Semantic Information Integration in the Large: Adaptability, Extensibility, and Scalability of the Context Mediation Approach

Thomas Gannon[1], Stuart Madnick[2], Allen Moulton[3],
Michael Siegel[3], Marwan Sabbouh[1], Hongwei Zhu[4]

[1]MITRE Coporation
{tgannon, ms}@mitre.org

[2]MIT Sloan School of Management & MIT School of Engineering
{smadnick}@mit.edu

[3]MIT Sloan School of Management
{amoulton, msiegel}@mit.edu

[4]MIT School of Enginering
{mrzhu} @mit.edu

Working Paper CISL# 2005-04

May 2005
(updated March 2006)

Composite Information Systems Laboratory (CISL)
Sloan School of Management, Room E53-320
Massachusetts Institute of Technology
Cambridge, MA 02142

Semantic Information Integration in the Large: Adaptability, Extensibility, and Scalability of the Context Mediation Approach

Thomas Gannon[1], Stuart Madnick[2], Allen Moulton[2],
Marwan Sabbouh[1], Michael Siegel[2], Hongwei Zhu[2]

[1]MITRE Corporation
202 Burlington Road
Bedford, MA 01730 USA
{tgannon, ms}@mitre.org

[2]Massachusettes Institute of Technology
30 Wadsworth Street, E53-320
Cambridge, MA 02142 USA
{smadnick, amoulton, msiegel, mrzhu}@mit.edu

Abstract

There is pressing need for effectively integrating information from an ever increasing number of available sources both on the web and in other existing systems. A key difficulty of achieving this goal comes from the pervasive heterogeneities in all levels of information systems. Existing and emerging technologies, such as the Web, ODBC, XML, and Web Services, provide essential capabilities in resolving heterogeneities in the hardware and software platforms, but they do not address the semantic heterogeneity of the data itself. A robust solution to this problem needs to be adaptable, extensible, and scalable.

In this paper, we identify the deficiencies of traditional approaches that address this problem using hand-coded programs or require complete data standardization. The COntext INterchange (COIN) approach overcomes these deficiencies by declaratively representing data semantics and using a mediator to create the necessary conversion programs using a small number of conversion rules. The capabilities of COIN is demonstrated using an intelligence information integration example consisting of 150 data sources, where COIN can automatically generate the over 22,000 conversion programs needed to enable semantic integration using only six parametizable conversion rules. This paper makes a unique contribution by providing a systematic evaluation of COIN and other commonly practiced approaches.

1. Introduction

In a recent report, "Making the Nation Safer", the National Research Council found that "Although there are many private and public databases that contain information potentially relevant to counter terrorism programs, they *lack the necessary context definitions (i.e., metadata) and access tools to enable interoperation with other databases and the extraction of meaningful and timely information*"[1]. Despite the fact that nearly 30% of IT dollars are spent on Enterprise Information Integration (EII)[2], organizations are still plagued by the lack of effective integration and interoperation. NIST found that lack of interoperability costs the U.S. capital facilities industry $15.8 billion per year [6]. As society becomes increasingly information intensive and the web continues to dramatically increase the range and number of sources easily available, semantic information integration is critical for effective exchange and utilization of valuable information. A viable solution to large scale integration has to be adaptable, extensible, and scalable.

Technologies already exist to overcome heterogeneities in hardware, software, and syntax used in difference systems. For example, the ODBC standard provides uniform access to relational databases across different DBMS platforms. On the Web, XML-based standards and web-wrapping tools facilitate data exchange and processing amongst distributed systems. With Web Services, one can access data as well as procedures in

[1] Emphasis added.
[2] See "Reducing Integration's Cost", Forrester Research, December 2001.

remote systems. While these capabilities are essential to information integration, they do not address the issue of heterogeneous data semantics. The data receiver still needs to reconcile semantic differences such as converting pounds and ounces into kilograms, or vice versa, depending on how the receiver wants to interpret the data. Hand-coding such conversions is only manageable on a small scale; alternative solutions are needed as the number of systems and the complexity of each system increase.

In this paper, we exemplify the need for robust semantic integration solutions and demonstrate COntext INterchange (COIN) as one such solution. Although COIN has been described in other reports [4, 7], there has only recently been research on evaluation criteria for flexibility [15]. In Section 2, we illustrate the issues and challenges of large scale semantic integration using a scenario of intelligence information integration. The shear volume of information on the web has made these issues even more challenging. In Section 3, we discuss various traditional ways of reconciling semantic differences and relate them to the example as well as general cases. These approaches include brute-force data conversion, global data standardization, and data interchange standardization. After summarizing the shortcoming of these approaches, we introduce the COIN ontology-based context-mediation approach in Section 4 and demonstrate how it is applied to the integration scenario to overcome identified shortcomings. In Section 5, we evaluate and compare these approaches in terms of adaptability, extensibility and scalability. A brief conclusion is given in Section 6.

2. Examples and Challenges of Intelligence Information Integration

Intelligence information usually is gathered by different agencies in multiple countries. Since no single agency is expected to have complete information, integration is necessary to perform various intelligence analyses, including basic questions such as "who did what, where, and when". Significant challenges exist when different agencies organize and report information using different conventions. We illustrate the challenges using several examples from the counter-terrorism domain. Similar issues exist in most other application domains where information integration is required, especially if heterogeneous semi-structured web sources are involved.

2.1 Person Identification

Identifying a person in a corporate database can be as simple as assigning a unique identification number, e.g., *employee_id*, for each person. This cannot be easily done across multiple agencies. Other attributes of a person are often used to help identify the records related to the person in different data sources.

Name of a person is a good candidate attribute, but different sources may record names differently, e.g., "William Smith" in one source and "Bill Smith" in another. Name spelling becomes more complicated when a foreign name is translated into English. For example, the Arabic name قذافي has been shown to have over 60 romanizations including: Gadaffi, Gaddafi, Gathafi, Kadafi, Kaddafi, Khadafy, Qadhafi, and Qathafi. There are numerous romanization and transliteration standards. But different agencies may choose different standards. For example from Arabic to English, the following romanization standards are commonly used: ALA-LC (library of Congress) 1972[3], DIN 31636 – 198 (Germany), EI (encyclopedia of Islam) 1960, ISO 233 –1984, UN 1972, USC – Transliteration of the Quran[4]

Other attributes such as weight and height of a person can be used conjunctively to help with identification matching. Again, different agencies may choose different standards for these attributes, e.g., a British agency may report weight in stones[5], while a U.S. agency might use pounds and a German agency might use kilograms. Similarly, these agencies may use feet, inches, and centimeters, respectively, for height.

It would be impossible to perform any useful intelligence analysis when the information from different sources is put together without reconciling these differences. To illustrate the difficulties, consider three records from three different sources shown in Table 1.

Table 1. Data from three different sources

Source	Name	Weight	Height	Place	Time	Event
UK	Gadaffi	12.14	5.67	London	12/11/2004 13:15	Plane arrives
US	Kadafi	170	68	London	11/15/2004 19:30	meeting
Germany	Qadhafi	77	173	Vienna	12/11/2004 11:30	Plane departs

In their present form, the three records apparently refer to three different people; it is unlikely that an analytical tool is able to generate any sensible output with these data. However, an important pattern will be revealed when the data from different sources are transformed into a uniform standard. For example, if the three records are converted to the standard used by the U.S. agency, we can relate the three records to the same person because after the conversion *Name*, *Weight* and *Height* are the same (e.g., 12.14 stones is equal to 179 lbs or 77 kg), and discover the pattern that a person named Kadafi, who weighs 170 lbs and measures 68 inches high, flew from Vienna to London on November 12, 2004 and later on November 15, 2004 had a meeting.

[3] See http://www.loc.gov/catdir/cpso/romanization/arabic.pdf
[4] See http://www.usc.edu/dept/MSA/quran/transliteration/table.html
[5] One stone is 14 pounds and widely used in England for reporting people's weight. See http://home.clara.net/brianp/weights.html for details.

2.2 Location Representation

Location information is often represented using place names, codes, and various geographic coordinates. Place names are not unique. A search for Cambridge at Weather.com returns eight cities located in Canada, UK, and U.S. Thus it is necessary to qualify a place name with other place names at different geographical granularities, e.g., Cambridge, MA, US or Cambridge, Ontario, CA. Here, country codes are used to indicate the country in which the city is located. Although country codes are compact and can eliminate problems with spelling and translation of country names, the same code sometimes represents different countries in different standards. The frequently used standards include the FIPS 2-character alpha codes and the ISO3166 2-character alpha codes, 3-character alpha codes, and 3-digit numeric codes. Confusions will arise when different agencies use different coding standards. For example, "explosion heard in the capital of BG" – is it in Bulgaria (if ISO 3166 2-charcter alpha code was used) or in Bangladesh (if FIPS code was used). Similarly, BD stands for Bermuda in FIPS, while it stands for Bangladesh in ISO 3166; and BM stands for Bermuda in ISO 3166 and for Burma in FIPS.

There are also multiple standards for airport codes. The two major ones are IATA and ICAO. For example, the code for Heathrow airport is LHR in IATA standard, EGLL in ICAO standard. A system that uses one code standard will not be able to recognize any airport designed with another standard.

One may contemplate that we should be able to identify a location by its geographical coordinate on earth. It turns out to be very complicated – there are over 40 widely-used, but different, geographic coordinate systems used around the world. Even within the U.S. Department of Defense (DoD) different standards are used by different branches of the armed forces, e.g, parts of the US Army and Marine Corps use the Universal Transverse Mercator (UTM) Grid and Military Grid Reference System (MGRS), while parts of the US Navy use latitude and longitude expressed in degrees, minutes and seconds, and parts of the US Air Force express them in degrees and decimal degrees. [6] Misinterpretation of these different representations can lead to ineffective coordination in the battle field or tactic operations in the war on terror.

2.3 Time Representation

The representations for other data elements could vary significantly among data sources. Time representation is particularly important for many applications. For example, date may be expressed using different calendars (e.g., besides the normal Gregorian calendar, there are others, such as the Jewish/Israeli calendar). Even when only the Gregorian calendar is used, year, month, and day can be arranged in different orders and using different punctuations, e.g., 11/12/2004 versus 12-11-2004, etc.

The time of day values can be at GMT time or local time (with different conventions for how to encode the time zone), standard time or daylight savings time, using either 12-hour or 24-hour format, etc. There is considerable variety of combinations and permutations[7].

2.4 An Integration Scenario

To further illustrate the challenges of integrating information from diverse sources, let us consider a scenario that involves many of the data elements discussed earlier.

After September 11, it became imperative that different agencies in the U.S. and among coalition countries share counter-terrorism intelligence information. Suppose there are a total of 150 such agencies, e.g., two dozen countries each having, on average, half dozen agencies. The shared information consists of person name, height, weight, airport, country, geo-coordinate of location, date, and time. To simplify explication, we assume that person name and time data have been standardized across the sources. For the rest of the attributes different agencies may use different conventions. The varieties of these conventions are summarized in Table 2.

Table 2. Semantic Heterogeneities in Data Sources

Data Types	Semantic varieties
Height	4 different units of measure: ft, in, cm, m
Weight	3 different units of measure: lbs, kg, stone
Airport	2 different coding standards: IATA, ICAO
Country	4 different coding standards: FIPS, ISO 2-Alpha, ISO 3-Alpha, ISO 3-digit
Geo-coordinate	4 different reference systems and datum parameters: MGRS_WGS84, BNG_OGB7, Geodetic_WGS84, UTM_WGS84
Date	4 different formats: mm/dd/yyyy, dd/mm/yyyy, dd.mm.yyyy, dd-mm-yyyy.

There are a total of 1,536 (i.e., 4*3*2*4*4*4) combinations from these varieties. Let us assume that each of the 150 data sources uses one of the combinations as its data representation convention. For example, a U.S. agency may choose to use inches for height, lbs for weight, IATA code for airport, etc., while a U.K. agency may choose to use feet for height, stones for weight, ICA for airport, etc. We use the term *contexts* to refer to these different ways of representing and interpreting data – there are potentially 1,536 unique contexts in this scenario.

[6] From http://www.findarticles.com/p/articles/mi_m0IAU/is_1_8/ai_98123571

[7] If there was not already enough variety, new time representations are being invented! For example, there is now a new "Internet Time invented and marketed by the Swiss watch company Swatch.

An agent from any of the 150 agencies may need information from all the other agencies to perform intelligence analysis. As shown in Table 1, when information from other agencies is not converted into the analyst's context, it will be difficult to identify important patterns. Therefore, a total of 22,350 (i.e., 150*149) conversion programs would be required to convert data from any source format to any other source format, and vice versa.

In practice, any specific analyst or analytical tool used by the analyst can have a context different from the agency's, e.g., an analyst from the CIA may use a tool that assumes height is in feet while the agency's databases use inches. Therefore, every data source and data receiver could have their own contexts, and in reality, there can be more than 150 information exchanging entities in the 150 agencies. For explication purposes, we continue the example with the assumption that there are only 150 sources/receivers.

Implementing tens of thousands of data conversions is not an easy task; maintaining them to cope with changes in data sources and receiver requirements over time is even more challenging. We will describe and discuss various approaches to this problem in the next two sections.

3. Traditional Approaches to Achieving Semantic Interoperability

3.1 Brute-force Data Conversions (BF)

The BF approach directly implements all necessary conversions in hand-coded programs. With N data sources and receivers, N(N-1) such conversions need to be implemented. When N is large, these conversions become costly to implement and very difficult to maintain. This is a labor-intensive process because many semantic differences have to be identified by humans and the conversions need to be implemented and maintained over time to account for changes in the underlying sources. This explains why nearly 70% of integration costs come from the implementation of these data conversion programs [2].

This approach might appear sufficiently inefficient that one might be surprised at how common it is. The reason is that usually the conversion programs are written incrementally – each individual conversion program is produced in response to a specific need. Writing "only one conversion program" does not seem like a bad idea – but over time, this process continues toward the N(N-1) conversion programs that must be maintained and updated.

3.2 Global Data Standardization (GS)

In the example, different data standards are used in the 150 agencies that need to exchange information. If they could agree on a uniform standard, e.g., standardizing height data to centimeters in all systems, all the semantic differences would disappear and there would be no need for data conversion. Unfortunately, such standardization is usually infeasible in practice for several reasons.

Often there are legitimate needs for storing and reporting data in different forms. For example, while height in centimeters makes sense to an agent in other NATO countries such as Germany, a U.S. agent may not find it useful until it has been converted to feet and inches. Since most integration efforts involve many existing systems, agreeing to a standard often means someone has to change current implementation, which creates disincentives and makes the standard setting and enforcement process extremely difficult. This difficulty is exacerbated when the number of the data elements to be standardized is large. For example, in 1991, the DoD initiated a data administration program that attempted to standardize nearly one million data elements[8]; by the year 2000, it only managed to register 12,000 elements, most of which were infrequently reused. After a decade of costly effort, the DoD realized its infeasibility and switched to an alternative approach to allow different communities of interest to develop their own standards [12].

The latter approach by the DoD manifests the reality of standard development, i.e., there are often competing or otherwise co-existing standards. As seen in the examples in Section 2, there are multiple standards for airport codes and for country codes. Different systems can potentially choose different standards to implement. Thus, in most cases, we cannot hope that semantic differences will be standardized away; data conversion is inevitable.

3.3 Interchange Standardization (IS)

The data exchange parties sometimes can agree on the format of what is to be exchanged, i.e., standardizing a set of concepts as well as interchange formats. The underlying systems do not need store the data according to the standard; it suffices as long as each data sender generates the data according to the standard. Thus each system still maintains its own autonomy. This is different from the global data standardization, where all systems must store data according to a global standard. With N parties exchanging information, the Interchange Standardization approach requires 2N conversions. This is a significant improvement over the brute-force approach that might need to implement conversions between every pair of systems.

This approach has been used for business transactions, e.g., EDI and various B2B trading standards. In the military setting, the U.S. Message Text Format (MTF)

[8] Since it was necessary to accommodate existing systems with different contexts, there were data elements for *fuel-load-in-liters* and *fuel-load-in-gallons*, without explicit acknowledgement of the relationship between these elements.

and its NATO equivalent, Allied Data Publication-3, have over 350 standard messages that support a wide range of military operations. This standard has been used for over 50 years and currently an XML version is being developed [11]. As a recent example, the DoD standardized exchange format of weather related data, which consists of about 1,000 attributes[9]. This standard has been successfully used by several systems that exchange weather data [12]. Similarly, the XML-based Cursor-On-Target (COT) standard, which consists of 3 entities and 13 attributes, has been used successfully by over 40 systems to exchange targeting information [12].

Although this approach has certain advantages, e.g., local autonomy and a smaller number of conversions required, it also has several serious limitations. First, all parties have to have a clear understanding about the domain, decide what data elements should go into the standard, and reach an agreement on the data format. This can be a costly and time consuming process. It took the DoD five years to standardize the weather data [10] interchange format. Furthermore, in many cases it is difficult to foresee what data needs to be exchanged or the requirements change over time, which makes it inappropriate to have a fixed standard. When the interested information is not specified in the standard, ad-hoc conversions have to be implemented. Lastly, any change to the interchange standard affects all systems and the existing conversion programs.

3.4 Summary of Traditional Approaches

Each of the three traditional approaches has certain drawbacks that make them inappropriate for integrating information from a large number of data sources. These weaknesses are summarized below:

- Brute-force data conversions (BF): requires a large number of hand-written conversions that are difficult to maintain;
- Global Data Standardization (GS): it is costly and sometimes impossible to develop a global standard. In addition to legitimate reasons of having multiple standards, there are technological difficulties and organizational resistance for a single standard;
- Interchange Standardization (IS): the standard is static, only suitable for routine data sharing and it still requires a large number of hand-written conversions.

In addition, these approaches lack flexibility to adapt to changes because the data semantics is hard-coded in the conversions for BF, in the standard in GS, and in both the conversions and the standard in the case of IS. A suitable approach needs to overcome these shortcomings. In the next section, we will discuss such an approach that automates code generation for conversions and requires no data standardization.

4. Ontology-based Context Mediation

Most of the shortcomings of the traditional approaches can be overcome by declaratively describing data semantics and separating knowledge representation from conversion implementation. There have been a number of research projects that utilize ontology to represent data semantics and to facilitate reconciliation of semantic differences [13]. Since an ontology is essentially an agreement on conceptual models, approaches that require a single, i.e. global, ontology have shortcomings similar to the data standardization approach. Therefore a multi-ontology approach is desirable to lower or eliminate the reliance on reaching a global agreement. In the following, we introduce the COntext INterchange (COIN) [1, 7, 8] approach, which allows each data source and receiver to describe its local ontology using a common language and also provides reasoning service to automatically detect and reconcile semantic differences.

4.1 The COIN Framework

The COIN framework consists of a deductive object-oriented data model for knowledge representation, a general purpose mediation service module that determines semantic differences between sources and receivers and generates a mediated query to reconcile them, and a query processor that optimizes and executes the mediated query to retrieve and transform data into user context (see Figure 1).

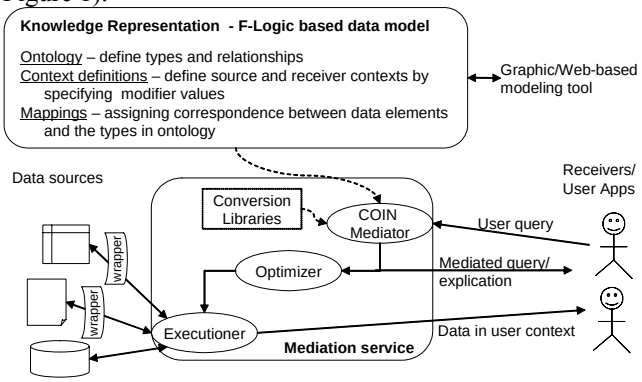

Figure 1. Architecture of COIN System

The COIN knowledge representation consists of three components. An *ontology* is used to capture common concepts and their relationships such as one concept being a property (i.e., attribute) or a sub-concept (i.e., is_a relationship) of another. A concept is roughly equivalent

[9] This is also known as the "communities of interests" approach, where organizations come together to develop standards for particulars domains in which they share common interests. These standards are equivalent to interchange standards when the organizations provide translations between the conventions in existing systems and the standards.

[10] Although now used by several DoD systems, it has not been adopted by all DoD legacy systems nor non-DoD systems (e.g., in private sector or foreign governments) that may need to interoperate with DoD systems.

to a class in object-oriented models and entity type in Entity-Relationship conceptual models[11]. Each concept may have *modifiers* as a special kind of property to explicitly represent specializations of the concept in the sources and receivers. We call the collection of declarative specifications of modifier values *context*. For each modifier, a rule or a set of rules are used to specify the conversions between different values of the modifier. The semantic mappings establish the correspondence between data elements in the sources and the concepts in the ontology. These components are expressed in the object-oriented deductive language F-Logic [10], which can be translated into Horn logic expressions that we use internally[12], or Web Ontology Language (OWL) and RuleML intended for the Semantic Web.

The core component in the mediation service module is the COIN mediator implemented in abductive constraint logic programming [9], where constraints are concurrently solved using Constraint Handling Rules (CHR) [5]. It takes a user query and produces a set of mediated queries (MQs) that resolve semantic differences. This is accomplished by first translating the user query into a Datalog query and using the encoded knowledge to derive the MQs that incorporate necessary conversions from source contexts to receiver context. The query processor optimizes the MQs using a simple cost model and the information on source capabilities, obtains the data, performs the conversions, and returns the final datasets to the user[13].

Within the COIN framework, the users are not burdened by the diverse and changing semantics in data sources, all of which are recorded in the knowledge representation component and are automatically taken into account by the mediator. Adding or removing a data source is accomplished by adding and removing declarations, which does not require any changes to the mediator or query processor – they will use the new knowledge to produce the new correct conversion programs, as needed.

4.2 Intelligence Information Integration using COIN

To apply COIN to the intelligence information integration scenario, none of the agencies need to change their current systems; they only need to record their context definitions by using the terms in a shared ontology. An excerpt of the ontology is shown in Figure 2.

In the ontology, concepts, i.e., types, are in rectangular boxes. There is a special type called *basic*, which has no modifier and serves as the parent of all the other types. We do not show the *is_a* relationship between the type *basic* and the rest of the types to avoid cluttering the graph. The shared ontology is completely different from a data standard in that it only contains basic concepts and their relationships, which are much easier to agree on than the representations of the types that are usually specified in a data standard. For example, the ontology only states that a person has weight, keeping silent about in what unit the weight should be. In fact, with this ontology, each data source and receiver can define their local ontologies by specifying modifier values to obtain desired specializations to the common types, e.g., specializing weight to weight in lbs. These specifications are called context definitions. Table 3 shows four example contexts that will be used later for a demonstration.

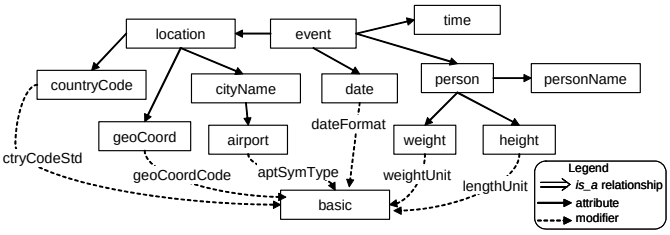

Figure 2. Excerpt of ontology

Table 3. Example contexts

Modifier	USA context	UK context	NATO context	Analyst context
dateFormat	mm/dd/yyyy	dd/mm/yyyy	dd.mm.yyyy	dd-mm-yyyy
ctryCodeStd	FIPS	ISO3166 2-alpha	ISO3166 3-digit	ISO3166 3-alpha
aptSymType	IATA	ICAO	ICAO	IATA
geoCoordCode	MGRS-WGS84	BNG-OGB7	Geodetic-WGS84	UTM-WGS84
lengthUnit	inches	feet	cm	m
weightUnit	pounds	stones	kg	kg

Both the ontology and the context definitions are declaratively defined and can be manipulated using graphic tools. For example, the following F-Logic formula states that in context *c_USA* the weight unit is *lb*:

$$\forall X : weight \; \exists Y : basic \vdash$$
$$X[weightUnit(c\_USA) \rightarrow Y] \wedge Y[value(c\_USA) \rightarrow 'lb'].$$

The modifiers of a type are represented as methods of the type. The *value* method returns a value in the context specified by the parameter. This method is implemented by the mediator to compare the modifier values between the source context and the receiver context; if they are different, conversions are introduced to reconcile the differences.

Conversions are defined for each modifier between different modifier values; they are called *component conversions*. The mediator automatically composes the overall conversion using the component conversions defined for relevant modifiers. In many practical cases, a component conversion can be parameterized to convert from any given context to any other given context for that modifier. For example, the following component

[11] Sometimes, the terms concept and type are used interchangeably.

[12] In this section and subsequent sections we will be describing the internal representations. There is a user-friendly interface for defining these knowledge representations.

[13] Details of the COIN implementation have been described in other papers [4, 7]. An example is presented in the next section to illustrate many of these points.

conversion definition can convert between any contexts of weight units:

$$\forall X : weight \mid\!- X[cvt(weightUnit, C2)@c1, u \rightarrow v] \leftarrow X[weightUnit(C1) \rightarrow C_f] \wedge X[weightUnit(C2) \rightarrow C_t] \wedge unit\_conv\_(F, T, R) \wedge F^{C2} = C_f \wedge T^{C2} = C_t \wedge R[value(C2) \rightarrow r] \wedge v = u * r.$$

Once all contexts are defined and the component conversions for each modifier are specified, a receiver in any context can query any data source in other context as if they were in the same context. The mediator automatically recognizes context differences and composes a conversion using the component conversions on the fly.

We will demonstrate the key features of COIN using the intelligence information integration scenario. Figure 3 shows an interface of the prototype. A receiver uses the system by supplying queries and identifying the desired context [14]. Any defined contexts, including source contexts, can be used as the receiver context. For demonstration purposes, this interface allows users to step through different mediation stages. Results are shown in the Result section.

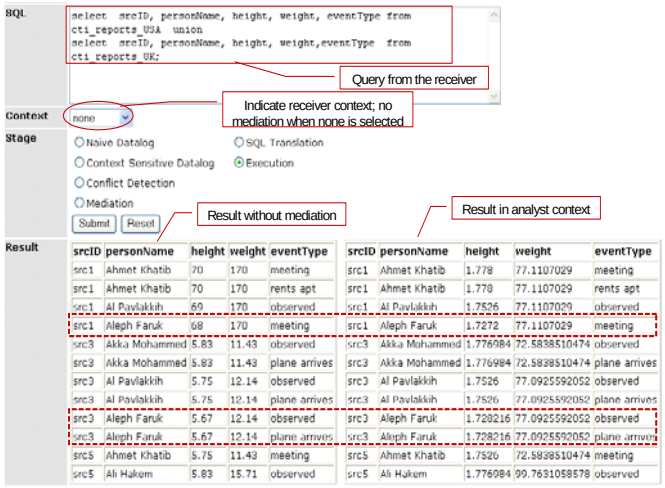

Figure 3. COIN prototype for intelligence information integration

In Figure 3 we show a query to two intelligence data sources[15], one in the USA context, the other in the UK context. Results without mediation are shown in the lower left of the figure. Note the data records in dashed rectangles. When comparing the height and weight data, it seems that the person named Aleph Faruk in src1 (which is in USA context) is different from the person with the same name in src3 (which is in the UK context). After the results are converted by the mediator by choosing a desired context, e.g., in the Analyst context, as shown in lower right of the figure, the height and weight data from two sources are almost identical. This will help the analyst to synthesize information to draw important conclusions, e.g., Aleph Faruk arrived by plane to have a meeting. Data conversions are introduced automatically by the mediator during query execution.

In addition to converting data from different contexts into the desired context, the mediator also has explication tools such as reporting detected semantic differences and generating mediated queries as intensional answers to the original query. For example, when the receiver in the Analyst context issues the following query to combine data from two sources:

```
select personName, height, weight, geoCoord,
cityName, airport, countryCode, eventDate,
eventType from cti_reports_UK  union
select personName, height, weight,  geoCoord,
cityName, airport, countryCode, eventDate,
eventType from cti_reports_USA;
```

at the Conflict Detection stage, the mediator reports all detected semantic differences as shown in Figure 4.

SemanticType	Modifier	Modifier value in source context	Modifier value in target context
eventDate	dateFmt	c_UK : European Style /	c_Analyst : American Style -
countryCode	ctryCodeStd	c_UK : ISO3166A2	c_Analyst : ISO3166A3
airportCode	aptSymType	c_UK : ICAO	c_Analyst : IATA
geoCoord	geoCoordCode	c_UK : BNG-OGB7	c_Analyst : UTM-WGS84
Weight	weightUnit	c_UK : stone	c_Analyst : kg
Height	lengthUnit	c_UK : ft	c_Analyst : m

SemanticType	Modifier	Modifier value in source context	Modifier value in target context
eventDate	dateFmt	c_USA : American Style /	c_Analyst : American Style -
countryCode	ctryCodeStd	c_USA : FIPS	c_Analyst : ISO3166A3
geoCoord	geoCoordCode	c_USA : MGRS-WGS84	c_Analyst : UTM-WGS84
weight	weightUnit	c_USA : lb	c_Analyst : kg
height	lengthUnit	c_USA : in	c_Analyst : m

Figure 4. Semantic differences detected by the mediator

The first table in the figure displays the semantic difference between the UK context and the Analyst context, while the second table shows the differences between the USA context and the Analyst context. Comparing the detected differences here with those summarized in Table 3 indicates that all semantic differences are correctly identified. For example, weight is expressed in stones in the UK context while it is in kg in the Analyst context; because both the USA context and the Analyst context use the same airport code standard, the airport code difference shown in the first table does

[14] In actual operation, this interface would not be used. The receiver would incorporate what would seem like normal SQL database calls within his/her application programs (including within cells in an Excel spreadsheet). Also, in such cases, the receiver context would normally be a constant for that organization.

[15] Due to space limitation, we demonstrate the results of using only two sources and a subset of the data elements.

not appear in the second table. In fact, for the same query, if the desired context is USA context, the second table will be empty.

The mediated query in Datalog syntax is shown in Figure 5. All semantic differences shown in Figure 4 are reconciled by the conversions automatically composed by the mediator. For example, the unit of measure difference for weight between UK context and the Analyst context is reconciled by using the *unit_conv* conversion function, which returns a conversion ratio (V15 indicated by a rectangle). The weight value in UK is V14 (indicated by an oval), which is multiplied by the conversion ratio to obtain V24 (in double-lined rectangle), which is *kg* as desired by the Analyst. Other semantic differences are reconciled similarly.

```
answer('V26', 'V25', 'V24', 'V23', 'V22', 'V21', 'V20', 'V19', 'V18'):-
        unit_conv("ft", "m", 'V17'),
        'V25' is 'V16' * 'V17',
        unit_conv("stone", "kg", 'V15'),
        'V24' is 'V14' * 'V15',
        cti_geoTran_convert2("BNG-OGB7-X", 'V13', "MGRS-WGS84-X", 'V23'),
        airporticao('V12', 'V21', 'V11'),
        cti_ctrycode('V10', 'V9', 'V8', 'V20', 'V7'),
        datexform('V6', "European Style /", 'V19', "American Style -"),
        cti_reports_UK('V5', 'V4', 'V8', 'V22', 'V12', 'V3', 'V13', 'V26',
        'V16', 'V14', 'V18', 'V6', 'V2', 'V1').

answer('V24', 'V23', 'V22', 'V21', 'V20', 'V19', 'V18', 'V17', 'V16'):-
        unit_conv("in", "m", 'V15'),
        'V23' is 'V14' * 'V15',
        unit_conv("lb", "kg", 'V13'),
        'V22' is 'V12' * 'V13',
        cti_geoTran_convert2("geodetic-WGS84-X", 'V11', "MGRS-WGS84-X", 'V21'),
        cti_ctrycode('V10', 'V9', 'V8', 'V18', 'V7'),
        datexform('V6', "American Style /", 'V17', "American Style -"),
        cti_reports_USA('V5', 'V4', 'V9', 'V20', 'V19', 'V3', 'V11', 'V24',
        'V14', 'V12', 'V16', 'V6', 'V2', 'V1').
```

Figure 5. Mediated query

When the same query is issued by a receiver in other contexts, the appropriate mediated query will be generated accordingly. For example, Figure 6 show the mediated query when the desired context is USA. Note that first sub-query now consists of necessary conversions between the UK context and USA context, e.g., weight conversion converts from *stone* to *lb*. The second sub-query does not include any conversion at all, because the source is already in the receiver context.

```
answer('V26', 'V25', 'V24', 'V23', 'V22', 'V21', 'V20', 'V19', 'V18'):-
        unit_conv("ft", "in", 'V17'),
        'V25' is 'V16' * 'V17',
        unit_conv("stone", "lb", 'V15'),
        'V24' is 'V14' * 'V15',
        cti_geoTran_convert2("BNG-OGB7-X", 'V13', "geodetic-WGS84-X", 'V23'),
        airporticao('V12', 'V21', 'V11'),
        cti_ctrycode('V10', 'V20', 'V9', 'V8', 'V7'),
        datexform('V6', "European Style /", 'V19', "American Style /"),
        cti_reports_UK('V5', 'V4', 'V9', 'V22', 'V12', 'V3', 'V13', 'V26',
        'V16', 'V14', 'V18', 'V6', 'V2', 'V1').

answer('V14', 'V13', 'V12', 'V11', 'V10', 'V9', 'V8', 'V7', 'V6'):-
        cti_reports_USA('V5', 'V4', 'V8', 'V10', 'V9', 'V3', 'V11', 'V14',
        'V13', 'V12', 'V6', 'V7', 'V2', 'V1').
```

Figure 6. Mediated query when receiver is in USA context

We have shown with this demonstration that the COIN approach overcomes the shortcomings of traditional approaches. That is, with COIN, the sources are not required to make any change or commit to any standard; they only need to record data semantics declaratively. Only a small number of component conversions need to be defined declaratively, which are used by the mediator to compose necessary conversions automatically. Changes in the sources can be accommodated by updating context definitions, no hand-written code need to be maintained. These features will be discussed further in the next section.

5. Adaptability, Extensibility, and Scalability – Comparison of Different Approaches

5.1 Adaptability and Extensibility Analysis

Adaptibility refers to the capability of accommodating changes, such as semantic changes within a data source (e.g., if UK changes its weight unit from stones to kg). *Extensibility* refers to the capability of adding or removing data sources with minimal effort. We use the term *flexibility* to collectively refer to the two properties.

The Brute-force (BF) data conversion approach has the least flexibility. With N sources, a change in any source would affect $2(N\text{-}1)$ conversion programs, i.e., N-1 conversion programs converting from the changing source to the other sources and vice versa. Adding or removing a source has similar effects.

This problem is somewhat reduced with the Interchange Standardization (IS) approach. But it still requires re-programming to handle changes, which can be tedious and error-prone. Especially when the interchange standard is changed, all the N sources need to be updated to accommodate the change. All hard-wiring approaches require the reconciliation of all semantic differences to be pre-determined and implemented in conversion programs. As a result, they lack flexibility.

The Global Data Standardization (GS) approach also lacks flexibility because any change requires agreement by all sources, which is difficult and extremely time consuming. Because it requires all systems to implement the changes, it sometimes causes disruption in operations.

In contrast, the ontology and context based COIN approach overcomes this problem. COIN has several distinctive features:

- It only requires that the individual contexts and individual conversions between a modifier's values (e.g., how to convert between currencies) be described declaratively. Thus it is flexible to accommodate changes because updating the declarations is much simpler than rewriting conversion programs (e.g., it is merely necessary to indicate that a source now reports in Euros instead of French Francs).
- The customized conversion between any pair of sources (as many conversion programs as are needed) is composed automatically by the mediator using conversions of the relevant modifiers.
- COIN is able to compose all the conversion in BF, but without the burden of someone having to

manually create and keep up-to-date all the pair-wise conversion programs.

- The COIN approach also avoids the multiple or unnecessary conversions that arise from the IS approach since the conversion programs that it generates only includes the minimally required conversions, including no conversions for certain (or all) modifiers, if that is appropriate[16].

As we will see from the next section, the COIN approach significantly reduces the number of pre-defined conversion components so that it can scale well when a large number of sources need to exchange information.

5.2 Scalability Analysis

In order for information from other sources to be correctly interpreted, it is critical to convert data into the desired context. The number of conversions that needs to be implemented and maintained over time is an appropriate measurement for the scalability of an integration approach. Our scalability analysis will focus on the number of conversions needed in each approach. The GS approach is scalable because it does not need any conversion at all. But it is often impossible to establish a global standard in large scale integration effort. We have informally discussed the scalability of the two other traditional approaches. We will summarize them followed by a detailed analysis on the scalability of the COIN approach.

Proposition 1 - Scalability of BF. With N data sources, the number of conversions for BF is N(N-1), which is $O(N^2)$.

Explanation: Each source needs to perform translations with the other N-1 sources; there are N sources, thus a total of $N(N\text{-}1)$ translations need to be in place to ensure pair-wise information exchange, which is $O(N^2)$.

Proposition 2 - Scalability of IS. With N data sources, the number of conversions for IS is 2N, which is $O(N)$.

Explanation: For each source there is a conversion to the standard and another conversion from the standard to the source. There are N sources, so the total number of conversions is $2N = O(N)$.

Proposition 3 - Scalability of COIN. With N data sources and an ontology that has m modifiers with each having n_i unique values, $i \in [1, m]$, the number of conversions for COIN is $O(mn_k^2)$, where $n_k = \max\{n_i \mid i \in [1, m]\}$; when m is fixed, the number of conversions defined in COIN is $O(n_k^2)$.

Explanation: As seen earlier, conversions in COIN are defined for each modifier, not between pair-wise sources. Thus the number of conversions depends only on the variety of contexts, i.e., number of modifiers in the ontology and the number of distinct values of each modifier. In worst case, the number of conversions to be defined is $\sum_{i=1}^{m} n_i(n_i - 1)$, where n_i is the number of unique values of the i^{th} modifier in the ontology, which is not to be confused with the number of sources; m is the number of modifiers. This is because in worst case for each modifier, we need to write a conversion from a value to all the other values and vice versa, so the total number of conversions for the i^{th} modifier is $n_i(n_i\text{-}1)$. Let n_k=max(n_1, …, n_m). When both m and n_k approach infinity, $\sum_{i=1}^{m} n_i(n_i - 1) = O(mn_k^2)$; for $\forall m$, $\sum_{i=1}^{m} n_i(n_i - 1) = O(n_k^2)$, as $n_k \to \infty$.

However, in the intelligence information example, and in many practical cases, the conversion functions can be parameterized to convert between all values of a modifier. For instance, the weight unit conversion given in Section 4 can convert between any two units of measure using the external relation *unit_conv*. The conversion functions for many other modifiers are also of this nature. Thus, only 6 of these parameterized conversion functions are necessary for converting between contexts that differ in weight, height, airport code, country code, geo-coordinate, and/or date format. The COIN approach can take advantage of these general functions because the overall conversion program between any two contexts is automatically generated[17].

When parameterization is difficult, we can exploit certain relationships among component conversion functions. In cases where the set of component conversions are essentially a set of inter-related equations, COIN can generate undefined conversions using its symbolic equations solver [3, 4] to reduce the number of conversion component declarations needed. For example, suppose we have three definitions for price: (A) base price, (B) tax included price, and (C) tax and shipping & handling included price. This can be modeled by using a modifier that has three unique values for *price* concept in the ontology. With known equational relationships among the three price definitions, and only two conversions (1) from base_price to base_price+tax (i.e., A to B) and (2) from base_price+tax to base_price + tax + shipping & handling (i.e., B to C), the COIN mediator can compute the other four conversions automatically (A to C and the three inverses). Thus the number of conversion definitions for a modifier can be reduced from $n(n\text{-}1)$ to n-1, where n is the number of unique values of the modifier. For price

[16] For example, if the global standard for currency was USDollar and the source context was UKPounds and the receiver context was UKPounds, the IS approach would convert UKPounds-> USDollar and then USDollar-> UKPounds. The COIN approach would not do any conversions for currency amounts.

[17] Note that this does not eliminate the need for manual effort for the hard-wiring approaches because all the pair-wise conversions still need to be programmed, even if most of the programs merely make calls to general functions using different parameters.

in the example, n=3, so we only need 2 conversion components. Thus we have the following proposition:

Proposition 4 – Scalability of COIN (parameterization and invertible conversion). When conversions can be parameterized, COIN requires m conversions. Otherwise, if the conversions are invertible functions, COIN needs $\sum_{i=1}^{m}(n_i - 1)$ conversions.

Furthermore, declaring the contexts can be simplified since contexts can be inherited with optional overriding in COIN. This significantly reduces the number of necessary declarations. For example, we can define a context k for a country because most agencies in the same country share the same context. If an agency in the country differs from the other agencies only with regard to say, weight unit, we can define its context as k' and specify only the particular weight unit in k'; by declaring k' as a sub-context of k, k' inherits all the other context definitions for context k. This keeps the size of the knowledge base compact when the number of sources grows. In addition, subtypes in the ontology inherit the modifiers and the conversion definitions of their parent types, which also helps keep the number of conversion component definitions small.

Table 4. Number of conversions to achieve semantic interoperability among 150 sources

Approach	General case	In the example
Brute Force (BF)	$N(N\text{-}1)$, N:= number of sources and receivers	22,350
Interchange Standard (IS)	$2N$, N:= number of sources and receivers	300
Context Interchange (COIN)	1) Worst case:, $\sum_{i=1}^{m} n_i(n_i - 1)$ n_i:= number of unique values of i^{th} modifier, m := number of modifiers in ontology 2) $\sum_{i=1}^{m}(n_i - 1)$ when equational relationships exist 3) m, if all conversions can be parameterized	1) worst: 56 2) actual number: 6

Table 4 summarizes the scalability of different approaches in terms of the number of conversions that need to be specified. Even in the worst case, the COIN approach requires significantly less conversions than the BF or IS approaches.

Recent research [14] extended COIN to represent and reason about semantics that change over time. For example, when comparing historic stock prices in different exchanges, some of them changed the currency assumed in the reporting (e.g., changed from reporting in French Francs to Euros). With the formalism and the mediation engine, these temporal changes can be captured and the semantic differences at different times (in addition to between different sources) can be automatically recognized and reconciled at run time. With these advanced features and its flexibility and scalability, COIN is ideal for large scale information integration.

6. Conclusion

Integrating information from diverse heterogeneous systems is one of the key challenges today. Any viable solution must be flexible and scalable in reconciling semantic differences amongst these information sources. In this paper, we described the COIN approach to this challenge. Our analysis shows that the COIN approach can efficiently handle large number of semantic conflicts and is flexible and scalable to meet the evolving requirements.

Acknowledgements: This work has been supported, in part, by MITRE Corp., the MIT-MUST project, the Singapore-MIT Alliance, and Suruga Bank.

References

[1] S. Bressan, C. Goh, N. Levina, S. Madnick, A. Shah, M. Siegel, "Context Knowledge Representation and Reasoning in the Context Interchange System", Applied Intelligence: The International Journal of Artificial Intelligence, Neutral Networks, and Complex Problem-Solving Technologies, 12(2), pp. 165-179, 2000.

[2] D. Doughty (2004) "The Achilles' Heal of Service-Oriented Architectures", Business Integration Journal, September, 44-47

[3] A. Firat, S.E. Madnick, B. Grosof, "Financial Information Integration In the Presence of Equational Ontological Conflicts", Proceedings of the Workshop on Information Technology and Systems (WITS), Barcelona, Spain, December 14-15, 2002, pp. 211-216.

[4] A. Firat, "Information Integration using Contextual Knowledge and Ontology Merging," PhD Thesis, MIT, 2003.

[5] T. Frühwirth, "Theory and Practice of Constraint Handling Rules," J. of Logic Programming, 37, 95-138, 1998.

[6] Gallaher, M. P., O'Connor, A. C., Dettbarn, J. L. and Gilday, L. T. (2004) "Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry", GCR 04-867, NIST

[7] C.H. Goh, "Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Systems", PhD Thesis, MIT, 1997.

[8] C.H. Goh, S. Bressan, S. Madnick, M. Siegel, "Context Interchange: New Features and Formalisms for the Intelligent Integration of Information", ACM Trans. on Information Systems (TOIS), 13(3), 270-293, July 1999.

[9] A.C. Kakas, A. Michael, and C. Mourlas, "ACLP: Integrating Abduction and Constraint Solving," Journal of Logic Programming, 44, pp. 129-177, 2000.

[10] M. Kiffer, G. Laussen, J. Wu, "Logic Foundations of Object-Oriented and Frame-based Languages", J. ACM, 42(4), pp. 741-843, 1995.

[11] R. Miller, M.A. Malloy, E. Masek (2003) “Transforming Tactical Messaging: Exploiting Web Information Standards for Interoperability”, Intercom, 44(1), 50-51.

[12] Rosenthal, L. Seligman, S. Renner (2004) “From Semantic Integration to Semantics Management: Case Studies and a Way Forward”, ACM SIGMOD Record, 33(4), 44-50.

[13] H. Wache, T. Vogele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, S. Hubner, "Ontology-Based Integration of Information – A Survey of Existing Approaches", Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing, Seattle, USA, 4 –5 August, 2001.

[14] Zhu, H., Madnick, S., and Siegel, M. (2004) “Reasoning about Temporal Context using Ontology and Abductive Constraint Logic Programming”, Workshop on Principles and Practices of Semantic Web Reasoning (PPSWR04), Saint Malo, France, Sep 8-9, in LNCS 3208, 90-101.

[15] H. Zhu, S.E. Madnick (2004) “Context Interchange as a Scalable Solution to Interoperating Amongst Heterogeneous Dynamic Service”, 3rd Workshop on E-Business, December 11, 2004, Washington, D.C., 150-161.