

# MIQIS: Modular Integration of Queryable Information Systems

C. M. Wyss

G. H. L. Fletcher

F. Erdinc

J. T. Engle

Indiana University at Bloomington  
cmw, gefletch, ferdinc, jtengle@cs.indiana.edu

## Abstract

Information integration is not a new problem. By all accounts, language has always taken a multitude of forms, thus the need for translating between different representations of our world has been acute throughout history. With the advent of the World Wide Web, however, this need takes on hitherto unseen dimensions in terms of both difficulty and promise. Automated, web-based integration could offer unparalleled access to worldwide perspectives, but the range of difference among sources is formidable — even in the case of a single underlying representational model. The MIQIS project aims to provide a high-level logic for reasoning about integration scenarios and their fundamental properties. A key contribution of this perspective is the unification of the *semantic*, *syntactic*, and *effective* aspects of information integration. Within MIQIS we research foundational properties of inter- and intra-representation integration frameworks, as well as produce practical, modular solutions as specific applications of the general theory.

## 1 Introduction

Classical information integration in a federated system of information sources has focussed on syntactic integration of types through source schema mappings into a mediated schema [12]. More recently, pairwise schema mappings have been the focus of research for *Peer-to-Peer* integration systems [5, 18]. Contemporary solutions recognize the need for semantic mappings between tokens; elegant implementations of semantic maps include the mapping relations of [14] and the relaxation labeling of [7]. Recent logical formalisms acknowledge the need to encompass both

semantic and syntactic integration aspects, at least within a single representation [17].

Thus, much progress has been made toward automated information integration during the last decades. However, there are still crucial elements missing from current solutions. The next paragraphs present three such elements, all of which MIQIS aims to encompass.

**Inter-Model Integration.** Current integration solutions adopt a single underlying representational model, such as relational, object-oriented, or semi-structured.<sup>1</sup> In contrast, many integration applications require translation between such models. It has been suggested that high-level “meta-model” frameworks can formalize inter-model integration [3, 4, 19]. MIQIS aims to provide inter-model integration using higher-order logics capable of manipulating queries and representational structures. Our underlying formalization of data model is *set-based*, and comes from the mathematical structures termed *Chu Spaces* [1]. Chu spaces have been successfully applied to concurrency theory [10], ontology matching [13], and information flow [2]; we aim to extend this work to information integration and develop applications for inter-model integration consistent with this theory.

**Parametrized Schemas.** Many integration scenarios involve translating between relational and multidimensional or spreadsheet data. One example is shown in figure 1. The translation shown is distinguished from canonical examples in that *the target schema depends on the source data instance*. To perform such integration, operations such as Pivot or Transpose [21] return results whose shape depends on the input data. Although such transformations have been supported in vendor applications for some time,<sup>2</sup> the theory underlying such translations was not well understood. To this end, we have developed an extended rela-

---

*Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the VLDB copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Very Large Data Base Endowment. To copy otherwise, or to republish, requires a fee and/or special permission from the Endowment.*

---

<sup>1</sup>An exception is the SilkRoute system, which translates relational to semi-structured data [8]. SilkRoute does not consider the general problem of integration when a target schema has been previously determined. However, SilkRoute includes rules for translating queries on XML documents into SQL, as well as schema mapping capabilities. In this regard, SilkRoute is paradigmatic of the inter-model approach we want to formalize.

<sup>2</sup>The operation “Pivot”, for example, is supported in Microsoft Access, Excel, and SQL Server 2005.

StudentName	AssignmentID	Grade	→	StudentName	Asg1Grade	Asg2Grade	Asg3Grade	...	AsgNGrade
-------------	--------------	-------	---	-------------	-----------	-----------	-----------	-----	-----------

Figure 1: Relational to Spreadsheet Schema Translation.

tional model, the *federated data model* that naturally handles parametrized schemas. Furthermore, we have characterized the cases where merging transposed data is well-defined [16]. This work is crucial for understanding relational to multidimensional integration.

**Integral Query Languages.** To our knowledge, the impact of heterogeneity among peer *query languages* has been largely ignored. Respecting query languages is critical, however, since interoperation may fail if the query capability at a given source is simply not powerful enough to run the translated queries that reach it (even if the representation supports answers). We term this facet of integration *effectiveness*. Effective integration is particularly important for the Web, where access to sources is very often given in terms of query *forms* with limited capabilities. In MIQIS, our formalization of an *information model* codifies the *syntactic*, *semantic*, and *effective* capabilities of constituent information sources.

The MIQIS project aims to automate information integration within a general theoretical framework that encompasses all three desiderata listed above. Furthermore, MIQIS aims to incorporate existing solutions and techniques where appropriate, or complement these solutions. In the next sections, we indicate the general approach of MIQIS, focussing on how the approach addresses the three desiderata given above.

## 2 Foundations of MIQIS

Central to any theory of information integration is a formalization of the notion of *data model*. Recent work views data models as *graphs* [3]. In contrast, MIQIS codifies the notion of data model as a *Chu Space* [1] or *Classification* [2], as follows.

### Definition 1

A data model is a triple  $\mathcal{D} = \langle A, T, \vDash \rangle$  where  $A$  is a set of tokens,  $T$  is a set of types, and  $\vDash$  is a relation  $\vDash \subseteq A \times T$ . In case  $(a, t) \in \vDash$  for  $a \in A$  and  $t \in T$ , we write “ $a \vDash t$ ” (read “ $a$  is of type  $t$ ”).

The benefit of Chu spaces is that (unlike plain sets) internal structure is “built in” to the formalism. Like sets, however, Chu spaces can provide a formal basis for the development of mathematical concepts [15]. The internal division of a Chu space into tokens and types means that this *duality* fundamentally follows ensuing theoretical development. We feel that this duality correctly captures the syntactic/semantic duality inherent in data models (or any linguistic representation).

As indicated above, for effective integration we must model the query language accompanying a data model as

well. We can use the existing concept of a *Local Logic* [2] to model this. We use the term *Information Model* instead of “Local Logic” to better reflect our intended use.

### Definition 2

An Information Model  $\mathcal{I}$  is a data model  $\mathcal{D} = \langle A, T, \vDash \rangle$  together with a relation  $\vdash$  on types, i.e.  $\vdash \subseteq T \times T$ . In case  $(t, t') \in \vdash$  for  $t, t' \in T$ , we write “ $t \vdash t'$ ”.

The relation  $\vdash$  models the type theory of a query language accompanying the data model  $\mathcal{D}$ . The *dual* of  $\vdash$  gives a high-level model of a query language as a *query relation*. We will notate the dual relation by  $\mathbb{Q}$ .

### Definition 3

Given an information model  $\mathcal{I} = \langle A, T, \vDash, \vdash \rangle$ , the Query Relation corresponding to  $\mathcal{I}$  is the relation  $\mathbb{Q} \subseteq A \times A$  such that  $(a, a') \in \mathbb{Q}$  iff there exist  $t, t' \in T$  such that (i)  $a \vDash t$  and  $a' \vDash t'$  and (ii)  $t \vdash t'$ .

The advantage of these definitions is that we can now formally state what we desire as the end product of information integration. From there, we can reason about the high-level properties the information models and/or maps between them must have to guarantee our desiderata.

Given two information models  $\mathcal{I}_1$  and  $\mathcal{I}_2$ , an *Integration Morphism* between them is a bi-level map from tokens-to-tokens and types-to-types that respects the structure of both models as well as both query relations. Formally, this is spelled out in Definition 4.

### Definition 4

Let  $\mathcal{I}_1 = \langle A_1, T_1, \vDash_1, \vdash_1 \rangle$  and  $\mathcal{I}_2 = \langle A_2, T_2, \vDash_2, \vdash_2 \rangle$  be information models. Let  $\mathbb{Q}_1$  and  $\mathbb{Q}_2$  be the induced query relations. An Information Morphism from  $\mathcal{I}_1$  to  $\mathcal{I}_2$  is a pair of contravariant maps  $f = (f^\vee, f^\wedge)$  where  $f^\vee : A_1 \rightarrow A_2$  and  $f^\wedge : T_2 \rightarrow T_1$  such that

1.  $a \vDash_1 f^\wedge(t)$  iff  $f^\vee(a) \vDash_2 t$  for all  $a \in A_1$  and  $t \in T_2$ ; and
2. for all  $(a, a') \in \mathbb{Q}_1$ ,  $(f^\vee(a), f^\vee(a')) \in \mathbb{Q}_2$ .

Due to the dual nature of information models, there are many equivalent definitions expressing the same properties. However, the above formalization best captures how MIQIS will implement integration of information models. Namely, the map  $f^\vee$  will be created *first*, and  $f^\wedge$  will be deduced from  $f^\vee$ . Once this is achieved, the query relations will be checked for compatibility. This step is important, and in general we may have to settle for an integration morphism from a *sub-model* of  $\mathcal{I}_1$ . Part of MIQIS is to codify how the integration process is impacted by the accompanying query relations.

**Critical Instances.** A crucial insight is the notion of a *critical instance* of a data model. A *critical instance* is a

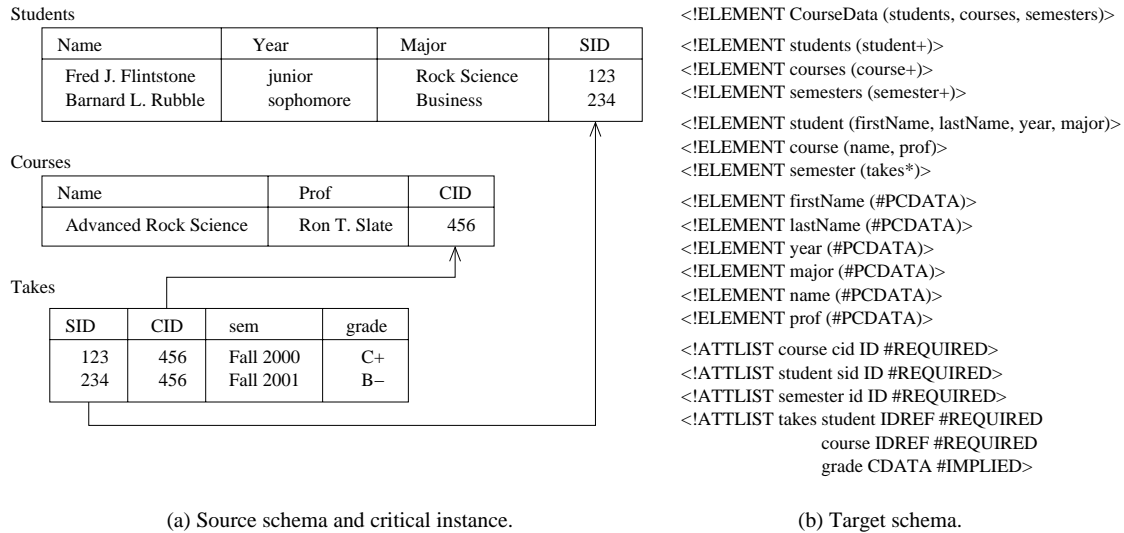


Figure 2: Input to MIQIS integration platform.

(small) instance of the source data model which is semi-automatically mapped to an equivalent instance of the target model. This instance acts as a *Rosetta Stone* to guide the integration. This approach is similar to deducing a formal query from a representative example in “Query-By-Example” interfaces.

Next, we give a representative example of the approach, indicating key contributions of MIQIS.

### 3 Illustrative Example

Consider the case of translating between relational and XML representations of data about students taking courses at a local university. Such a translation is desired, for example, in order to display course data on the web. Sample source data as well as both source and target schemas are depicted in Figure 2. In the example, both the relational and XML schema are determined by the application (unlike in SilkRoute, where the point is to generate an appropriate XML translation of relational data), and the task is to automate the process of mapping the relational data (and queries on it) to the XML data.

In order to start the process, we first semi-automatically create a semantic map between a critical instance of the source schema and an equivalent instance of the target schema. A GUI interface will be provided to assist in declaring the source and target schemas, entering the critical instance, and semi-automatically generating the initial semantic map. The system will provide a similar interface to Lixto [9] for selecting tokens of the source model and mapping them to tokens of the target model. The MIQIS system is more general, however, and will include modules for interoperating among several information models, beginning with relational, semi-structured, and multidimensional models. A mock-up of the GUI interface for manag-

ing critical instances is shown in Figure 3.

Once the semantic map for the critical instance is created, the system extrapolates the mapping to  $f^V$ , and subsequently to  $f^A$ . The result may be parametrized, either by the source data or the source or target schemas (or all of these). An advantage of this *semantics-directed* integration approach is that the critical instance will be small, so a minimum of effort is required on the part of the user. Nevertheless, the critical instance must satisfy certain properties in order to induce a well-defined integration morphism. Part of ongoing work is to precisely classify desired properties of critical instances.

In the simple example given in Figure 2, only a single tuple is needed per relation, although the values in the tuples should be *distinct* to rule out coincidental matches.<sup>3</sup>

Assuming a correct semantic map for the critical instance, schema mapping rules are generated. These rules have the form:

$$Q_S \rightarrow Q_T$$

where  $Q_S$  is a term in the source query language and  $Q_T$  is a term in the target query language. Unlike previous rule-based mappings, MIQIS uses native query languages to produce model-specific rules. Since the query terms on both sides of the rewrite rules are taken from the information models themselves, these rules will yield effective integration morphisms.<sup>4</sup> This entails that the overarching language of these rules is *higher-order*, as it must manipulate parametrized query terms as well as data. Furthermore, both metadata and data appear in the rules (and may even be cross-compared). The basis for this research is existing frameworks on reflexive and higher-order query languages

<sup>3</sup>Technically, this is unnecessary since the user will assist in creating the semantic map, but it ensures the default semantic map generated is more correct (i.e. less work for the user).

<sup>4</sup>The source and target query languages will need to satisfy commonly held properties to ensure this, such as compositionality.

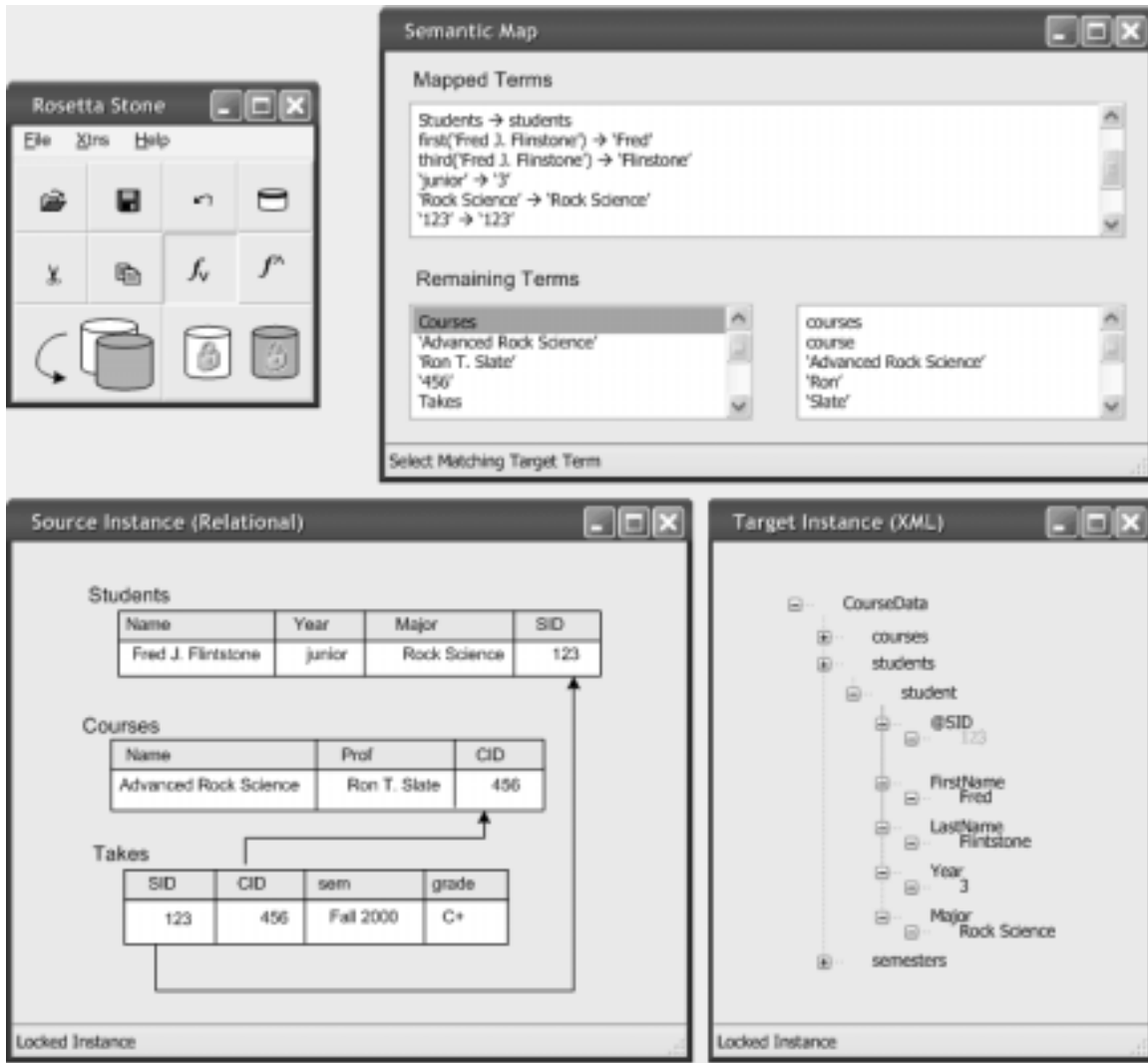


Figure 3: Mock-Up of User interface for managing critical instance.

[6, 20, 21] and higher-order logics [11].

In the example, we can summarize the mapping rules using parametrized Relational Algebra terms for the source side and parametrized XPath terms on the target side (the parameters unify both sides). As an illustration, our examples generates rules including the following:

$$\text{takes} \rightarrow /CourseData/semesters,$$

$$\sigma_{\text{sem}=X}(\text{takes}) \rightarrow /CourseData/semesters/semester[@id=X],$$

$$\sigma_{\text{SID}=X \wedge \text{CID}=Y \wedge \text{grade}=Z}(\text{takes}) \rightarrow /CourseData/semesters/semester/takes[@student=X \text{ and } @course=Y \text{ and } @grade=Z].$$

**Token Normal Form** One of our main tools for deducing integration morphisms from critical instances is a normal

form suggested by the Chu space formalism [2]. In this normal form, which we term *Token Normal Form*, a data model instance is represented as a list of pairs of the form  $(a, \{t_1, \dots, t_n\})$  where  $t_1, \dots, t_n$  is an enumeration of the types of  $a$  according to the instance. We hypothesize that any integration morphism can be deduced correctly from a finite critical instance. The general procedure is as follows. Suppose both the source and target data of our critical instance are in TNF. The semi-automatically generated semantic map tells us which token(s) in the target instance correspond to a given source token. It remains to match up the source and target types of the two TNF representations, yielding our parametrized rules. From these, the higher-order host language should enable deduction of a generalized integration morphism. Ideally, this host language should also support proofs that the morphism satisfies the properties of Definition 4.

## 4 Summary and Future Work

In this paper, we have presented an overview of the aims and approach of the MIQIS project, currently being undertaken at Indiana University. Future work on MIQIS will follow three general paths:

- *Foundational* — we have suggested throughout the paper directions that research will take on a high-level framework for reasoning about integration scenarios. Where possible, MIQIS will extend and/or unify existing research appropriately.
- *Instantiation* — the modular aspect of MIQIS is that specific information models may be “plugged in” as needed. Initial work has begun on supporting the relational, semi-structured, and multidimensional information models.
- *Implementation* — work has begun on an interface for defining source and target schemas and managing critical instances. The underlying engine is being designed for a Prolog implementation.

All three avenues of research encompass inter-model integration, parametrized schemas, and integral query languages. In this way, the MIQIS project provides a foundation for semantic, syntactic, and effective information integration.

## References

- [1] M. Barr. “The Chu Construction.” In *Theory and Applications of Categories*, 2(2):17-35 (1996).
- [2] J. Barwise and J. Seligman. *Information Flow : The Logic of Distributed Systems*. CUP, 1997.
- [3] P. A. Bernstein, L. M. Haas, M. Jarke, E. Rahm, G. Wiederhold. *Panel: Is Generic Metadata Management Feasible?* VLDB 2000.
- [4] Philip A. Bernstein, Alon Y. Halevy, Rachel Pottinger. *A Vision of Management of Complex Models*. SIGMOD Record 29(4):55-63 (2000).
- [5] D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. *Logical Foundations of Peer-to-Peer Data Integration*. SIGMOD 2004.
- [6] W. Chen, M. Kifer, D. S. Warren. *HiLog as a Platform for Database Languages*. DBPL 1989.
- [7] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. *Learning to Map Between Ontologies on the Semantic Web*. WWW 2002.
- [8] M. Fernández, Y. Kadiyska, D. Suciu, A. Morishima, and W.-C. Tan. *SilkRoute: A Framework for Publishing Relational Data in XML*. TODS 27(4):438-493 (2002).
- [9] G. Gottlob, C. Koch, R. Baumgartner, M. Herzog, and S. Flesca. *The Lixto Data Extraction Project: Back and Forth between Theory and Practice*. PODS Keynote address, 2004.
- [10] V. Gupta. *Chu Spaces: A Model of Concurrency*. PhD Thesis, Stanford University, 1994.
- [11] D. Leivant. *Higher Order Logic*. Indiana University Technical Report TR-388 (1993).
- [12] M. Lenzerini. *Data Integration: A Theoretical Perspective*. PODS 2002.
- [13] Y. Kalfoglou and M. Schorlemmer. *Formal Support for Representing and Automating Semantic Interoperability*. ESWS 2004.
- [14] A. Kementsietsidis, M. Arenas, and R. J. Miller. *Mapping Data in Peer-to-Peer Systems: Semantics and Algorithmic Issues*. SIGMOD 2003.
- [15] V.P. Pratt. *Chu Spaces*. Course Notes for the 1999 Summer School in Category Theory and its Applications.
- [16] E. Robertson and C. Wyss. *Optimal Tuple Merge is NP-Complete*. Indiana University Technical Report TR-599 (2004).
- [17] L. Serafini, F. Giunchiglia, J. Myopoulos, and P. A. Bernstein. *Local Relational Model: A Logical Formalization of Database Coordination*. CONTEXT 2003.
- [18] I. Tatarinov and A. Halevy. *Efficient Query Reformulation in Peer Data Management Systems*. SIGMOD 2004.
- [19] M.-N. Terasse, M. Savonnet, E. Leclercq, and G. Becker. *Building Platforms for Information System Interoperability: A UML-based Metamodeling Approach*. EFIS 2003.
- [20] J. Van den Bussche, S. Vansummeren, G. Vossen. *Meta-SQL: Towards Practical Meta-Querying*. EDBT 2004.
- [21] C. Wyss and D. Van Gucht. *A Relational Algebra for Data/Metadata Integration in a Federated Database System*. CIKM 2001.