straightforward. We are currently exploring using partial information, and this seems to be the most economical approach. It builds on the JR-methodology rather than replacing it, but it does depend upon the legacy system supporting partial information. Further mathematical analysis is required, and so are a range of other case studies to determine its applicability.

8 Acknowledgements

The research reported here has been supported by the Australian Research Council. We are also grateful to the New South Wales Department of Health whose data models were used in the tests described in this paper.

References

[1] K. Baclawski, D. Simovici and W. White. A categorical approach to database semantics. *Mathematical Structures in Computer Science*, 4, 147–183, 1994.

[2] M. Barr and C. Wells. *Category theory for computing science*. Prentice-Hall, second edition, 1995.

[3] Robert Colomb, C. N. G. Dampney and Michael Johnson. The use of category-theoretic fibration as an abstraction mechanism in information systems. *Acta Informatica*, 38, (2001) 1–44.

[4] C. N. G. Dampney and Michael Johnson. Fibrations and the DoH Data Model. Consultants' report to NSW Department of Health, 1999.

[5] C. N. G. Dampney, Michael Johnson and G. M. McGrath. Audit and Enhancement of the Caltex Information Strategy Planning (CISP) Project. Consultants' report to Caltex Oil Australia, 1993.

[6] Zinovy Diskin and Boris Cadish. Variable set semantics for generalised sketches: Why ER is more object oriented than OO. In *Data and Knowledge Engineering*, 2000.

[7] A. Islam and W. Phoa. Categorical models of relational databases I: Fibrational formulation, schema integration. Proceedings of the TACS94. Eds M. Hagiya and J. C. Mitchell. *Lecture Notes in Computer Science*, 789, 618–641, 1994.

[8] Michael Johnson and C. N. G. Dampney. On the value of commutative diagrams in information modelling. In The Unified Computation Laboratory, eds Rattray and Clarke, *Springer Workshops in Computing*, 77–84, Springer-Verlag, 1994.

[9] Michael Johnson and Robert Rosebrugh. View updatability based on the models of a formal specification. *Proceedings of FME 2001*, 534-549, *Lecture Notes in Computer Science* 2021, 2001.

[10] Michael Johnson and Robert Rosebrugh. Reverse engineering legacy information systems for internet based interoperation. *Proceedings of the IEEE International Conference on Software Maintenance*, 32–39, 2001.

[11] Michael Johnson, Robert Rosebrugh, and R. J. Wood. Entity relationship attribute designs and sketches. *Theory and Application of Categories* 10, 94-112, 2002.

[12] Michael Johnson and Robert Rosebrugh. Database interoperability through state based logical data independence. *International Journal of Computer Applications in Technology*, 16, (2003) 97–102.

[13] Michael Johnson and Robert Rosebrugh. Three approaches to partiality in the sketch data model. *Electronic Notes in Theoretical Computer Science*, 78, 1–18, 2003.

[14] E. Lippe and A ter Hofstede. A category theoretical approach to conceptual data modelling. *RAIRO Theoretical Informatics and Applications*, 30, 31–79, 1996.

[15] Saunders Mac Lane. *Categories for the Working Mathematician*. Graduate Texts in Mathematics 5, Springer Verlag, 1971.

[16] F. Piessens and Eric Steegmans. Categorical data specifications. *Theory and Applications of Categories*, 1, 156–173, 1995.

[17] F. Piessens and Eric Steegmans. Selective Attribute Elimination for Categorical Data Specifications. Proceedings of the 6th International AMAST. Ed. Michael Johnson. *Lecture Notes Computer Science*, 1349, 424-436, 1997.

[18] G. Southon, C. Sauer, and C. N. G. Dampney. Lessons from a failed information systems initiative: issues for complex organisations. *International Journal of Medical Informatics*, 55, 33–46, Elsevier Science, 1999.