Knowledge Representation Architecture for Context Interchange Mediation: Fixed Income Securities Investment Examples

Allen Moulton Stuart E. Madnick Michael D. Siegel
MIT Sloan School of Management
amoulton@mit.edu *smadnick@mit.edu* *msiegel@mit.edu*

Abstract

We examine a knowledge representation architecture to support context interchange mediation. For autonomous receivers and sources sharing a common subject domain, the mediator's reasoning engine can devise query plans integrating multiple sources and resolving semantic heterogeneity. Receiver applications obtain the data they need in the form they need it without imposing changes on sources. The KR architecture includes: 1) data models for each source and receiver, 2) subject domain ontologies, containing abstract subject matter conceptualizations that would be known to experienced practitioners in the industry, and 3) context models for each source and receiver that explain how each source or receiver data model implements the abstract concepts from a subject domain ontology. Examples drawn from the fixed income securities industry illustrate problems and solutions enabled by the proposed architecture.

1. Introduction

Efficiently integrating new sources of information from outside the enterprise is often critical to success in a world of global competition, interdependency, and rapid market change. Within an organization, data can be created, stored, and used by people and computers sharing a common implicit understanding of data semantics. We use the term *context* to refer to this implicit understanding of the relationship between data elements and structures and the real world that the data represents. The context interchange problem arises when organizations with different contexts must exchange information[4].

A context interchange (COIN) mediator is an automated reasoning engine to assist an organization in resolving semantic conflicts between its own receiver context and the contexts of data sources (Goh et al., 1999). Because context definitions are declarative, they need only be prepared once for each source and receiver context[1]. Data sources may be relational databases, XML documents, HTML webs wrapped to appear as relations with limited query capability[2], and stateless computational procedures. Using declarative context knowledge, a COIN mediator identifies semantic conflicts and designs plans for combining sources with data conversions to meet receiver semantic requirements.

Our work is based on the semantic proposition that interchange of information can be mediated if sources and receivers share a common subject domain (or interlocking subject domains). Sources and receivers are seen as autonomous implementations of a common subject domain abstraction (or interlocking abstractions). Source and receiver system designers make decisions about how to conceptualize abstract constructs and about how to represent that conceptualization in data and programs. The COIN mediator has the task of applying declarative information about the context of each source and receiver to device plans for integrating sources to meet receiver requirements.

Given a large number of component systems operating in a diversified and dynamic environment, COIN mediation facilitates: rapid incorporation of new information sources, dynamic substitution of information sources, extension and evolution of semantics, data representation in the user's context, access to the meaning of data represented, identification and selection of information source alternatives, and adaptation to changes in user and business operations.

Building on earlier work by Goh[3], we are exploring knowledge representation and reasoning methods to expand the functionality of COIN mediation to include: 1) identifying data representation conflicts and introducing conversions to transform data from source to receiver form, 2) applying subject domain and context knowledge to map between receiver schema and source schemata, 3) determining when and how to combine sources, feeding data from one source to another with appropriate data conversions, 4) deriving missing data by applying domain ontology, context knowledge, or by combining sources. In addition to databases and web-based data sources, we also include computational procedures as transformational sources. Where possible we employ a knowledge representation consistent with common system design practices (e.g., UML, E-R, and repositories).

1529-4188/01 $10.00 © 2001 IEEE

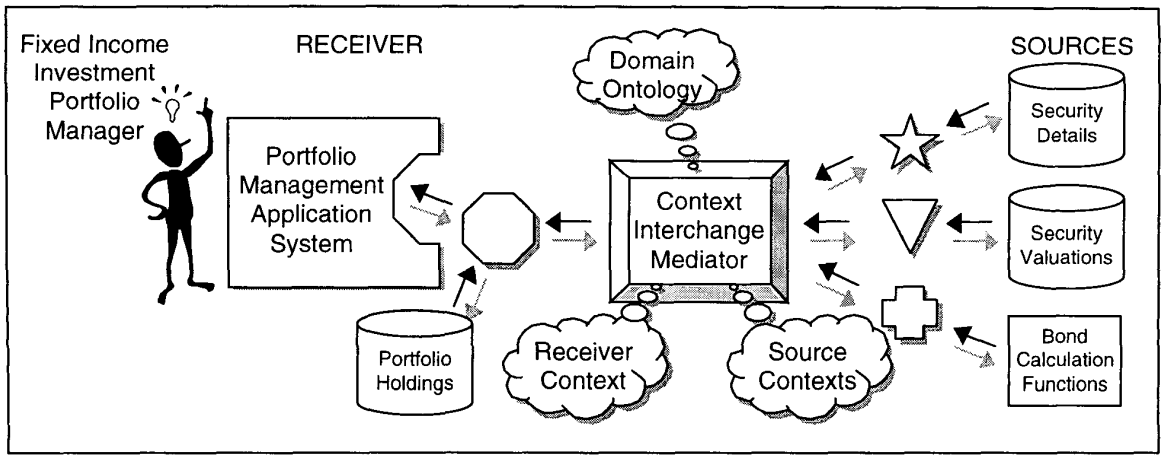


Figure 1. Fixed income securities investment mediation scenario

2. Fixed income securities examples

Equity and fixed income investment provide examples of rich, and largely disjoint, subject domains where COIN mediation can play an important role. Effective investment decision-making often involves combining internal information with information from autonomous, external sources. Equity investors may need market prices, historical corporate financial data, and analyst predictions of future performance. Fixed income investment managers also need to integrate information on market valuations, details of security characteristics, and analytic models and calculations.

Fixed income securities are debt obligations, including debt issued by corporations, governments and non-public entities, as well as aggregations of mortgage loans, car loans, and credit card debt. Equity investors take ownership interests in companies; fixed income investors own the rights to future cash payments detailed in the security itself. Equity investors deal with at most a few thousand stocks actively traded on exchanges; fixed income investors deal with millions of unique and rarely-traded securities, few of which are listed on exchanges. Equity investors use accounting information and models to evaluate companies; fixed income investors use mathematical models and analytic methods to evaluate cash flows. Dealers play a central role in the fixed income market, underwriting new securities and making markets and finding opportunities to match buyers and sellers of existing securities.

In the fixed income securities industry, portfolio managers may need to draw upon external sources for data about security characteristics, for market valuation information, and for models and calculations[6]. All these sources may need to be combined with internal portfolio holdings data and accessed through a decision support application system (Fig. 1). One task involves obtaining current dealer offerings and presenting the portfolio manager, in a consistent manner, information about securities offered and dealer prices. The table below shows a partial schema and semantics for an offerings application requirements.

Receiver relation R (application requirements)		
attribute	**sample data**	**semantic notes**
secidn	191219AN4	CUSIP security identifying number
matdat	02/01/2012	maturity date, mm/dd/yyyy
cprate	8.500	interest rate, percent, decimal fraction
price	116.08	dollar price, percent, fraction in 32nds

The first three attributes provide information about the security offered (standard CUSIP identifier, maturity date, and coupon interest rate). The last attribute is the price asked by the dealer. The application query would be:

```
SELECT secidn, matdat, cprate, price
  FROM R
  WHERE <criteria>
```

To explore context interchange problems, we consider two alternative sources for offerings. Dealer A provides a web page that has been wrapped for a relational query interface. Dealer B provides an XML document.

2.1 Data representation semantics

The table below shows a section of the schema for Dealer A offerings.

Source relation A (dealer A offerings web page)		
attribute	**sample data**	**semantic notes**
cusip	191219AN4	CUSIP security identifying number
maturity	40940	maturity date, Lotus/Excel 1900 date
coupon	0.08500	interest rate, factor, decimal fraction
price	116.25	dollar price, percent, decimal fraction

The first step in developing a mediation plan is to note that each row in A matches the requirement for a row in R – each represents a dealer offering for a security. Next, by examining the semantic notes, it is seen that, although three attribute names are different, each attribute in R can be obtained from one attribute in A.

The final step is to resolve data representation differences. R.secidn and A.cusip are the same. R.matdat requires a date in "mm/dd/yyyy" format; A.maturity is provided as a Lotus date sequence number. R.cprate is in percent; A.coupon is in factor form commonly used in spreadsheets. The price attributes, though named the same, are subtly different in semantics. R.price requires a percent with fraction in 32nds; A.price is expressed as a percent with decimal fraction. Failure to convert from decimal to 32nds could result in a substantial error in the price. Having identified the semantic conflicts, the mediator inserts appropriate data conversions: multiplying by 100 to convert factor to percent, the Excel "dollarfr" function to convert a decimal fraction into 32nds, and a wrapped date conversion function, source F:

Source F (wrapped date conversion function)		
attribute	**sample data**	**semantic notes**
out	02/01/2012	reformatted date output
outformat	"mm/dd/yyyy"	format for output date
in	40940	date input
informat	"Lotus"	format for input date

The query rewritten in terms of the source schema with necessary data representation conversions would be:

```
SELECT cusip as secidn, v.out as matdat,
  coupon*100 as cprate, dollarfr(price,32) as price
 FROM A, F
 WHERE <criteria>
  AND F.outformat = "mm/dd/yyyy"
  AND F.in = maturity AND F.informat = "Lotus"
```

2.2 Derived data and multiple source integration

The use of XML simplifies the access to data in many respects, but still leaves a wide range of semantic issues to be resolved. Adoption of standards can reduce the degree of semantic heterogeneity. Nevertheless, in the securities industry and many others, innovation will proceed faster than standards. Consider Dealer B offerings provided as an XML document such as:

```
<OFFERSHEET>
   <OFFER>
      <BOND> 191219AN4 </BOND>
      <PRICE> 103.28 </PRICE>
   </OFFER>     ...
</OFFERSHEET >
```

Dealer B offerings have a tabular structure that can be represented as a relation as shown below. Comparing the semantics of B to requirements R, we note that two of the attributes of the security are missing. Furthermore, the price is expressed as a "nominal spread" in "basis points" instead of a "dollar price" in percent. To meet the receiver's requirements, general industry knowledge and additional data sources must be brought to bear, along with conversion of units and scaling.

Source relation B (dealer B offers XML document)		
attribute	**sample data**	**semantic notes**
BOND	191219AN4	CUSIP security identifying number
PRICE	103.28	nominal spread, basis points

To resolve the semantic conflict, the mediator must know 1) source C can provide security details, 2) nominal spread means the difference between yield on a security and a benchmark yield, 3) the on-the-run 10-year T-note yield is an appropriate benchmark, 4) which can be obtained from source D, 5) bond calculation source object E can convert yield to price given the security's interest rate and other details, 6) rules for converting data codes, 7) basis points are $1/100^{th}$ of a percent, and 8) methods for converting data representations as discussed above.

After analyzing the semantic differences between the receiver and source B, the mediator identifies additional data sources C and D, and calculation source E (see tables at the end of the paper). The mediator must insert data conversions for dates and percentages as discussed above. Data codes for payment frequency from source C must be mapped to E's context and the day count basis inferred from market conventions. Combining these sources, data conversions, mappings, and inferences, the resultant mediated query would look like:

```
SELECT B.cusip as secidn, v.out as matdat,
   C.coupon as cprate, dollarfr(E.price,32) as price
 FROM B, C, D, E, F v, F w, F x
   Cfreq, Cmarket, Efreq, Ebasis, Mmarket
 WHERE <criteria>
   AND v.outfmt = "mm/dd/yyyy"
   AND v.in = C.maturity AND v.infmt = "mm-dd-yyyy"
   AND C.cusip = B.BOND
   AND E.settlement = x.out
   AND x.outfmt = "Lotus"
   AND x.in = "11/01/2001" AND x.infmt = "mm/dd/yyyy"
   AND w.outfmt = "Lotus"
   AND w.in = C.maturity AND w.infmt = "mm-dd-yyyy"
   AND E.rate = C.coupon/100
   AND E.yld = ( B.PRICE/100 + D.10yr ) / 100
   AND E.redemption = 100
   AND E.frequency = Efreq.xcode
   AND Cfreq.freq = Efreq.freq
   AND C.payFreq = Cfreq.xcode
   AND E.basis = Ebasis.xcode
   AND Mmarket.daycount = Ebasis.daycount
   AND Mmarket.mcode = Cmarket.market
   AND C.market = Cmarket.xcode
```

Without mediation, the portfolio manager would see a price of 116.25 from Dealer A and 103.28 from Dealer B. With mediation, Dealer B's quote is converted to a dollar price of 117 28/32 and the comparison is reversed.

3. Knowledge representation architecture

Our knowledge representation architecture divides the knowledge used for mediation into three layers: 1) a domain ontology containing abstract subject domain concepts used by experienced practitioners and system designers in the industry, 2) data models for each source and receiver with the kind of information programmers would use to access data, and 3) context models for each source and receiver that explain how each source or receiver data model implements the abstract concepts from a subject domain ontology.

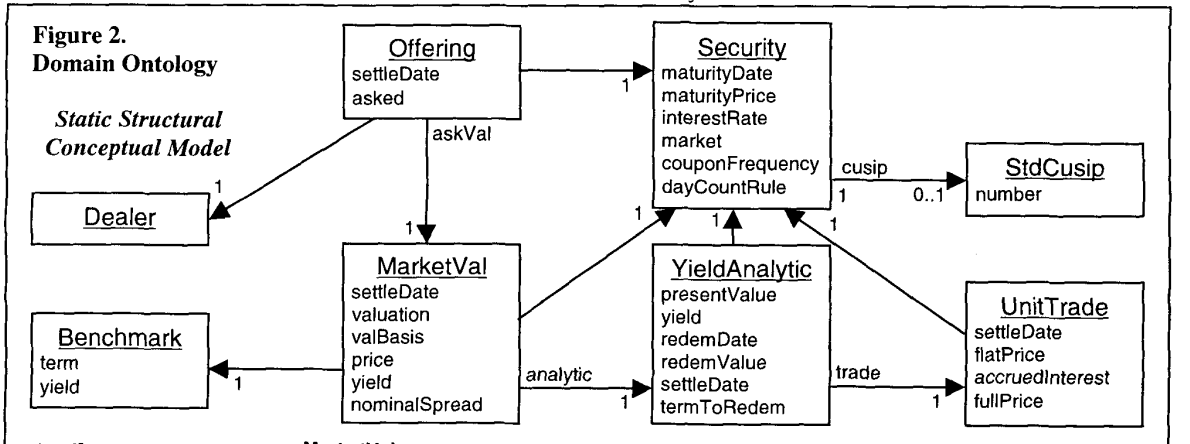
The framework of a domain ontology (e.g., Fig. 2) is a structural conceptual model with classes of abstract objects, attributes of objects, and relationships. Semantic types capture alternative data representations as in [3]. Enumerated conceptual categories model object property distinctions which may be implemented differently by each source and receiver. Rules capture functional relationships among conceptual model attributes that would be known from general domain knowledge. Default and contingent rules allow for deriving attributes based on partial information following the reasoning that industry participants would use.

Data models for the receiver and for relational sources use schema and catalog information. For XML, an XML schema or DTD can be used or the schema inferred from documents themselves. For HTML sources, the data model is provided by the web wrapper. For computational procedural sources, arguments and return values are treated as relational attributes in a data model that is augmented with functional dependency and input-output *combination constraints*.

Context models for each source and receiver explain how each data model implements the general concepts in the domain ontology. Classes from the domain ontology conceptual model can be used directly or augmented with context-specific extensions. Context-specific functional or equivalence relationships tie elements of the conceptual model to elements of the data model. For coding schemes, enumerated attribute domains are mapped to conceptual categories from the domain ontology. Semantic types can be used to logically encapsulate data attributes and associate context-specific modifier values to identify the data representation used.

A domain ontology is not a global schema. Rather, it is an abstract representation of the subject matter that each source and receiver data model implements in its own way. Neither sources nor receivers need to accept the domain ontology as the "right way" of representing information about the subject matter at hand, avoiding some of the practical user acceptance problems noted in [5]. By allowing each context model to extend the domain ontology and to explain how context-specific concepts map to general domain ontology concepts, mediation is facilitated without imposing the rigidity seen in view-based systems.

Figure 2.
Domain Ontology



Attribute constraint and default rules

Offering
security = askVal.security
settleDate = askVal.settleDate
asked = askVal.valuation

MarketVal
security = analytic.security
settleDate = analytic.settleDate
price = analytic.trade.flatprice
yield = analytic.yield
nominalSpread = yield – benchmark.yield
benchmark.term = analytic.termToRedem
valuation = choose(valBasis, price, yield, nominalSpread)

YieldAnalytic
security = trade.security
settleDate = trade.settleDate
presentValue = trade.fullPrice
Defaults:
redemDate = security.maturityDate
redemValue = security.maturityPrice

Enumerated conceptual category domains

Mfreq	
mcode	description
X43	every 6 months

Mmarket		
mcode	description	daycount
X19	Corporate	X75

Mdaycount	
mcode	description
X75	30/360 NASD

Defining context using our architecture is analogous to the process that a programmer would follow to design a program to extract data from sources. The first step is to model each source or receiver relation using conceptual objects from the domain ontology. In the example above, rows of R are modeled by *Offering*. Next, each data attribute in R is modeled with a conceptual attribute from the ontology, e.g.:

attribute in R	path from Offering concept in ontology
secidn	security.cusip.number
matdat	security.maturityDate
cprate	security.interestRate
price	asked

The *price* attribute in R is associated with the *asked* conceptual attribute in *Offering*. The other three attributes are located in the related *Security* or *StdCusip* objects, as indicated by the dotted path notation above. Sources A and B would be modeled similarly.

Source C would be modeled with a *Security* object. The one-to-one relationship between a *StdCusip* and a *Security* allows the mediator to associate the *Security* implicitly referred to in B with the data available from C.

Attributes *C.payFreq* and *E.frequency* represent the same concept using context-specific symbols. In defining a context, tables of symbols or codes are gathered from documentation or usage and then associated with conceptual categories from the ontology, e.g.

Cfreq	
xcode	freq
Semi-Annual	X43

Efreq	
xcode	freq
2	X43

Here the code *Semi-Annual* in C is associated with the meta-code *X43* from the ontology conceptual category domain *Mfreq*. Since the code *2* in E is similarly modeled, the mediator can a C code to an E code when the sources need to be joined.

Deriving the *basis* attribute in E requires an additional step. Context E code table *Ebasis* ties the code *0* with the *30/360 NASD* in the ontology conceptual category domain *Mdaycount*. Source C provides a code for the market of the security, but no day count basis.

Cmarket	
xcode	market
US Corporate	X19

Ebasis	
xcode	daycount
0	X75

In order to combine sources C and E, the mediator must draw on knowledge of the industry convention that U.S. corporate bonds use a day count of *30/360 NASD*. We capture this knowledge by linking the conceptual category domain *Mmarket* to *Mdaycount*. The mediator uses this general knowledge to infer the day count basis.

4. Conclusion

Context interchange mediation brings automated methods to the important task of assuring that data exchanged across organizations can meet the semantic integrity of the receiver – and do so without requiring the source organizations to accommodate the needs of the receiver.

References

[1] S. Bressan, C. H. Goh, N. Levina, S. E. Madnick, A. Shah and M. D. Siegel. "Context Knowledge Representation and Reasoning in the Context Interchange System," Applied Intelligence (13:2), Sept. 2000, pp. 165-179.

[2] A. Firat, S. E. Madnick, and M. D. Siegel. "The Caméléon Web Wrapper Engine," Proc. VLDB 2000 Workshop on Technologies for E-Services, Sept., 2000.

[3] C. H. Goh S. Bressan., S. E. Madnick and M. D. Siegel "Context Interchange: New Features and Formalisms for the Intelligent Integration of Information," ACM Trans. on Office Information Systems, July 1999, pp 270-293.

[4] S. E. Madnick. "Metadata Jones and the Tower of Babel: The Challenge of Large-Scale Heterogeneity," Proc. IEEE Meta-Data Conf., April 1999.

[5] A. Moulton, S. Bressan, S. E. Madnick and M. D. Siegel. "An Active Conceptual Model for Fixed Income Securities Analysis for Multiple Financial Institutions," Proc. ER 1998, pp. 407-420.

[6] A. Moulton, S. E. Madnick and M. D. Siegel. "Context Mediation on Wall Street," Proc. CoopIS 1998, pp. 271-279.

Source relation C (security characteristics web site)		
attribute	**sample data**	**semantic notes**
coupon	8.500	interest rate, percent, decimal fraction
maturity	02-01-2012	maturity date, MM-DD-YYYY
cusip	191219AN4	CUSIP security identifying number
datedDate	02-11-1992	issue date, MM-DD-YYYY
firstCoup	08-01-1992	first payment date, MM-DD-YYYY
market	US Corporate	market/type of security, text
payFreq	Semi-Annual	interest payment interval

Source relation D (Treasury yield curve web site)		
attribute	**sample data**	**semantic notes**
10yr	5.091	yield on current 10 year T-note

Source E (Excel analytic toolkit function PRICE)		
attribute	**sample data**	**semantic notes**
price	117.875	flat price, percent, decimal fraction
settlement	37196	settlement date, Excel 1900 date
maturity	40940	maturity date, Excel 1900 date
rate	0.0850	interest rate, factor, decimal fraction
yld	0.061238	yield, factor, decimal fraction
redemption	100	redemption value, percent
frequency	2	coupon frequency per year (1,2,4)
basis	0	day count basis, code (0,1,2,3,4)