

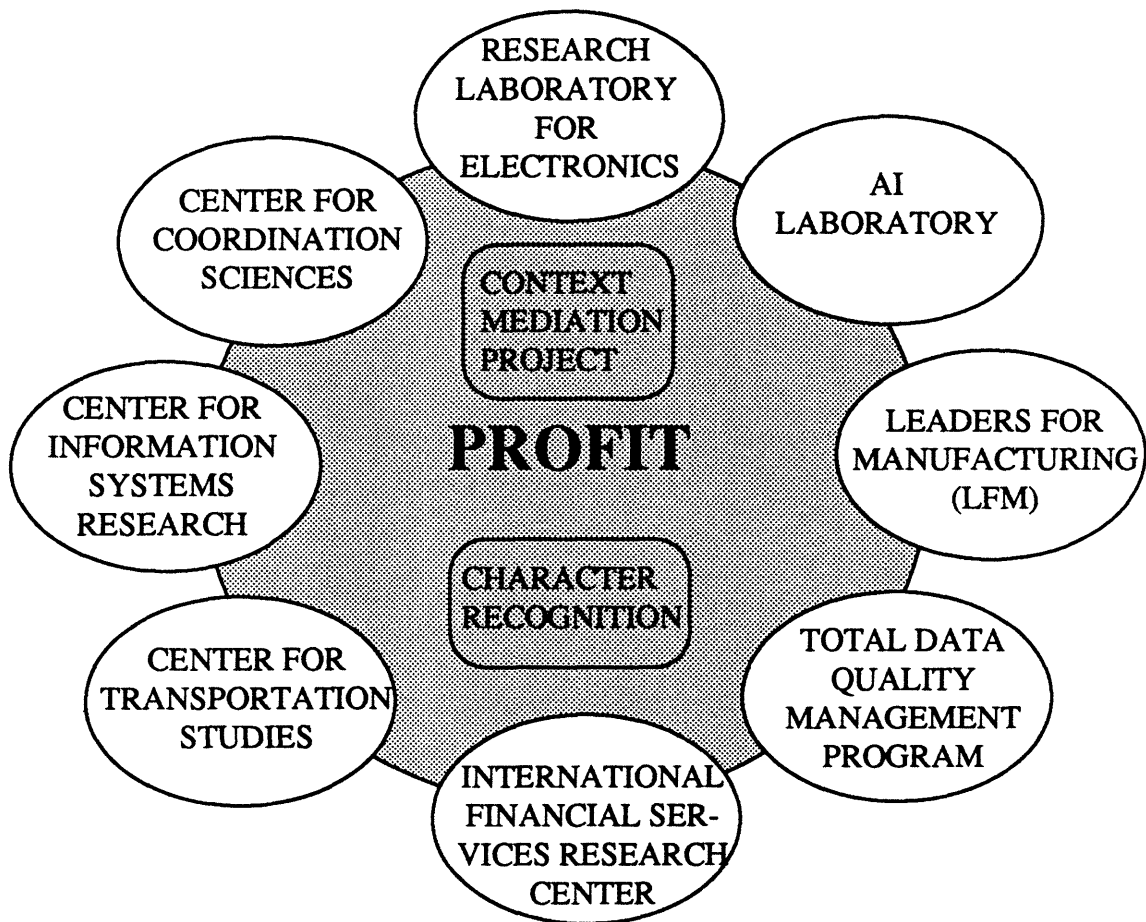
**An Ontological and Semantical
Approach to Source-Receiver Interoperability**

Jacob Lee
Michael D. Siegel

WP #3777 October 1994
PROFIT #94-16

Productivity From Information Technology (PROFIT)

The Productivity From Information Technology (PROFIT) Initiative was established on October 23, 1992 by MIT President Charles Vest and Provost Mark Wrighton "to study the use of information technology in both the private and public sectors and to enhance productivity in areas ranging from finance to transportation, and from manufacturing to telecommunications." At the time of its inception, PROFIT took over the Composite Information Systems Laboratory and Handwritten Character Recognition Laboratory. These two laboratories are now involved in research related to context mediation and imaging respectively.



In addition, PROFIT has undertaken joint efforts with a number of research centers, laboratories, and programs at MIT, and the results of these efforts are documented in Discussion Papers published by PROFIT and/or the collaborating MIT entity.

Correspondence can be addressed to:

The "PROFIT" Initiative
Room E53-310, MIT
50 Memorial Drive
Cambridge, MA 02142-1247
Tel: (617) 253-8584
Fax: (617) 258-7579
E-Mail: profit@mit.edu

*To appear in the Proceedings of the Workshop On Information
Technologies and Systems 1994*

**An Ontological and Semantical
Approach to Source-Receiver
Interoperability**

October 1994

Jacob L. Lee*
Michael D. Siegel**

* Rm E53-322
Sloan School of Management
MIT
Cambridge, MA 02139
lhlee@mit.edu

** Rm E53-322
Sloan School of Management
MIT
Cambridge, MA 02139
msiegel@mit.edu

An Ontological and Semantical Approach to Source-Receiver Interoperability

Jacob Lee
Sloan School of Management
MIT
Cambridge, MA 02139
lhlee@mit.edu

Michael Siegel
Sloan School of Management
MIT
Cambridge, MA 02139
msiegel@mit.edu

Abstract

In this paper, we propose an approach to address the issue of semantic interoperability between a data *source* and a data *receiver* in the framework of the *context interchange architecture*. A key component in this architecture is the *context mediator*, an intelligent agent which performs data conversions between the source and the receiver. In this paper, we are concerned with what a context mediator 'knows' and how it 'behaves' from an external perspective. We introduce the notion of a *conversion axiom*, which is a formal and declarative means of specifying knowledge required by the mediator to perform desired conversions. We also state, in formal terms, the behavior of the mediator which uses conversion knowledge to translate data from the *source context* to the *receiver context*. Formal characterizations of what a context mediator should know and its external behavior provide a specification for the subsequent design of the knowledge representation and reasoning processes internal to the mediator. This approach draws upon ideas and concepts discussed in Mario Bunge's Ontology and Semantics.

1. Introduction

The source-receiver model, in the framework of the *context interchange architecture*, has been used in analyzing the problem of integrating multiple autonomous databases [4, 5, 6, 7, 8]. A source can, for example, be a relation or a database while a receiver can be an application query. The source-receiver model is shown in Fig. 1. A key component in this architecture is the *context mediator*, an intelligent agent which performs data conversions between a source and a receiver. In this paper, we take an external perspective of the context mediator. We are concerned with what a context mediator 'knows' and how it 'behaves'. We introduce the notion of a *conversion axiom*, which is a formal and declarative means of specifying knowledge required by the mediator to perform desired conversions. Examples of how conversion axioms may be specified for various types of conversions will be presented. We will also state, in formal terms, the behavior of the mediator which uses conversion knowledge to translate data from the *source context* to the *receiver context*. Formal characterizations of what a context mediator should know and its external behavior provide a specification for the subsequent design of the knowledge representation and reasoning processes internal to the mediator. These techniques and concepts although described in terms of the source-receiver model can also be generalized to multiple autonomous heterogeneous database systems.

Analysis of this model can be separated into two related but distinct levels: the *symbol* and *construct* level. Symbols (e.g. character strings) are physical objects that stand for constructs. Constructs on the other hand are the meaning assigned to the symbols and have no physical existence apart from

mental processes [2: pp. 21-23]. In cases where there is potential ambiguity, we will use double quotes " " to indicate constructs. At the symbol level, we are concerned with how constructs are represented by symbols in a non-uniform manner (i.e. *symbolic heterogeneity*). Examples of such heterogeneities are naming differences leading to homonyms and synonyms, format differences and structural differences [1]. In this paper, we move beyond the symbol level to the construct level where we are primarily concerned with issues related to the comparison and conversion of constructs (i.e. *construct heterogeneity*). Construct heterogeneities can exist even within a common symbolic representation. Differing monetary units is an example of a construct heterogeneity. Construct heterogeneity has important implications for achieving source-receiver interoperability.

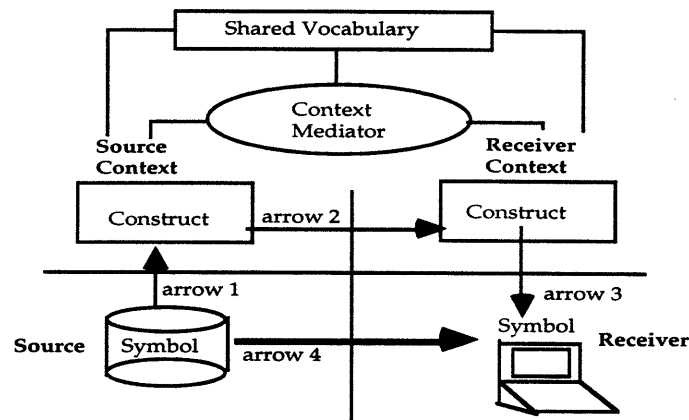


Fig. 1 Source-Receiver Model

An information agent (i.e. source or receiver) has an associated set of constructs referred to as its *context*. A source context defines the set of possible constructs that may be represented by a data source. The receiver context defines the set of constructs that are meaningful or acceptable to the receiver. One type of construct in a context is the *statement*¹. A statement is a proposition or fact to which we can attach a truth value. A data source represents some collection of statements or facts about the world which may be retrieved by a receiver. A statement in a source context will not always be meaningful to a receiver. Often, in order for the receiver to 'understand' the source statement, it has to be converted to an 'equivalent' statement that exists in the receiver's context. For instance, the statement "The latest trade price of the IBM stock is 80 USdollars" which is represented by a database may not be meaningful to an application which expects all monetary amounts to be in Yen currency. In this case, the source statement should be converted to "The latest trade price of the IBM stock is 8800 Yen" (assuming a conversion rate of 110) which can then be processed meaningfully by the receiver. Given a source context and a receiver

¹The other types of constructs in a context are *predicates* and *predicate arguments*. These constructs will be further explained in the discussion on context in Section 2.2.2.

context, the flow of data from a source to a receiver in response to a query² can be analyzed in three stages as shown in Fig. 1. First, symbols in the source are mapped to a corresponding collection of statements in the source context (arrow 1). Second, these source statements are then converted to 'equivalent' statements in the receiver context (arrow 2). Third, these statements in the receiver context are then mapped onto their symbolic counterparts on the receiver side (arrow 3). Arrow 4 represents the overall process. In this paper, we restrict our analysis to arrow 2 which represents a fundamental and essential aspect of source-receiver interoperability. We are concerned with the *convertibility* of statements in the source context to 'equivalent' statements in the receiver context.

In Section 2, we introduce some basic concepts and ideas from Mario Bunge's Semantics [2] and Ontology [3] which will serve as a conceptual foundation for the research presented here. In Section 3 we apply these concepts to context interchange. Specifically, we define the notion of a context and a *shared vocabulary*. We then discuss conversion axioms and give examples of how these may be specified. We also give a formal description of the context mediator in these terms. In Section 4 we discuss the implications for further research on and the design of systems based on the context interchange architecture. This paper concludes with Section 5.

2. Bunge's Ontology and Semantics: A Brief Introduction

2.1 Bunge's Ontology

According to Bunge '...an ontology is not a set of things but a philosophical theory concerning the basic traits of the world' [2: p. 38]. In Bunge's Ontology, the world is made up of *things*. Things possess *properties*. We perceive the properties of things via *attributes*. That is, we only know properties as attributes. A property is a feature that a thing possesses even if we are ignorant of this fact. On the other hand, an attribute is a feature we assign to a thing [3: p. 58]. Therefore there is a distinction between an attribute and a property of a thing. An attribute, (e.g. weight) represents a *property in general* while an attribute *value* (e.g. 36 kilograms) represents a *property in particular*. A collection of objects possessing the same property in particular is referred to as a *class*.

There are simple and complex things. A complex thing, or *system*, is made up of simpler things that *interact*. There is a part-whole relationship between a system and its components. An example of a system is the computer which is made up of simpler, interacting things such as the CPU, memory etc. (Fig. 2). This is called an *aggregation* hierarchy. Complex things have *inherited* and *emergent* properties, and hence, emergent and inherited attributes. An inherited attribute is an attribute common to a system and one of its components and share the same attribute value. For example, the clock speed of computer is an inherited attribute because it is also the clock speed of the computer's CPU. An emergent attribute is

²For the purpose of this abstract we ignore issues of data flow in the other direction from receiver to source (e.g. query conversion).

an attribute of a complex thing but not of any of its components (e.g. the weight of the computer). We denote the relationship between a particular CPU "a" and its respective computer "b" as APO (a, b, P) where P is the inherited attribute.

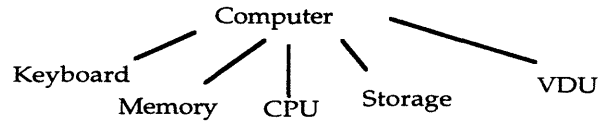


Fig. 2 Aggregation Hierarchy

The *state* of a thing is represented by a combination of its attribute values. For example, the title of an employee in an organization refers to a state of an employee. An *event* is a change in the state of a thing and is represented as an ordered pair of states. For example, a promotion is an event where the title of an employee changes. For a more detailed discussion on Bunge's Ontology and its application to information systems, the reader is referred to [9, 10, 11, 12, 13].

2.2 Bunge's Semantics

2.2.1 Statements, Predicates and Predicate Arguments

Bunge describes the study of Semantics as '...concerned not only with linguistic items but also, and primarily, with the constructs such items stand for and their eventual relation to the real world' [2: p. 2]. Three types constructs we are concerned with in Bunge's Semantics are statements, *predicates* and *predicate arguments*.

An *n*-ary predicate P is a function

$$P: A_1 \times A_2 \times \dots \times A_n \rightarrow S$$

from domains A_1, A_2, \dots, A_n of predicate arguments to a set of statements S. In other words, predicates are functions that evaluate to statements when the argument list is instantiated. Statements are propositions and can be assigned truth values³. Predicate arguments (constants or variables) can be classified according to the following types (1) *object*, (2) *property*, (3) *space*, (4) *time*, (5) *unit* and (6) *scale* [2: p. 40].

The object variable ranges over a class of things. The object constant refers to an individual thing. The property variable ranges over a set of attribute values while a property constant corresponds to an attribute value. Time and space values indicate the temporal and physical location of a thing. For clarity, the type of the argument is indicated by '/' followed by the argument type in italics when it is useful to do so. Furthermore, arguments preceded by '?' are variables, otherwise, they are constants. As an example, the predicate "TradePrice (?x/*object*, ?y/*property*, ?u/*unit*, ?t/*time*)" evaluates to "TradePrice of ?x (a stock) is ?y (some value) in ?u units (some currency) for a particular period of time ?t" where

³Note that this definition of a predicate differs from that in predicate calculus in that the range is a set of statements instead of Boolean constants (i.e. True and False).

"TradePrice" is the attribute. The evaluation of predicates to statements is an important issue and a further discussion is presented in Section 3.

One useful property of space-time variables not discussed by Bunge but which will prove useful is the concept of *granularity*. Consider for example the day 4 July 1987. This date refers to a 24-hour period *within* the month of July 1987 which is *within* the year 1987. This is an example of a granularity hierarchy of time (Fig. 3a). The smallest granularity of time is an *instant*. No time periods can therefore appear below an instant in a granularity hierarchy of time. Similarly, Chicago is in the USA is an example of a granularity hierarchy of space (Fig. 3b). The smallest granularity of space is a *point*. No space values can appear below a point in a granularity hierarchy of space. We will denote the granularity relationship among time values and space values as InTime (a,b) and InSpace (c,d) respectively where "a" is within "b" and "c" is within "d".

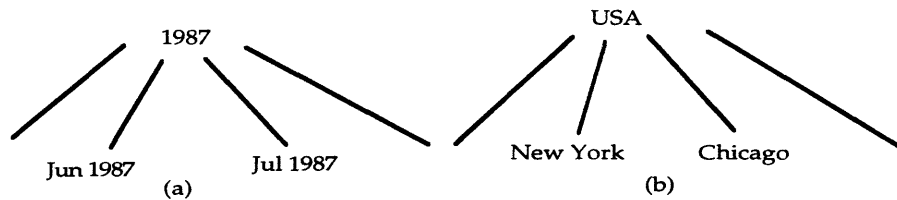


Fig 3. Granularity Hierarchy of (a) Time and (b) Space

Conversion functions between units and between scale factors are also important relationships not discussed by Bunge but which are needed for statement conversions. We therefore define two relations. The relation ConvUnit is defined over $D_1 \times D_2 \times D_3$ where D_1 and D_2 are domains of units and D_3 is a domain of numbers and $\langle ?x, ?y, ?z \rangle \in \text{ConvUnit}$ iff $?z$ is the unit conversion factor from $?x$ to $?y$. The relation ConvScale is defined over $D_1 \times D_2 \times D_3$ where D_1, D_2 and D_3 are domains of numbers and $\langle ?x, ?y, ?z \rangle \in \text{ConvScale}$ iff $?z = ?x + ?y$.

2.2.2 Context, Import, Purport and Intension

Bunge defines a *context* as the ordered triple $\langle S, P, D \rangle$ where P is a set of predicates, D is the collection of all the arguments of the predicates and S is a set of statements formed by instantiating the predicates in P with arguments from D . A source context is the space of possible constructs that may be represented by a data source, while a receiver context is the space of constructs that a receiver 'understands'.

The relationships among statements in S and predicates in P are formalized through the notion of *sense*. There are three dimensions of the notion of *sense*: *import*, *purport* and *intension* [2: pp. 115-118]. The purport of a predicate is a set of predicates that "define" or prove it [2: p. 146]. The import of a predicate is the set of predicates which, possibly in conjunction with other statements predicates, it *implies*. Consider the formula $\text{ParentOf}(?x, ?y) \text{ AND } \text{ParentOf}(?x, ?z) \text{ AND } (?x \neq ?y) \Rightarrow \text{SiblingOf}(?y, ?z)$. $\text{ParentOf}(?x, ?y) \text{ AND } \text{ParentOf}(?x, ?z) \text{ AND } (?x \neq ?y)$ is the purport of $\text{SiblingOf}(?y, ?z)$ and $\text{SiblingOf}(?y,$

?z) is the import of ParentOf(?x,?y) AND ParentOf(?x,?z) AND (?x≠?y). Similarly ParentOf(Peter,Mary) AND ParentOf(Peter,Paul) AND (Mary≠Paul) ⇒ SiblingOf(Mary, Paul) is an example of the purport-import relationship among statements in S.

The intension of a predicate is the set of predicates which it *subsumes*. In Bunge's terms, a more specific predicate subsumes a more general one. We will make adapt the notion of subsumption more precisely as follows: Consider two predicates with predicate symbols P and Q and domains D_P and D_Q respectively. Then the predicate with the predicate symbol P subsumes the predicate with predicate symbol Q if and only if $D_P \subseteq D_Q$ and $P(?x_1, \dots, ?x_n) \Rightarrow Q(?x_1, \dots, ?x_n)$. For example FatherOf (?x, ?y) ⇒ ParentOf(?x, ?y). We represent subsumption relationships between predicates with the relation ISA where $\langle P, Q \rangle \in ISA$ if and only if P subsumes Q. Similarly, FatherOf (Peter,Paul) ⇒ ParentOf (Peter,Paul) is a subsumption relationship among statements in S. These relationships will serve as *conversion axioms* for context interchange. Statements in the source context may be converted to statements in the receiver context via logical implication as specified by these axioms.

In summary, a context is made up of constructs which have relationships with each other. At the highest level, a context is made up of a set of predicates P, a set of predicate arguments D and a set of statements S. Predicates and statements are related to one another via purport, import and intensions. The elements of D can be divided into several types: object, property, space, time, unit and scale. There are also various relationships among each of these types as specified by the relations APO, InTime, InSpace, ConvScale and ConvUnit. These various relations are important because, as we illustrate in the next section, they are a convenient means for specifying various classes of conversion axioms.

3. Applying Bunge's Ontology and Semantics to Context Interchange

We propose to use Bunge's formalization of the notion of context i.e. $\langle S, P, D \rangle$. Let $C_S = \langle S_S, P_S, D_S \rangle$ and $C_R = \langle S_R, P_R, D_R \rangle$ be the source and receiver contexts respectively. Intuitively, C_S is the space of constructs that are potentially expressible by the source. At any instant, the set of statements asserted by the source as true is $S_t \subseteq S_S$. Intuitively C_R is the set of constructs that a receiver understands and can accept. S_r is the set of statements that are potential answers to a query. The shared vocabulary is defined as $V = \langle S_V, P_V, D_V, A \rangle$ where $S_S, S_R \subseteq S_V$; $P_S, P_R \subseteq P_V$; $D_S, D_R \subseteq D_V$; A is the set of conversion axioms. We now formally state the behavior of the context mediator:

Given A, S_r and S_t , the context mediator outputs $S_q \subseteq S_r$ that is the set of *logical consequences* of A and S_t .

However, there may be potential problems in specifying conversion axioms. For example, explicitly specifying conversions between all possible pairs of currency units for all predicates representing monetary attributes is clearly not practical. This is where the various relations we defined in

the previous section become useful. Various classes of conversion axioms may be specified simply by specifying these relations. This is because these classes of conversion axioms may be derived from these relations. We will show several derivation rules that allow us to derive conversion axioms from the relations for three classes of statements. We will focus on statements about (1) the attribute values (or state) of things, (2) the class of things and (3) the *instantaneous* events undergone by things. Correspondingly, we identify three types of predicates in P, namely attribute, class and event predicates.

The attribute predicate maps to statements describing the state or attribute value of a thing over a particular period of time. The general structure of such a predicate is "Attribute-Predicate (object, property, unit, scale, time)" which can be evaluated as "The Attribute-Predicate of object is property in unit and scale, for all of time". For example, "Title (John/object, Sales Manager/property, 1987/time)" is evaluated as "The Title of John is Sales Manager for all of 1987". The event predicate maps to statements describing instantaneous events (i.e. changes in states of a thing) that occur at particular instant and at a particular point in space. The general structure of such a predicate is "Event-Predicate (object, time, space)" which can be evaluated as "The Event-Predicate of object is at some instant within time and at some point within space". For example, "Birth (John/object, 4 Jul 1958/time, Chicago/space)" is evaluated as "The Birth of John is at some instant within 4 July 1958 and at some point within Chicago". The class predicate maps to statements describing the class of a thing. The general structure of such a predicate is "Class-Predicate (object)" which can be evaluated as "The object is in the class Class-Predicate". For example, "Male(John/object)" can be evaluated as "John is in the class Male".

We now present the derivation rules. We will use ' \Rightarrow ' to stand for 'implies' or equivalently 'is convertible to'. For clarity, the predicate names and arguments which are being converted in statements are highlighted in **bold**.

Conversion Axioms based on ISA:

IF $\langle P, Q \rangle \in \text{ISA}$ THEN $P(?x, ?y, ?z, \dots) \Rightarrow Q(?x, ?y, ?z, \dots)$ is a conversion axiom.

E.g. Since $\langle \text{LatestTradePrice}, \text{TradePrice} \rangle \in \text{ISA}$, **LatestTradePrice** (IBM/object, 80/property, 1/scale, USDollars/unit) \Rightarrow **TradePrice** (IBM/object, 80/property, 1/scale, USDollars/unit).

Conversion Axioms based on ConvUnit:

IF $\langle ?x, ?y, ?z \rangle \in \text{ConvUnit}$, THEN $P(\dots, ?e/\text{property}, ?x/\text{unit}, \dots) \Rightarrow P(\dots, ?e*?z/\text{property}, ?y/\text{unit}, \dots)$ is a conversion axiom.

E.g. TradePrice (IBM/object, **80**/property, 1/scale, **USDollars**/unit) \Rightarrow TradePrice (IBM/object, **8800**/property, 1/scale, **Yen**/unit).

Conversion Axiom based on ConvScale:

IF $\langle ?x, ?y, ?z \rangle \in \text{ConvScale}$, THEN $P(\dots, ?e/\text{property}, ?x/\text{scale}, \dots) \Rightarrow P(\dots, ?e*?z/\text{property}, ?y/\text{scale}, \dots)$ is a conversion axiom.

E.g. TradePrice (IBM/object, 8800/property, 1/scale, Yen/unit) \Rightarrow TradePrice (IBM/object, 88/property, 100/scale, Yen/unit).

Conversion Axioms based on APO:

IF $\langle ?x, ?y, P \rangle \in \text{APO}$, THEN $P(?x/\text{object}, \dots) \Leftrightarrow P(?y/\text{object}, \dots)$ is a conversion axiom.

E.g. (See Fig. 2). Since APO (CPU/object, Computer/object), Speed (Computer/object, 33/property, 10^6 /scale, Hz/unit) \Leftrightarrow Speed (CPU/object, 33/property, 10^6 /scale, Hz/unit).

Explanation: The CPU is a part of the Computer in this aggregation hierarchy. In this example, the speed of a computer is 33 MHz and speed is an inherited attribute from the CPU. Therefore we can perform conversions both ways.

Conversion Axioms for Attribute Predicates based on InTime:

IF P is an attribute predicate AND $\langle ?x, ?y \rangle \in \text{InTime}$ THEN $P(\dots, ?y/\text{time}, \dots) \Rightarrow P(\dots, ?x/\text{time}, \dots)$ is a conversion axiom.

E.g. (See Fig. 3a). Title (John/object, Sales Manager/property, 1987/time) \Rightarrow Title (John/object, Sales Manager/property, Jun 1987/time)

Explanation: "John was Sales Manager for all of 1987" means that "John was Sales Manager for all of Jun 1987".

Conversion Axioms for Event Predicates based on InTime:

IF P is an event predicate AND $\langle ?x, ?y \rangle \in \text{InTime}$ THEN $P(\dots, ?x/\text{time}, \dots) \Rightarrow P(\dots, ?y/\text{time}, \dots)$ is a conversion axiom.

E.g. (See Fig. 3a). Birth (John/object, 4 Jul 1958/time, Chicago/space) \Rightarrow Birth (John/object, Jul 1958/time, Chicago/space).

Explanation: Since John was born at some instant within the day 4 Jul 1958, it is also true that John was born at some instant within the month Jul 1958.

Conversion Axioms for Event Predicates based on InSpace:

IF P is an event predicate AND $\langle ?x, ?y \rangle \in \text{InSpace}$ THEN $P(\dots, ?x/\text{space}, \dots) \Rightarrow P(\dots, ?y/\text{space}, \dots)$ is a conversion axiom.

E.g. (See Fig. 3b). Birth (John/object, 4 Jul 1958/time, Chicago/space) \Rightarrow Birth (John/object, 4 Jul 1958/time, USA/space).

Explanation: Since John was born at some point within Chicago, John was therefore born at some point in the USA.

Other Classes of Conversion Axioms

There are other classes of conversion axioms which may not be specifiable through such relations and have to be specified directly.

E.g. ParentOf(?x,?y) AND ParentOf(?x,?z) AND (?x \neq ?y) \Rightarrow SiblingOf(?y, ?z).

4. Implications and Future Research

The reliability of data interchange in a federation of information agents based on the context interchange architecture is determined by the correctness and consistency of specifying the shared vocabulary, source and receiver contexts. As a new source or receiver is introduced into an existing federation of interoperable information agents, its context is defined in terms of constructs that are in the shared vocabulary. This task is referred to as *context definition*. The shared vocabulary must therefore be sufficiently rich so that the source and receiver context can be appropriately defined. Methodologies and an application independent specification language need to be developed to facilitate specification.

Specification of contexts and the shared vocabulary must be encoded in some representation for implementation purposes. We assume that the shared vocabulary, source and receiver contexts will share a common representation (i.e. a common language). In all likelihood, this common language will differ from that used by the information agents. Therefore, after the context of an information agent is defined, *vocabulary mappings* need to be specified. Vocabulary mappings relate the symbolic items in the language of an information agent to the common language used to represent its context. It is through the vocabulary mappings that symbolic heterogeneities are resolved. Developing the appropriate representation for contexts and shared vocabulary as well as techniques for specifying vocabulary mappings are further research issues.

Finally, this research has implications for the design of the context mediator. We have already specified the external behavior of the context mediator in terms of conversion axioms, source context, receiver context and the output expected of the mediator. The choice of an appropriate knowledge representation for building the context mediator is an ongoing research issue and will be influenced by this specification.

5. Conclusion

In this paper, we proposed an approach to address the issue of semantic interoperability between a data source and a data receiver in the framework of the context interchange architecture. This approach draws upon ideas and concepts discussed in Mario Bunge's *Ontology and Semantics*. We introduced the notion of a conversion axiom, which is a formal and declarative means of specifying knowledge required by the mediator to perform desired conversions. We gave examples of how conversion axioms may be specified for various types of conversions. We also state, in formal terms, the behavior of the mediator which uses conversion knowledge to translate data from the source context to the receiver context. Formal characterizations of what a context mediator should know and its external behavior provide a specification for the subsequent design of the knowledge representation and reasoning processes *internal* to the mediator. We also discussed implications and future research issues.

References

- [1] Bright, M. W., Hurson, A. R., & Pakzad, S. H. A Taxonomy and Current Issues in Multidatabase Systems. *IEEE Computer*: 50-60, March 1992.
- [2] Bunge, M. *Semantics I: Sense and Reference*. D. Reidel Publishing Company, Boston, 1974.
- [3] Bunge, M. *Ontology I: The Furniture of the World*. D. Reidel Publishing Company, Boston, 1977.
- [4] Goh, C. H., Madnick, S., & Siegel, M. Context Interchange: Overcoming The Challenges of Large-Scale Interoperable Database Systems in a Dynamic Environment. In *Third International Conference on Information and Knowledge Management*, Blatimore, MD, 1994.
- [5] Sciore, E., Siegel, M., & Rosenthal, A. Context Interchange Using Meta-Attributes. In *First International Conference on Information and Knowledge Management*: 377-386, Baltimore, MD, 1992.
- [6] Sciore, E., Siegel, M., & Rosenthal, A. Using Semantic Values to Facilitate Interoperability Among Heterogenous Information Systems. *To appear in Transactions on Database Sytems*, 1994.
- [7] Siegel, M., & Madnick, S. Context Interchange: Sharing the Meaning of Data. *SIGMOD Record*, 20, 4: 77-79, 1991.
- [8] Siegel, M., & Madnick, S. E. A metadata approach to resolving semantic conflicts. In *Proceedings of the 17th International Conference on Very Large Data Bases*: 133-145, Barcelona, Spain, 1991.
- [9] Wand, Y. A Proposal for a Formal Model of Objects. In W. Kim & F. Lochovsky (Eds.), *Object-Oriented Concepts, Databases, and Applications*: 602, ACM Press, New York, N.Y., 1989.
- [10] Wand, Y., & Weber, R. An Ontological Analysis of Some Fundamental Information Systems Concepts. In *Proceedings of the Ninth International Conference on Information Systems*, Minneapolis, Minnesota, USA, 1988.
- [11] Wand, Y., & Weber, R. Mario Bunge's Ontology as a Formal Foundation for Information Systems Concepts. In P. Weingartner & G. J. W. Dorn (Eds.), *Studies on Mario Bunge's Treatise*, Rodopi, Amsterdam, 1990.
- [12] Wand, Y., & Weber, R. An Ontological Model of an Information System. *IEEE Transactions of Software Engineering*, 16, 11: 1282-1292, 1990.
- [13] Wand, Y., & Weber, R. Toward a Theory of the Deep Structure of Information Systems. In *Proceedings of the Twelfth International Conference on Information Systems*, 1991.

Using Ontologies to Facilitate Context Interchange

Cheng Goh (chgoh@mit.edu)

Scott Hofmeister (hofmeist@mit.edu)

Stuart Madnick (smadnick@mit.edu)

Michael Siegel (msiegel@mit.edu)

It is the goal of our project to provide explication and resolution of context differences among information systems and users. Our approach is declarative, it requires that source and receivers describe their context in some common vocabulary. A mediator approach is used to compare these contexts to determine semantic conflicts and, where possible, provide transformations to resolve these conflicts. We hope to simplify the process of developing context descriptions by extending ontologies of concepts that have been developed under other projects. These ontologies will need to be extended so that attributes, meta-attributes (i.e., properties used to define the context or assumptions at the source or receiver), and their values can be mapped to terms in the ontology.

...need to put some meat in here about data objects vs. non-data objects ...perhaps we can include pictures for those who can read postscript....anything else we can do to talk about our needs for ontologies and desing methods for building data objects...stuff etc.....temptations is to spend a minimum amount of time on this and see how it looks...i.e., determine if there is value to send it..