To appear in: Proc 5th Int Conf on CS and Informatics (CS&I'2000), 2000.
http://www.cs.utexas.edu/users/pclark/papers

Semantic Integration of Heterogeneous Information Sources Using a Knowledge-Based System

Thomas Adams[1], James Dullea[1], Peter Clark[2], Suryanarayana Sripada[2], and Thomas Barrett[1]
Boeing Phantom Works, The Boeing Company,
[1]Philadelphia, PA; [2]Seattle, WA
Email: {thomas.l.adams, james.dullea, peter.e.clark, surya.sripada, thomas.m.barrett}@boeing.com

Abstract

A growing number of decision support applications require the ability to formulate ad hoc queries that access information from heterogeneous information sources. The Boeing Company recently conducted a study to investigate the feasibility of applying a high-performance knowledge base to interface the information sources to the decision support system. A commercially available knowledge-based development environment provided a number of useful features for this problem including an expressive knowledge representation language, an efficient inference engine, and a large repository of common-sense knowledge. Our objectives were to determine if the pre-stored knowledge would significantly reduce the time required for knowledge engineering in large scale applications, to determine if the domain model was expressive enough to support the wide range of ad-hoc queries typical of our applications, and to determine if the common-sense knowledge would extend the inference power beyond simply combining information from the available data sources.

Our preliminary results indicate that the upper and middle ontologies do in fact provide a significant knowledge infrastructure that facilitates creation of the domain model. We were able to express a large fraction of the domain in terms of existing concepts with simple extensions. The existing knowledge about human activities, spatial and temporal concepts, and physical and functional decomposition both simplified the generation of the domain model and supported a wide range of ad hoc queries. The knowledge representation language was relatively easy to acquire and the domain models we created are readily extensible.

1. Introduction

This paper describes a knowledge-based approach to semantic integration of heterogeneous information sources. The knowledge-based system we describe consists essentially of two elements—a means for representing domain knowledge and an inference mechanism, which answers queries and performs problem-solving. The knowledge representation scheme itself consists of two elements—a basic knowledge representation language and an ontology. The basic knowledge representation language (CycL) is a highly expressive formalism (in our case, an extension of first-order predicate calculus (FOPC)) that describes facts in the domain in terms of assertions. It serves as a substrate for the ontology in the same sense that a programming language (in our case, LISP) serves as a substrate for the basic knowledge representation language. The ontology includes both the conceptual vocabulary (terms and relations) and the rules and axioms relating terms in the vocabulary. Some researchers prefer to limit the use of the term ontology to simply the conceptual vocabulary and use the term knowledge base to refer to the combination of the ontology and the rules and axioms. This distinction is not important in this paper, since both conceptual vocabulary and rules are written using a common substrate; hence we use the terms ontology and knowledge base more or less interchangeably.

The Boeing Company recently conducted a research effort to study the use of knowledge-based systems to integrate legacy databases. The databases used in this study were three systems that support a military aircraft mechanic. They contain personnel records, maintenance records, and training information. Figure 1 shows some of the information stored in these

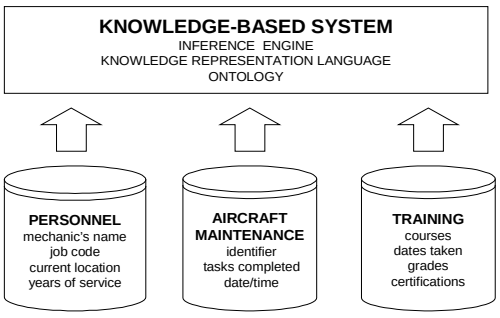

Figure 1: Knowledge-Based Approach to Information Integration

systems. Our experience shows that traditional systems holding this type of information exist as separate data stores with no common data model. This partition of information makes it difficult to retrieve information spread across multiple database systems. The mechanic should be able to make a request such as: *find all local mechanics (and their phone numbers) that have performed a specified task in the last 12 months and that have a level 3 certification (able to performed unsupervised) for this or related tasks.* The context of this work was to integrate these data sources, so that a single query could be submitted by a mechanic without his being encumbered by knowledge of the data structures and query mechanisms of the databases. We also wanted to explore how some of the "field" data, once integrated, could be used to aid decision makers such as planners, schedulers, training personnel, and reliability engineers.

This problem can be conceptualized as a federation of heterogeneous and autonomous database systems [SHET90], which do not share a common global data schema [LITW90]. We decided to investigate the feasibility of using knowledge-based techniques to address this problem in order to achieve the following objectives:

- semantic integration of data from the various databases without the use of a global data schema
- use of the knowledge representation language as a multi-database query language.

We found that the knowledge based approach not only satisfied these objectives, but also provided several ancillary benefits:

(1) the knowledge representation language provides the capability to do advanced inference

(2) the ontology provides a rich, predefined vocabulary that serves as a stable conceptual interface to the databases and is independent of the database schemas

(3) any domain specific knowledge and inference mechanisms become part of the knowledge base and are available in the evaluation of subsequent queries

(4) the inference chains provide a natural means of justification for any computed answers, a feature that is not readily available using traditional coding

(5) advanced applications such as planning can be supported more naturally.

As a tool for this study, we selected Cyc [LENAT95], a commercial high-performance knowledge-based system, as a baseline for evaluating the use of knowledge-based techniques for the semantic integration of heterogeneous information sources. Specifically, we wanted to determine how valuable the supplied ontologies would be in reducing the knowledge engineering workload, to determine if the language would be expressive enough for typical queries in our domain, and to verify that the inference capabilities of the knowledge base would extend the range of queries that could be addressed. This paper is a preliminary report of the study conducted at Boeing. Performance and scalability issues have not been addressed in this study, but are planned for the future.

We explain three salient aspects of our approach. Firstly, we describe some of the issues that arise in mapping from a conventional data modeling language into a domain representation based on ontologies. This mapping is required, since translation between the database specific data model and the common ontology is a fundamental aspect of our approach. Secondly, we describe how queries in our domain can be expressed directly in terms of the knowledge representation language. This property of knowledge-based systems simplifies the generation of queries and permits the knowledge to be accessed from multiple points of view using a single query language. Finally, we describe the advantages of using a knowledge based front-end in decision support applications.

2. Approach

To provide integrated access to multiple distributed heterogeneous information sources, it is necessary to define a common data model where the domain concepts can be expressed at a conceptual level. Each of the individual information sources can then be mapped to this common reference model. We use a knowledge-based approach to arrive at this common data model because of the advantages garnered from the following characteristics:

- In support of reuse, the ontology provides several levels of abstraction. The upper ontology defines general terms that are common to all domains and applications; the middle level ontology provides terms that are specific to the target domain, while independent of a particular application, and the lower ontology provides application specific terminology.
- Problem-solving using the ontology is supported by a high performance inference engine. The ontology defines important relationships among model entities and the inference engine is capable of deducing the consequences of these relationships and answering queries within the competence of the ontology without the burden of writing additional code.
- The language used as an infrastructure for the ontology is expressive enough to address the complex queries typical of decision support applications; furthermore, the ontology is extensible so that queries that are not anticipated during requirements definition can easily be addressed.
- The knowledge represented by the ontology is sufficiently comprehensive to support translation of all the relevant information sources into its common frame

of reference; creation of the domain model is governed by epistemological principles rather than the idiosyncrasies of any particular implementation.

- The ontology supports consistency management and the recognition of inconsistent data.

2.1. Data Mapping Issues

The integration of heterogeneous information requires two primary functions. Firstly, the meanings of the terms must be expressed in terms of some common frame of reference, so that syntactically different references to the same concept can be identified. For example, the same concept may be named differently in two different databases or the units of time and quantity may differ. Secondly, important information regarding the relationship among the entities in two different databases may not be explicitly stated in either database. For example, if the mechanic skills database states explicitly that a mechanic is capable of working on electromechanical components and the maintenance history database records repairs for servo units, then it is necessary to have the additional, common-sense knowledge that servo units are a kind of electromechanical device in order to draw the inference that this mechanic might be appropriate for a servo unit repair.

A limitation of traditional data models is that they fail to capture much of the semantics of the objects and relations in the data model. They specify what relationships among the entities may exist, but they do not support reasoning about what additional conclusions may be warranted by the facts currently present in the data. Approaches for determining the structural validity of entity-relationship models that can easily be incorporated into the database modeling and design process have also recently become available [DULL98]. Those modeling languages that do provide support for writing down semantics, such as the Object Constraint Language (OCL) for the Unified Modeling Language (UML) use the semantics almost exclusively for constraint checking, and typically do not support inference. Deductive databases [CERI89] attempt to supply an inference capability by adding rules to databases. However, although deductive databases are both efficient and have guaranteed termination, the languages supporting them typically have less expressive power than CycL, and they do not come equipped with the large amount of built-in common sense knowledge that is available with the Cyc ontology. What is lacking is a substrate of common-sense knowledge that both supports the mapping of concepts into a common frame of reference and also supplies the missing information to support the linking of information that goes beyond what is explicitly stated in the database.

2.2. Query Formulation Using the Knowledge Representation

The characteristics of the query language that were most important in our application are as follows:

- The knowledge representation should be highly expressive. This is desired because we wish to provide support for complex queries.
- The conceptual vocabulary should be readily extensible. The user should be able to write new concepts in terms of already existing concepts and to specify new relationships and axioms among existing concepts. This property is required to facilitate reuse of the knowledge representation and to permit extending the upper level ontologies by adding application specific lower level ontological concepts.

These objectives are achieved by specifying the knowledge representation in two layers, the first layer being a highly expressive formal language and the second layer, built from primitives supplied by the first layer, providing a set of terms for describing domain conceptualizations. The first layer of the knowledge representation used in our experiments is a formal language, CycL, whose syntax derives from first-order predicate calculus, with extensions to handle some basic functionality not easily expressible in FOPC, such as quantitative, spatial and temporal reasoning. This language provides the basic building blocks for constructing conceptualizations. The second layer of the knowledge representation uses the first layer to build the ontologies, which provide a set of terms for describing the domain. The upper level ontology provides the infrastructure for reasoning about time and dates, spatial relations, quantities, mathematics, action descriptions, part/whole decompositions, the composition of substances, information, etc. The elements of the middle level ontology that were most useful in our application provide descriptions of materials, devices, financial concepts, transportation concepts, human activities, etc.

We conclude this section with the formal representation of the example query from the introduction in order to illustrate the general ideas described above. The problem example stated in the introduction is to determine the phone number of any aircraft mechanic working at Dover AFB that is certified at level 3 to perform repairs on an F-18 right servo valve and drive unit repair, and who was also a member of a maintenance team that has performed that task within the last twelve months This query is expressed formally as:

```
(and
    (isa ?PERSON AircraftMechanic)
    (pointOfContactInfo ?PERSON Workplace addressText-NC "Dover AFB")
    (certificationLevel ?PERSON F-18-ServoValveAndDriveUnit 3)
    (recentlyCompleted (RepairingFn F-18-RightServoValveAndDriveUnit) ?PERSON)
    (pointOfContactInfo ?PERSON Workplace phoneNumberText ?PHNUM)
```

The name of the mechanic, his currently assigned location, and telephone would typically be located in a personnel database. The information pertinent to the previous work performed by that mechanic would be in a maintenance history database, and the qualifications of the mechanic would typically be found in a training database. In solving the above query, the inference engine will route the queries to the appropriate databases and combine the results to answer the specific question posed by the user.

The task of repairing an F-18 right servo valve and drive unit is described in an activity ontology, in which various actors play different roles. For example the right servo valve and drive unit is the object which is acted upon and the task is performed by a particular maintenance team, of which the retrieved individual is a member. The repair event is a temporal object and hence has a starting and ending time. This example also illustrates the following notions:

- Temporal Knowledge—the task is constrained to be completed within the last 12 months
- Part/Whole Relationships – the maintenance team which performed the task is composed of a set of mechanics
- Instantiation of Objects Belonging to General Class Descriptions – the person retrieved is an instance of aircraft mechanic; the task described is an instance of repairing an F-18 right servo valve and drive unit
- Use of Non-binary Predicates—certificationLevel (ternary) and pointOfContactInfo (quaternary).

Axioms provide an important mechanism for expressing relationships between concepts. For example, the following axiom:

```
(implies
    (and (performedBy ?TASK ?TEAM)
         (hasMember ?TEAM ?PERSON))
    (performedInPartBy ?TASK ?PERSON))
```

implements the common-sense notion that, if a task is performed by a team of individuals, then every person belonging to that team has performed some aspect of that task. This common-sense axiom is involved during processing of the example query in making an important connection between the data described in the maintenance log, which describes team composition and task assignment by team, and the user intent embodied in one of the constraints of the query – finding an individual who has recentlly participated in performing the task.

The example query can easily be generalized to incorporate other notions of skill matching. For example, rather than requiring that the selected individual has previously performed the specified task, we could require only that the previously performed task be similar to the required task. Various realizations of similarity could be encoded using similarity predicates. For example, a similarity predicate could state that working on a symmetric part (e.g., right versus left servo valve) is sufficient to accept the individual for the task. Even more generally, a skills decomposition hierarchy could be used to relate general skills (e.g., dexterity, familiarity with a particular subsytem) to the capabilities required to perform specific tasks, so that an individual with the requisite skills (acquired either through training or demonstrated by having performed a task that requires those skills) could be selected for the task.

The advantages of using axioms to express the common-sense knowledge about concept relationships include:

- The knowledge can be expressed at the highest level of abstraction at which it is applicable; for example, the teaming axiom above applies to all members of any organization performing jointly on any task
- Much of the common-sense knowledge can be pre-packaged into carefully engineered microtheories, each of which describes the competence required in a particular domain; this facilitates the creation of off-the-shelf knowledge components and reduces the manual entry of knowledge. Knowledge engineering becomes in large part a matter of selecting the microtheories with the appropriate competence as opposed to manual entry of individual rules
- In the case of "missing knowledge", the ontology provides an infrastructure of terms and

relationships that facilitates the manual entry of any additional knowledge required; this new knowledge enhances not only the applications for which it was explicitly developed, but also improves the performance of any other applications which make use of the microtheory to which the new knowledge was added.

2.3. Knowledge-Based Front End for Decision Support

The approach described here can also be viewed as a knowledge-based front end for applications such as decision support. The knowledge-based approach we used has several desirable features for decision support applications:

- Knowledge has *declarative semantics* and is interpretable by a domain independent inference engine. Thus coding is reduced to knowledge specification. Knowledge manipulation and interpretation are achieved through specifying and proving problem goals.
- Knowledge is *active*. The ontological approach supports the use of forward chaining rules; therefore the current context or state of knowledge can trigger forward chaining rules which generate additional knowledge.
- Knowledge can be used to *answer questions and achieve goals*. The ontological approach also supports backward chaining and means-ends analysis, thus enabling the inference process to construct dynamic plans for answering queries.

3. Related Applications

The Carnot project developed semantic modeling techniques that enabled the integration of static information resources and pioneered the use of agents to provide interoperation among autonomous systems [WOEL96]. The Infosleuth Project extends the capabilities of the Carnot technologies to dynamically changing environments, where the identities of the resources may be unknown at the time the application is developed [BAYA97]. The Information Manifold [LEVY96] is an implemented system that provides uniform access to a heterogeneous collection of more than 100 information sources, many of them on the WWW. The contents of an information source and its query capabilities are described declaratively. The source descriptions are used to create query plans that can access several information sources to answer a query. HERMES [SUBR00] provides a general, declarative language for defining a mediator, which expresses semantic integration of information from diverse data sources and reasoning systems. SIMS [AREN96] also uses a domain model for describing the available information sources. Queries to SIMS are posed using terms from the domain model, and reformulation operators are used to dynamically select an appropriate set of information sources and to determine how to integrate the available information to satisfy a query. Our work differs from those above in that we are primarily interested in evaluating the suitability of an existing commercial knowledge-based system, with its extensive substrate of common-sense knowledge, to facilitate the rapid knowledge formation that is required of large-scale information-integration applications.

4. Conclusion

Our current research has demonstrated that it is relatively straightforward to answer queries requiring information from heterogeneous databases using ontologies as a common data model. The primary reasons for this are two-fold. Firstly, we found that the concepts already defined in the commercial upper ontology provided a rich vocabulary about human activities, space and time. This meant that we only had to define the additional concepts needed for our application and link them to the concepts in the upper ontology. Secondly, in our application domain, aviation logistics maintenance, the information describing the important domain concepts--task descriptions, skill descriptions, parts breakdowns, parts and materials requirements, maintenance histories, etc.--is well documented and relatively easy to transform into ontological terms. The primary difficulty was not unavailability of information, but rather that it is poorly integrated. The consequence of these two observations is that a relatively small knowledge engineering effort is required to apply the approach to a full-scale implementation.

The expressive power of Cyc facilitates both reuse of existing knowledge and the integration of legacy knowledge representations (e.g, SQL, UML, conceptual graphs). The concept hierarchy is the most extensively developed aspect of Cyc; the existing human activities ontology provided almost all of the conceptual vocabulary required to describe aviation maintenance events. Extensive interconnection of concepts is provided though the axioms, which describe important relationships between the concepts. Most of the existing axioms are focused on common-sense reasoning, such as the fact that a member of a team that has performed a task has also partially performed that task, or that access to a closed container can be obtained by opening the door of that container. The common-sense axioms often supply surprisingly perspicacious interconnections among high level concepts. However, it is also necessary to supply domain level axioms in order to make the problem-solving description complete. We found that the vocabulary available at the upper level does facilitate developing the domain dependent knowledge at the lower level.

The advantages of the expressiveness available in Cyc do come at a small price. Some effort is required to familiarize oneself with the vocabulary of the upper level ontology. The tools Cyc provides to do this are well designed; however, it is our belief that technically skilled knowledge engineers are still required at this time to perform this task effectively. Work is currently underway in the DARPA Rapid Knowledge Formation project to address direct knowledge entry by domain specialists who do not possess knowledge representation skills. Another well-known problem we rediscovered with Cyc is the tradeoff between expressiveness and computational efficiency. Although Cyc has several internal mechanisms for improving efficiency, some fine tuning of inference is still required for a particular application. One way to accomplish this is to use Cyc to integrate special purpose knowledge representations, which are known to be efficient on specific sub-problems in the domain. Thus, Cyc can be used not only to integrate databases, but is also well suited for integrating legacy knowledge representations. We do not envision that Cyc will become a universal knowledge representation, but rather that it is a useful tool for integrating existing special-purpose knowledge representations.

We close with a few comments on the conditions under which, the approach described in this paper is most appropriate:

(1) support for a broad range of queries, the form of which may not be known in advance, is required

(2) the application must support the use of open information environments in which new information sources may appear and old ones become unavailable, so that flexible inference is a paramount concern

(3) technically skilled personnel are available to perform translations between the Cyc ontology and existing legacy knowledge representations and to develop the lower level ontology

(4) the number of subsystems to be integrated is moderately large, so that the conventional approach of pair-wise integration is not feasible

Additional examples of domains with which we are familiar that have these characteristics are:

- cockpit situation awareness and decision aiding for free flight
- integration of air traffic management and flight information services
- integrated design environments
- modeling and control of market-responsive, complex manufacturing systems.

5. References

[AREN96] Arens, Y., C. Knoblock, and W. Shen. "Query Reformulation for Dynamic Information Integration". *Journal of Intelligent Information Systems*, 1996.

[BAYA97] Bayardo Jr, R. J., W. Bohrer, R. Brice, A. Cichocki, J. Fowler, A. Helal, V. Kashyap, T. Ksiezyk, G. Martin, M. Nodine, M. Rashid, M. Rusinkiewicz, R. Shea, C. Unnikrishnan, A. Unruh, and D. Woelk, "Infosleuth: Agent-Based Semantic Integration of Information in Open and Dynamic Environments, *SIGMOD '97*, p. 195-205.

[CERI89] Ceri, S., G. Gottlob, L. Tanca, "What You Always Wanted to Know About Datalog (And Never Dared to Ask)", *IEEE Transactions on Knowledge and Data Engineering*, 1 (1), March 1989, pp. 146-166.

[DULL98] Dullea, J. and I. Song, "An Analysis of Structural Validity in Unary and Binary Relationships in Entity Relationship Modeling*", Proceedings of the Fourth International Conference on Computer Science and Informatics*, Volume 3, pp. 329-334, Joint Conference on Information Sciences, Research Triangle Park, NC, October 23-28, 1998.

[LENAT95] Lenat, D. B., "CYC: A Large-Scale Investment in Knowledge Infrastructure,*" Communications of the ACM* 38 (11), November 1995, pp. 33-8

[LEVY96] Levy A. Y., A. Rajaraman, and J. J. Ordille, "Querying Heterogeneous Information Sources Using Source Descriptions," *Proceedings of the 22nd International Conference on Very Large Databases, VLDB-96*, Bombay, India, September, 1996.

[LITW90] Litwin, W., L. Mark, and N. Roussopoulos, "Interoperability of Multiple Autonomous Databases," *ACM Computing Surveys*, 22 (3), September 1990, pp. 267-293.

[SHET90] Sheth, A.P. and J. A. Larson, "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases," *ACM Computing Surveys*, 22 (3), September 1990, pp. 183-236.

[SUBR00] Subrahmanian, V. S., S. Adali, A. Brink, R. Emery, J. J. Lu, A. Rajput, T. J. Rogers, R. Ross, C. Ward, "HERMES: A Heterogeneous Reasoning and Mediator System (submitted for publication).

[WOEL96] Woelk D., P. Cannata, M. Huhns, N. Jacobs, T. Ksiezyk, R. Lavender, G. Merdith, K. Ong, W. Shen, M. Singh, and C. Tomlinson, "Carnot Prototype", in Object-Oriented Multidatabase Sysetms, O. Bukhres and A. Elmagarmid (editors), 1996.