Problems in Heterogeneous Knowledge Source Integration: TraumaGEN and the Metathesaurus

Terrence G. Harvey
Department of Computer Science
University of Delaware

September 21, 1998

Abstract

This paper reports on some of the difficulties inherent in the integration of information from heterogeneous knowledge sources (KSs). It reviews the approaches and limitations of some recent systems, and presents example problems from an attempt to integrate two KSs in the medical terminology domain.

1 The Metathesaurus

The National Library of Medicine has developed the Unified Medical Language System (UMLS) to aid medical system developers. One intended application is to help map a user's medical terms to those in the controlled vocabularies that are part of the UMLS. The UMLS consists of the Metathesaurus, the Semantic Network, the Specialist Lexicon, and the Information Sources Map.

The Metathesaurus has 330,000 different medical concepts, with a Concept Unique Identifier (CUI) for each one, even if two concepts have the same name. For example, the string "cold" is an ambiguous string connected to two concepts: Cold<1>, a disease, and Cold<2>, a temperature.

The information is an aggregation of over 30 medical domain vocabularies. When a new concept is added to the Metathesaurus, new relationships may be determined between concepts and terms from different source vocabularies. Thus the information available from the Metathesaurus is a superset of the knowledge from all of its sources.

The value of Metathesaurus is threefold: uniform single query access to a large number of heterogeneous knowledge sources; a single concept identifier that is appropriately matched in each source; and the recording of relations that are only available across sources.

The Specialist Lexicon contains medical vocabulary, and also English words commonly used in medical citation records and dictionaries. It is designed to meet the lexical information needs of the SPECIALIST Natural Language Processing System, and so includes syntactic, morphological, and graphemic information.

2 TraumAID, TraumaTIQ, and TraumaGEN

TraumAID (Webber et al., 1992) is a decision support system for the initial management of multiple trauma. TraumaTIQ (Gertner and Webber, 1996)is a module that compares a physician's inferred management plan with TraumAID's plan, and creates critiquing information for each significant difference between them. The set of critique information is then processed by a text planner, TraumaGEN (Carberry and Harvey, 1997) that generates coherent, integrated messages.

One limitation on TraumaGEN's ability to integrate messages properly is that it does not have semantic knowledge about the relations between medical concepts. This occasionally results in output that could be made more concise or natural by taking advantage of the relations between terms used in a sentence or paragraph.

For example, one message now contains references to both a "possible intra-abdominal injury" and a "possible GI tract injury". With a well-designed medical term hierarchy, it would be possible for a reasoner to recognize that "possible GI-tract injury" is a concept that is subsumed by the concept "possible intra-abdominal injury". This subsumption would then allow TraumaGEN to make the message more concise and coherent. TraumaGEN currently does some matching of concepts via a hash table. However, the entries to the table must be made by hand, and can only be made after a potential improvement is noticed by a human reader.

If the hash table entries could be made automatically, then two benefits would be derived. First, as new medical terms and procedures are introduced to the system, they would not require hand comparison against existing terms for potential subsumption. Second, by identifying potential subsumption and making an appropriate hash table entry, the system would prevent the generation of sub-optimal output. This would eliminate reliance on user identification and reporting of possible improvements.

Another area in which TraumaGEN has room for improvement is the use of local medical vocabulary. Different regions in the U.S. sometimes use different terminology for the same concept. An example of one such concept is a "pericardial tamponade", which is the East coast name for a concept called "cardiac tamponade" on the West coast. The existence of a medical "thesaurus" could allow TraumaGEN to customize its output for different locations. And if foreign language term equivalents were available, they would facilitate TraumAID's adaptation for use outside the U.S.

All of this information is available from the Metathesaurus. The task of using the Metathesaurus to augment TraumaGEN's functionality is essentially one of integrating two KSs. TraumaGEN has its own KB, with vocabulary, structure, and accessor functions, albeit much simpler than those of the Metathesaurus.

Data in the Metathesaurus will only be useful to the extent that it relates to TraumaGEN's KB. The task becomes one of determining the relationships which hold between data in the two KBs.

For this application, some integration has already been performed by the Metathesaurus. Foremost is the uniform query interface, which allows the user to query any or all of the sources using a single syntax.

Two types of additional integration are needed to make the Metathesaurus useful to TraumaGEN. First, while the query interface is uniform, the responses to the query are not. The responses will require a two level conceptual breakdown: separation by source, and interpretation by a translator which knows the source's output format.

This kind of mixed output could be avoided by separately querying the Metathesaurus for the information from each source, but this increases the amount of work for both sides of the system. One value of the Metathesaurus to TraumaGEN lies in the broad range of data which are available from one location. While we must write a translator for responses, we do not have to do so for queries.

Second, the terminology used by TraumaGEN must be mapped to the concepts of the Metathesaurus. It possible to do this task by hand, especially since TraumAID's domain is currently limited to a small subset of its intended domain. However, the large size of the eventual medical vocabulary, and the likelihood of small, regular additions or changes to its knowledge base, argue for an automated procedure. Automation will also potentially allow TraumaGEN to benefit from improvements in the Metathesaurus, which otherwise would require human monitoring and changes.

3 Integrating multiple databases

A large number of heterogeneous information sources are available to computer users, and the integration of these sources is required to make the information useful and manageable. Multiple databases within corporations, i.e. sales, R&D, production, and accounting systems, each designed to meet disparate needs, must share information efficiently. Merging corporations have even greater need for automated integration. WWW resources represent a vast, mostly free data supply with limited processing capabilities (and hence limited value).

The goal of automated integration is to free the user from having to locate relevant sources, query sources in isolation, and combine the results in a coherent way. Furthermore, an integration system can derive new information by examining diverse sources or verifying the contents of one database against another.

The need for this kind of information processing is demonstrated by the popularity of a simple implementation. BargainFinder (Krulwich, 1995) is a simple integration agent designed to operate on the Internet. As an Internet tool, it represents a substantial improvement over keyword search, since it can query databases instead of just searching web page text. A user can, for example, fill out forms to request CD information from databases, and the system performs very low level data integration:

it places all CD prices and the associated URLs on one page. The popularity of this tool (it had over 200,000 hits in nine months) proves the great demand for integration, even at a low level.

3.1 Intelligent agent approaches

Agent systems are well-suited for the task of heterogeneous KS integration (Knoblock et al., 1994; Decker et al., 1995). A major advantage of these systems is modularity. No person, software, or hardware needs to maintain a global picture of the system. Agent action and interaction are determined by decision making and communication protocols resident in each agent, not in some giant overseer. The agents can be developed and maintained separately, even by different organizations. Modularity can also increase resilience and help to make systems fail-soft.

Agents are entities which can choose their own path to achieve a goal (and possibly even their own goals). They may maintain a representation of their knowledge, communicate with other agents, construct plans to achieve goals, and learn rules and information which will help them perform their task.

The benefits of modularity encourage the use of different classes of agents. Many KS integration systems use two classes: translator agents which are each responsible for communicating between the system and one KS; and mediator agents, which serve as intermediaries between the user and the system, or between other agents.

Well-designed agent systems are also easy to update and maintain. To add a new source to a modular integration system, simply create a new translator between the KS and the virtual language of the agents (in an ideal world, that is; real world KS integration may involve modifications to mediators as well).

Distributed agent systems have also been developed to encourage independent agent specialization, so that over time agents can become "expert" in a particular domain (Decker et al., 1995; Knoblock et al., 1994). These systems also allow agents to become knowledge sources, so that agents can build upon the work of other agents. This also creates hierarchies which can form and reform dynamically.

A further benefit of distributed agent systems is the ability to modify the number of agents, or the ratio of source agents to mediators. This allows the system to maintain an efficient balance when work loads change (Decker et al., 1997).

4 Other work

Much work has been done in the area of integrating heterogeneous knowledge bases. Two systems, TSIMMIS (Chawathe et al., 1994; Hammer et al., 1997) and HERMES (Adali and Subrahmanian, 1994), are not integration systems, but rather tools for designing integration systems. These tools automate the creation of agents, which will in turn integrate the information sources.

4.1 TSIMMIS

The TSIMMIS architecture consists primarily of translators and mediators. It is the task of the translator to convert information between the input/output formats of a knowledge source and a standard representation. Given a source's query and output formats, TSIMMIS can generate such a translator. The common representation is OEM, or Object Exchange Model (Papakonstantinou et al., 1995), which consists of self-describing objects that allow simple nesting. The OEM representation is then passed to the mediator.

TSIMMIS mediators are relatively simple. They direct queries to appropriate sources, and may eventually eliminate duplicate information or convert time stamps to a common format, though these abilities do not appear to be currently implemented. All communication to or between mediators is in OEM.

4.2 HERMES

HERMES (Adali and Subrahmanian, 1994) is another system development tool. Like TSIMMIS, it helps users develop translators and mediators, but it appears to be a more substantial design tool, especially for mediators.

HERMES separates the integration process into two parts, domain and semantic integration. Domain integration consists of writing a translator for a new source. HERMES guides the user through the creation of a catalog file for each domain (essentially a "wrapper" for the new source), which makes it usable by HERMES. The file contains information on opening the source, a list of functions to be accessed by the mediator, and specifications of input and output format for each function. The construction of the catalog enables HERMES to accept as "sources" not only traditional data sources, but programs such as reasoners or search engines.

The semantic integration, or mediation, covers "pooling" information from distinct sources, and detection and resolution of conflict between sources. "Pooling" in HERMES is more than simple aggregation; it refers to the potential use of data from multiple sources, along with inference mechanisms, to create "new" information that could not be inferred from the data of any single source. A mediator is designed to identify complementary data from two or more sources, pass the information to a reasoner or processor, and report or store conclusions. Conflicts between data will be discussed later in this paper.

"Learning" in HERMES refers only to the ability to save new strategies written by the mediator author/user, when told, for incorporation into the strategy menu.

4.3 The Information Manifold

The Information Manifold (IM) (Kirk et al., 1995) is a system for browsing and querying multiple networked information sources. The authors focus on minimizing the number of sources queried; their algorithm guarantees the relevance of each source. The system has to have an accurate global domain model as well as information about the relative completeness of each KS. Establishing relative completeness of many classes of data in multiple sources, and maintaining the accuracy of relative completeness data could be difficult.

The integration algorithm shows a preference for the most complete source of data, apparently without regard to costs. In an example in the paper, IM ignores one source whose information is a proper subset of a larger source. Unmodified, this algorithm will prevent the IM from gathering new information about the contents of the small source. It also neglects the importance of maintaining a variety of sources for accuracy checking and as back-up resources.

To work, the algorithm requires an accurate, detailed description of every information source including a semantic representation of all relations in the KS and a detailed representation of the KS internal structure.

4.4 SIMS

In the SIMS (Knoblock et al., 1994) architecture, each agent carries a detailed model of its own expertise, and models of the knowledge sources available to it (including other agents). SIMS agents communicate using two languages: KQML (Knowledge Query and Manipulation Language) (Finin et al., 1994) is the meta-language used to carry information about agent desires and intentions, while LOOM (Brill, 1991) semantic nets are used to represent the information "goods" passed between agents.

Four steps comprise the SIMS algorithm. First, an agent selects appropriate sources for a query based on content and cost. Next, it creates a plan to retrieve and possibly process the data. Then it refines the plan based on its knowledge about the KS, and finally it executes the plan.

Two kinds of agent "learning" take place in SIMS. The first is simply caching: SIMS caches data that is frequently used or that is difficult or expensive to obtain. Cached data is stored in LOOM, and is indexed with a description of the information gleaned from the initial query.

The second kind of learning is Semantic Query Optimization (SQO). When the system detects an expensive query it attempts to learn (using ID3 (Quinlan, 1986)) information that will reduce the cost of similar future requests. This learning assumes the presence of a detailed representation of the information in the KS, including: a fine-grained semantic description and an associated cost for each type of data/function available (cost information could be learned and recorded by the agent); domain knowledge about the relations between data; and a reasoning system.

As an example, (Knoblock and Ambite, 1997) present a query that retrieves ship types whose range is greater than 10,000 miles. Given the presence of domain knowledge that ships with this range have a draft over twelve feet, and cost knowledge indicating that getting draft information is cheaper than range information, SQO converts the query to one requesting draft information that will return the same information as the original query as long as the cost and domain knowledge remain true.

4.5 MACRON

MACRON (Decker et al., 1995), an intelligent agent approach, is not designed as an automater of KS integration, nor as a design tool, but to answer specific user queries. Nevertheless, it faces many of the same issues as the other systems discussed here. Working with heterogeneous KSs, it is designed to take advantage of the distributed nature of finding information on the Internet. In its algorithm, information is aggressively acquired and organized in anticipation of future need, not just retrieved upon request. Multiple agents dynamically develop expertise in different areas as a result of careful acquisition of information. The system uses five performance criteria: speed, completeness, certainty, resource use, and cost, to evaluate the knowledge sources available. Its multiple semi-autonomous agents cooperatively communicate information and plans with one another to exploit the work being done by other agents.

The system is also intended to provide coherent, integrated data, but hasn't specified the mechanism for this yet. The system is in the early stages of implementation.

5 Translator complexity

The many kinds of computational complexity involved in database queries are also a major obstacle to efficient translators, so much of the work in making database queries more efficient will be applicable. However, certain problems arise in heterogeneous knowledge base integration that do not appear in querying single sources.

It is the task of the translator to convert a mediator query into an equivalent query in the language of the source. Suppose mediator M has a user request $C(x)$ such that $C(x) = f(x) \cup g(x) \cap h(x)$. And translator T has operators F, G, and H that return $f(x)$, $g(x)$, and $h(x)$, respectively. The problem facing T is to determine whether it can satisfy M's request for C-type information, and if so, how? This is a major issue for KS integration, because determining equivalence is a combinatorial problem. And while T, in this example, may easily provide this information, the request may have also gone to other translators that are fruitlessly combing combinations of their primitive operators to see if they can match the request.

Translation also depends on the existence of domain information, such as $seaside\_resort(x) = geo\_loc(x) \wedge resort\_loc(x) \wedge coastal\_loc(x)$, which to date must largely be entered by human hands. Information of this type can be discovered from data by noting that $\{x|resort\_loc(x)\} \subseteq \{x|geo\_loc(x)\}$; but rule discovery of this kind is expensive, and must be re-evaluated each time the data set is modified. The learning algorithm for SQO in SIMS attempts to identify robust knowledge that is less subject to alterations or corrections, and thus more profitable for use in rule derivation, since maintenance costs will be low (Hsu and Knoblock, 1996). Intuitively, data that rarely or never change (such as dates in a history database) may be considered more likely to be part of some fundamental characteristic of a domain.

A reasoner can also work out from hierarchy or logical information that $geo\_loc(x) \wedge resort\_loc(x) = resort\_loc(x)$ but again, this computation is expensive, and could be prohibitively so depending on the amount of domain knowledge present.

Attempts are being made to reduce the negative impacts of query re-mappings. "Planning by rewriting" (Knoblock and Ambite, 1997) generates low-cost but possibly sub-optimal plans. Another effort, "recursive information gathering" (Duschka and Levy, 1997) generates a maximally contained set of query rewritings (the set contains information subsets of the query, not just equivalents). By exploiting functional dependencies in the domain model, queries can be translated in time polynomial in the number of query terms. This requires the presence and prior specification of functional dependencies in the source.

6 Mediator complexity

A primary task for mediators is planning. Mediators plan to meet goals of information, cost, accuracy, and speed in a dynamic environment. Planning which translators (or other mediators) to interrogate for each request is an important step in achieving the goals. The Information Manifold uses extensive knowledge of its sources to ensure that only relevant sources are polled in its plan. It can also guarantee that a minimal number of sources were polled. Occam (Kwok and Weld, 1996) claims to be a sound, complete, and efficient planner, provided that every piece of information knowable about each KS is carefully written into the meta-language. This will result in a meta-language that is "meta" in name only: to work as stated, the language will need to be the union of every KS language below it. Both IM and Occam are useful for clarifying the boundaries of mediator complexity, but their requirements for extremely detailed KS representation will render them unwieldy for life-sized domains.

Perhaps the most difficult aspect of integration mediation is the handling of inconsistent data. The problem is two-fold: identification of conflict, and resolution of the conflict. Most systems have no conflict strategy or a single simple strategy. Such strategies allow systems to function in a complex world, but also greatly reduce their potential as integrators.

One such strategy is to accept information if and only if no source disagreed with the information. Thus one clerical error or out-of-date datum could prevent the acceptance of data that is identically reported by several other sources. Data reported from a single source would be assumed accurate (since even the best agent can't do anything about false information unless there is some other information that contradicts it). One circumstances not addressed by these approaches occurs when a query of an up-to-date resource has no information (possibly because it has detected uncertainty about the information), but an older source responds. How can a system determine when a lack of information should be considered inconsistent with the presence of information? Any default answer to these questions will certainly be wrong in some situations.

An alternative is to have KSs "vote", and the information reported by the highest number of sources wins. Numeric information can be averaged, or reported as a function of each datum and some relative confidence measure. Again, these solutions are hardly robust, and indicate the need for mediators to have a broad spectrum of resolution strategies.

The HERMES system, which facilitates the design of mediators, has a conflict resolution tool kit. The kit is an interface with the mediator designer which facilitates resolution of the conflicts which arise during semantic integration. It offers a menu of resolution strategies, which assume a set of integrity constraints in the form of semantic relations between data sources. Selection of a strategy causes the system to generate a set of rules which become part of the mediator, or the author can write a new strategy.

The value of the HERMES tool kit lies mostly in its recognition of the need for a broad spectrum of conflict resolution strategies. The ability to add strategies to the tool kit in response to new classes of conflicts allows mediator authors to code certain problems for human intervention. For example, a conflict detection during integration could generate an email to a human administrator, who could choose to arbitrate the conflict instance (and respond by email) or develop and implement a new strategy.

The Metathesaurus assists mediation by directing source queries, and by providing an Information Sources Map to help mediators choose a source. However, the Metathesaurus has some default preferences used when ordering the answers to some queries, and such defaults are important for the mediator designer to keep in mind.

7 Translator design

The design of translators is also difficult to automate. An early piece of a translator I wrote for the Metathesaurus requested a list of concepts which were identified, in some source's tree structures, as children of a concept C. The code processed the answer by scanning for CUIs and eliminating references to C. This worked for most concepts, but reported extraneous CUIs for some. It turned out that one of the KSs responded to a request for children by returning both the parents and the children of the concept. This is not likely to be an oversight on the part of the Metathesaurus designers; rather it is a means of bypassing the complexity of trying to design a uniform representation which is general enough to incorporate all the information of current and future sources, yet specific enough to be useful. This translator's problem was easily fixed, but it points out that reliable automatic design of translators is a long way off. For now, human examination and re-examination of the data format is critical.

A further problem is changes in the data format of a source over time. While translators can be designed to recover from small changes in spacing or order, adaptation to many changes will require human oversight. An example: during the months I accessed the Metathesaurus, the response to a simple concept name request changed from returning just the name and unique identifier to returning both of these and all synonyms, including foreign language terms. Further, the function which formerly found synonyms was inactivated.

8 Integrating TraumaGEN and the Metathesaurus

My initial approach was to take medical term names from TraumaGEN output and find them in the Metathesaurus. Then I planned to use the relationship information from the Metathesaurus to build semantic trees around the terms. These trees would then allow me to aggregate some terms under the name of a common parent, or subsume one term within another.

Finding term matches turned out to be complex. First, consider the medical term "pericardial tamponade". A concept search for this term returns a CUI and the string "cardiac tamponade", with "pericardial tamponade" listed as a synonym. A human reading of the definition of "cardiac tamponade" confirms that it refers to the same medical concept as "pericardial tamponade", so this match query worked simply and correctly.

A second example reveals some difficulties. "Intra-abdominal injury" does not return a match from a concept search. Now, we can envision an algorithm that might proceed as a human would, checking the components of the term. "Injury" does return a match, but "intra-abdominal" does not. Further tries result in a Specialist Lexicon entry matching "intra", a prefix, and a Metathesaurus entry "abdominal". Left-trimming the original we try "abdominal injury" with no match.

However, the Metathesaurus allows concept searches using approximate matching. Again, we can easily imagine an algorithm which gets a list of possible matches from an approximation algorithm, and then somehow picks a "best match". This is aided by the Metathesaurus's own attempt to rank the matches, which is not solely string based. The approximate match request for "intra-abdominal injury" returns 22 possibilities, and the top concept is "abdominal injuries". This class of injuries in the Metathesaurus actually contains many kinds of intra-abdominal injuries, but no subclass that would match the "intra" prefix exactly.

```
>ks -meta -c ''lacerated diaphragm''
Query Term:  lacerated diaphragm
Term not found.

>ks -meta -c -am ''lacerated diaphragm''
Vaginal Diaphragm
Vaginal Contraceptives
Diaphragm
Laceration, NOS
Contraceptive Agents
Muscles
Vaginal
Psyche, NOS
Psychic
Chin
Poisoning by vaginal contraceptive, NOS
Insertion of diaphragm into vagina
Removal of diaphragm from vagina
Emko Vaginal Contraceptive Foam
Encare Vaginal Contraceptive Suppositories
Semicid Vaginal Contraceptive Inserts
Today Vaginal Contraceptive Sponge
Stimulators, Diaphragmatic/Phrenic Nerve
Biopsy of diaphragm
Congenital absence of diaphragm
```

Figure 1: Two queries to the Metathesaurus: a direct query for a concept identifier, and the same query using the Metathesaurus' approximate matching facility.

Despite the lack of an exact match, it is conceivable that information derived from the trees containing "abdominal injuries" could still provide TraumaGEN with useful information. Perhaps we could use a rule indicating that a term with a prefix was likely a sub-class of that term alone. Then the hierarchy information derived from the trees could be used. Still, the term matching process is starting to look quite involved if it is to proceed to this point.

"Lacerated diaphragm" is another term for which there is no exact match. A request for an approximate match yields 20 possible matches, though the ranking by the Metathesaurus leaves something to be desired, as shown in Figure 1.

"Tenderness" from TraumaGEN returns an exact string match, complete with CUI, from the Metathesaurus. Unfortunately, a quick definition comparison reveals that the two concepts are not the same. TraumaGEN, though it does not explicitly encode this information, in fact intends the meaning "abdominal tenderness" when it refers to "tenderness".

This kind of conflict, a false match, is a very dangerous possibility in KS integration, since it could easily go unnoticed if the process were automated. This false positive could be avoided if more elaborate semantic information were available in TraumaGEN's KB. Augmentation of a KS prior to integration might include definitions, or placing the information in a semantic net. Even a large amount of extra information for each term would require powerful techniques and substantial domain knowledge to provide an accurate mapping. Because of the potential harm that an error in such a setting could cause, human oversight would almost certainly still be required. But more importantly, obtaining that information was the whole intention of the integration, so requiring it at the outset defeats the purpose

```
ks -meta -raw -rel tenderness
CXT|C0234233|S0409125|RCD95|X75r0|1|ANC|1|Read thesaurus||.....|||
CXT|C0234233|S0409125|RCD95|X75r0|1|ANC|2|History and observations||X76sU|||
CXT|C0234233|S0409125|RCD95|X75r0|1|ANC|3|Clinical history and observations||X76sV|||
CXT|C0234233|S0409125|RCD95|X75r0|1|ANC|4|Pain and sensation||Xa05Q|||
CXT|C0234233|S0409125|RCD95|X75r0|1|ANC|5|Pain observations||Xa0yP|||
CXT|C0234233|S0409125|RCD95|X75r0|1|CCP||Tenderness|C0234233|X75r0||+|
CXT|C0234233|S0409125|RCD95|X75r0|1|CHD||Abdominal tenderness||X75rQ||+|
CXT|C0234233|S0409125|RCD95|X75r0|1|CHD||Bone tenderness|C0238807|Xa0vz|||
CXT|C0234233|S0409125|RCD95|X75r0|1|CHD||Facial tenderness|C0239532|Xa0vx|||
CXT|C0234233|S0409125|RCD95|X75r0|1|CHD||Joint tenderness|C0240094|Xa0w0|||
CXT|C0234233|S0409125|RCD95|X75r0|1|CHD||Skin tenderness|C0241164|Xa0w2||+|
...
CXT|C0234233|S0409125|SNM2|F-82610|1|ANC|1|SNOMED-2|C0220966||||
CXT|C0234233|S0409125|SNM2|F-82610|1|ANC|2|Function Axis|||||
CXT|C0234233|S0409125|SNM2|F-82610|1|ANC|3|Function and Abnormal Function of the Nervous and Musculoskeletal Systems|||||
CXT|C0234233|S0409125|SNM2|F-82610|1|ANC|4|Functions and Functional Disorders of the Nervous System|||||
CXT|C0234233|S0409125|SNM2|F-82610|1|ANC|5|Pain Disorders|||||
CXT|C0234233|S0409125|SNM2|F-82610|1|CCP||Tenderness|C0234233|F-82610|||
CXT|C0234233|S0409125|SNMI95|F-A2610|1|ANC|1|SNOMED International|C0220967||||
CXT|C0234233|S0409125|SNMI95|F-A2610|1|ANC|2|FUNCTION|C0031843||||
CXT|C0234233|S0409125|SNMI95|F-A2610|1|ANC|3|BIOLOGICAL FUNCTIONS OF THE NERVOUS SYSTEM|C0234067||||
CXT|C0234233|S0409125|SNMI95|F-A2610|1|ANC|4|SENSATION AND PAIN|C0234185||||
CXT|C0234233|S0409125|SNMI95|F-A2610|1|ANC|5|TYPES OF PAIN|C0234223||||
CXT|C0234233|S0409125|SNMI95|F-A2610|1|CCP||Tenderness|C0234233|F-A2610|||
```

Figure 2: A request for concept context. The first five lines, labeled ANC and consecutively numbered one through five, represent a path through a tree. The CUI should appear after the name of each node. Note that the first five node names have no CUI, and so are not concepts in the Metathesaurus. The sixth line, labeled CCP, is the concept we searched on, Tenderness, a child of Pain Observations. The five lines below Tenderness, labeled CHD, are some of the children of Tenderness, each of which has a CUI except the one we are looking for, Abdominal Tenderness. Two more trees are shown below the first.

---

of the exercise.

If we assume that all terms in TraumaGEN are mapped by hand to CUIs in the Metathesaurus, we are left with the task of creating semantic hierarchies to enable the term subsumption discussed earlier. This task also presents problems that are difficult to solve in an automated fashion.

For example, while the Metathesaurus will return the semantic trees for a specific CUI from each KS, not every concept that appears in the trees is a concept in the Metathesaurus (see Figure 2). This makes it very difficult to compare trees between KSs, or to create more complete trees.

9 Conclusions

The process of automating the integration of TraumaGEN and the Metathesaurus is fraught with difficulties. Though the attempt at implementation first stopped at Web interface difficulties (since the Metathesaurus uses a site secured by certificates), the design process was also stymied at developing algorithms for a translator and mediator. Thus this paper is a chronicle of the (currently) overwhelming problems facing automated integration of heterogeneous knowledge sources.

A survey of the current literature shows that other systems have not been drastically more successful, unless their scope was sufficiently narrowed so that it was no longer comparable to this integration project.

Problems that will continue to be major barriers to integration are:

- inconsistent, incomplete, or conflicting data
- false data that is not contradicted
- the need to observe or calculate large amounts of information about the nature of a knowledge source and its contents before said contents can be used for effective integration.

10 Speculation

As noted earlier, the popularity of the BargainFinder project hinted at the demand for data integration. We can assume that if even moderately powerful integration agents are available to the public, then the public will use them.

The success and popularity of systems like AltaVista and Yahoo rely on a consistent method of encoding information which has been uniformly adopted across the Web: HTML. Likewise a successful integrator will depend on the presence of extensive information about the sources it mines. Sources which do not provide such information will be ignored by the users who increasingly depend on integration agents.

Is it likely that KS owners will spend substantial resources on the development of meta-knowledge about their systems? The power of corporations to mandate such processes internally may fuel the development of agent systems which can take advantage of such knowledge to integrate KSs. If so, and if such agents are made freely available on the Web, then the eventual result will be that organizations and individuals who want (need) computers to find and use their sites will pay the cost of meta-knowledge development. This type of cost is already being born by companies who hire programmers to enhance sites, trying to ensure their site's appearance on various "hot-lists".

Adoption of meta-knowledge standards (like KQML) will depend on the clout of early successful mediators. A sufficiently successful internet shopping aid, for example, could encourage vendors who wish to be included to represent their information according to the shopping aid's standard. (This assumes that the vendors expect that they will benefit from the free flow of information, a conclusion which was common, but not unilateral among the subjects of BargainFinder).

The near-term success of knowledge integration depends on the development and wide adoption of standards for encoding meta-information. Examples of easily derived erroneous information in this paper should give pause to those considering large scale semantic analysis of text as a preliminary step in knowledge integration.

References

S. Adali and V.S. Subrahmanian. 1994. Amalgamating knowledge bases, III: Distributed mediators. *International Journal of Intelligent Cooperative Information Systems*.

D. Brill, 1991. *LOOM Reference Manual*. USC-ISI, 4353 Park Terrace Drive, Westlake Village, CA 91361, version 1.4 edition, August.

Sandra Carberry and Terry Harvey. 1997. Generating coherent messages in real-time decision support: Exploiting discourse theory for discourse practice. In *Nineteenth Annual Conference of the Cognitive Science Society*, pages 79–84, Palo Alto, California.

S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom. 1994. The tsimmis project: Integration of heterogeneous information sources. In *Proceedings of IPSJ Conference*.

K.S. Decker, V.R. Lesser, M.V. Nagendra Prasad, and T. Wagner. 1995. MACRON: an architecture for multi-agent cooperative information gathering. In *Proceedings of the CIKM-95 Workshop on Intelligent Information Agents*, Baltimore, MD.

K. S. Decker, K. Sycara, and M. Williamson. 1997. Middle-agents for the internet. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, pages 578–583, Nagoya, Japan, August.

O.M. Duschka and A.Y. Levy. 1997. Recursive plans for information gathering. In *ijcai97b*, ijcai97a, ijcai97m.

T. Finin, R. Fritzson, D. McKay, and R. McEntire. 1994. The kqml information and knowledge exchange protocol. In *Proceedings of the Third International Conference on Information and Knowledge Management (CIKM-94)*, Gaithersburg, MD.

A. Gertner and B. L. Webber. 1996. A Bias Towards Relevance: Recognizing Plans Where Goal Minimization Fails. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, Portland, OR.

J. Hammer, H. Garcia-Molina, J. Cho, R. Aranha, and A. Crespo. 1997. Extracting semistructured information from the web. In *Proceedings of the Workshop on Management of Semistructured Data*, Tucson, Arizona, May.

Chun-Nan Hsu and Craig A. Knoblock. 1996. Discovering robust knowledge from dynamic closed-world data. In *aaai96b*, aaai96a, aaai96m. aaai.

Thomas Kirk, Alon Y. Levy, Yehoshua Sagiv, and Divesh Srivastava. 1995. The information manifold. Technical report, AT&T Bell Labs.

C. Knoblock and J. Ambite. 1997. Planning by rewriting: efficiently generating high-quality plans. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, Providence, Rhode Island, July.

Craig Knoblock, Yigal Arens, and Chun-Nan Hsu. 1994. Cooperating agents for information retrieval. In *Proceedings of the Second International Conference on Cooperative Information Systems*, Toronto, Canada.

B. Krulwich. 1995. Bargain finder agent prototype. Technical report, Anderson Consulting. `http://bf.cstar.ac.com/bf/`.

C. Kwok and D. Weld. 1996. Planning to gather information. Technical Report 96-01-04, University of Washington, Department of Computer Science and Engineering, January. Available via FTP from `pub/ai/` at `ftp.cs.washington.edu`.

Y. Papakonstantinou, H. Garcia-Molina, and J. Widom. 1995. Object exchange across heterogeneous information sources. In *Proceedings of the Data Engineering Conference*, Taipei, Taiwan.

J. R. Quinlan. 1986. Induction of decision trees. *Machine Learning*, 1(1).

Bonnie L. Webber, Ron Rymon, and John R. Clarke. 1992. Flexible support for trauma management through goal-directed reasoning and planning. *Artificial Intelligence in Medicine*, 4:145–163.