August 4, 1995

Collaborative Development of the InterMed Vocabulary Model

I. Goals of the InterMed Collaboratory

The InterMed Collaboratory is a recently formed consortium involving medical informatics researchers who have an interest in network-based collaborative-development activities. The laboratories currently participating are: 1) the Department of Medical Informatics at Columbia University, 2) the Section on Medical Informatics at Stanford University, 3) the Decision Systems Group at Brigham and Women's Hospital, 4) the Laboratory of Computer Science at Massachusetts General Hospital, and 5) the Department of Medical Informatics at the University of Utah. The collaborators are working on ways to share medical informatics resources from their various sites and to co-develop new resources by using the Internet as the primary means for interaction.

If participating sites are to share computer-based medical resources that were developed independently for a variety of purposes at distributed locations, they must have ways of dealing with the heterogeneity of medical terminologies used by those applications and institutions. In addition, as new applications are developed, it would be valuable to be able to turn to a national standard for controlled clinical vocabularies. However, if a national standard is to evolve and enjoy widespread use, robust methodologies must be devised that will enable such a vocabulary to be developed, maintained, and used. The Collaboratory aims to develop a generic vocabulary model in response to this national need, to demonstrate its relationship to existing vocabularies (including the NLM's Unified Medical Language System), and to implement the resulting model as a medical vocabulary server accessible via the Internet (Figure 1). Thus, a major research focus of the InterMed project is the collaborative development of a vocabulary model that is compatible with the needs of applications at the various sites also suitable for implementation as a prototype server accessible on the Internet.

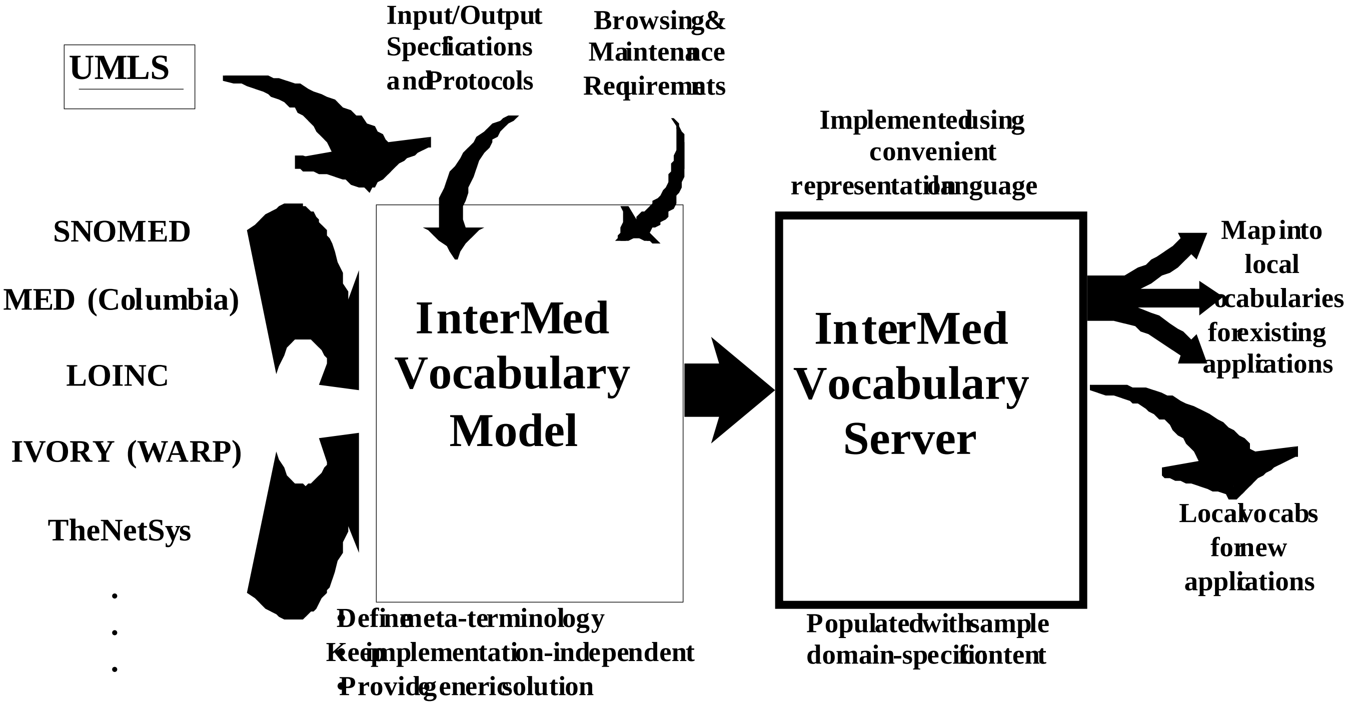
UMLS
Input/Output Specifications and Protocols
Browsing & Maintenance Requirements
Implemented using convenient representation language
SNOMED
MED (Columbia)
LOINC
IVORY (WARP)
TheNetSys
. . .
InterMed Vocabulary Model
InterMed Vocabulary Server
Map into local vocabularies for existing applications
Local vocabs for new applications
•Define meta-terminology
•Keep implementation-independent
•Provide generic solution
Populated with sample domain-specific content

Figure 1. InterMed's Collaborative Model Development Approach

There are accordingly two major issues to consider in this work: the design of the vocabulary model, including the input/output specifications for a vocabulary server (box at left in Figure 1), and its subsequent implementation as a server populated with concepts that can provide services to users (box at right). The left box, therefore, depicts design and the right box depicts implementation. When a server is implemented and in use, the server vocabulary can be used by mapping the server vocabulary to local vocabularies of existing applications or can be downloaded to local sites for new application development (arrows at far right).

This paper focuses on the left box in the diagram. That is, it describes only the vocabulary model and the input/output specifications for a server. It does not state what the best languages or systems would be for implementation. It does, however, attempt to define the requirements that an implemented system should fulfill. In order to demonstrate the model, the InterMed group plans to create a prototype implementation using a convenient representation language and using a collection of sample concepts to populate the vocabulary.

The InterMed model as shown above includes 1) a generic vocabulary model, and 2) input/output specifications for a vocabulary server. Also required for a vocabulary model for clinical medicine is a domain-specific model. The differences between the generic model and the domain-specific model are described below. However, the InterMed group is not planning to complete the domain-specific model that would be required for full implementation of a clinical vocabulary server.

It is appropriate to focus first on the design of the vocabulary model to determine what information will be stored about each concept and how the concepts will be organized.   After the vocabulary model has been developed, one can then focus on the services that will be provided by the vocabulary server. This makes it possible to describe functional specifications for the vocabulary server software according to the vocabulary model previously defined.

The process of designing the InterMed vocabulary model and server specifications has been a collaborative process involving the InterMed institutions, each of which has its own perspective and its own requirements for controlled medical vocabularies. The emphasis in each of these laboratories is on controlled vocabularies intended primarily for clinical purposes rather than primarily for billing or literature retrieval. Columbia uses the Medical Entities Dictionary (the MED) to provide an organized collection of terms that represent the medical concepts used by computer systems at their medical center. Stanford has dealt with a controlled vocabulary in their T-HELPER system for recording outpatient data. Massachusetts General Hospital has a long history of maintaining the controlled vocabulary for the Computer-Stored Ambulatory Record (COSTAR) and is currently developing a clinical workstation that has similar requirements for a controlled vocabulary. The group at Brigham and Women's Hospital has worked with structured data entry of clinical terminology for radiology reports. The University of Utah has been using controlled vocabularies in their clinical information systems for years and continues to move forward with efforts to coordinate their vocabularies from disparate systems. Local needs and experience therefore have contributed to the chosen design.

In addition to the influence of local vocabularies, the design of the InterMed vocabulary has been influenced by other existing vocabularies available nationally. The Unified Medical Language System (UMLS) is important because 1) it provides translations among different vocabularies, 2) it emphasizes unique identification of a concept based on its meaning, and 3) it provides some broad groupings of concepts by meaning. Some of the source vocabularies of the UMLS with broad clinical relevance have also influenced InterMed design. An important feature of many of the source vocabularies is the use of a hierarchical structure of concepts based on meaning. Examples of UMLS source vocabularies that have most influenced the InterMed vocabulary developers are Medical Subject Headings (MeSH), International Classification of Diseases, 9th edition, Clinical Modification (ICD-9-CM), Current Procedural Terminology, 4th edition (CPT-4), Diagnostic and Statistical Manual of Mental Disorders (DSM-IV), and the Systematized Nomenclature of Human and Veterinary Medicine (SNOMED International).

Unlike MeSH which was designed for access to medical journal articles in Medline, unlike ICD-9-CM and CPT-4 which are used in the United States for billing purposes, and unlike UMLS which was designed to enhance conceptual connections between users and information sources[1], the InterMed vocabulary is driven to a greater extent by the applications needed to support clinical care. In order for software to be useful and support the needs of end-users, it should be designed with an understanding of who the end-users are and what the applications are that will make use of the vocabulary. For patient care systems, the types of applications that a clinical vocabulary server would be designed for are applications intended for clinical data entry, clinical data retrieval, and decision-support.

Because knowledge and human language are not static but ever-changing, controlled vocabularies used in software applications are also ever-changing. This is particularly true in the practice of medicine, which changes rapidly. A vocabulary must be maintained over time to keep it up-to-date, and it should remain backwards-compatible with previous versions of the vocabulary.

In the domain of clinical medicine, a vocabulary that provides broad coverage of medical concepts to serve the electronic medical record will undoubtedly be very large and grow over time. Issues of both size and maintenance then are especially important for a model intended to support clinical vocabularies, and these issues present some significant challenges. For example, in a controlled vocabulary of mammoth proportions, it may be difficult for a user or application program to find a desired concept. It will also be difficult to assure that a new concept being added does not already exist by some other name. In addition, in order to use large collections of concepts, it is helpful to have them grouped according to meaning and be able to use subsets of related concepts. For these reasons, the concepts in a large vocabulary should be well-organized and appropriate groupings or classifications should be included to provide structure that facilitates both use and maintenance.

Development of a clinical vocabulary model requires that decisions be made regarding what information is specified about each concept and how the concepts are organized into a

[1] Humphreys BL and Lindberg DAL. The UMLS project: making the conceptual connection between users and the information they need. *Bulletin of the Medical Library Association* 81(2) April 1993, pp. 170-177.

usable, maintainable structure. One can describe a general vocabulary model that would be general enough to represent concepts in any domain. Such a model would provide a framework for storing and retrieving concepts. However, a general vocabulary model is insufficient for multiple clinical experts to contribute independently to a large collection of concepts and classifications. The clinical expertise required to create and maintain a controlled vocabulary for clinical medicine is not to be underestimated, and the task will have to be shared by multiple experts in a wide range of specialties. If some basic decisions can be made about the major classifications that are necessary for medical concepts stored in an electronic medical record, then multiple contributors can contribute to the evolving vocabulary. If the principles of clinical concept organization are clearly defined, then a contributor should be able to contribute a module of concepts in a subdomain that is compatible with the rest of the vocabulary.

There is then a distinction between a general vocabulary model that specifies a general structure of representing and storing concepts and the clinical vocabulary model, or domain-specific model, that specifies high-level concepts in the clinical classification scheme and provides patterns to follow for clincial concept organization. The clinical vocabulary model allows users to become familiar with a consistent structure in the vocabulary and allows maintainers to maintain that consistency. If a large vocabulary were to be developed in a domain other than clinical medicine, the general vocabulary model could also be used but a different domain model would be necessary for that particular domain.

A distinction should also be made between model development and content development. Model development is the process of planning a vocabulary framework. Content development is the process of populating the vocabulary with clinically-relevant concepts. The general vocabulary model requires no content development. The domain-specific vocabulary model does require some content development in order to specify high-level concepts and principles of clinical concept organization.   Finally, the process of populating the system with concepts and filling in all the expected information about those concepts would be the bulk of the content development process.

In this paper, we are emphasizing model development rather than content development. We will first describe the general vocabulary model in detail. We will then describe what would be required to create a domain-specific clinical vocabulary model but will only give examples of what it might actually contain. The InterMed group does not intend to build a comprehensive vocabulary that is fully populated with medical concepts. Such a major engineering task will be a costly endeavor, although it will be less costly to have multiple parties contributing to a shared resource than to have the same work duplicated by many. Without a principled, structured model that facilitates use and maintenance of the vocabulary, a large and ever-growing system will become so unwieldy that nobody will be able to use it, contribute to it, or maintain it.

As in any software development, the price paid downstream for poor planning early on can be extremely high, and a controlled medical vocabulary will rapidly become obsolete if it is too costly to use or maintain. It is therefore crucial to design a system that follows a model both users and maintainers can understand. It must be a model, which, when implemented and populated with concepts, can support applications used in clinical care.

The goal of this research is to design such a model.

II. Vocabulary Model Development

A. General Vocabulary Model

The general vocabulary model is a specification of the information that must be stored about each concept in the vocabulary. In general, the information that must be stored is identifier information, definitional information, and information about where the concept is placed in concept hierarchies.

In the general model, the following information is specified for each concept in the vocabulary:

1) Concept unique identifier
2) Concept preferred name
3) English definition
4) Synonyms
5) Foreign language translations
6) External coding scheme translations
7) Free text documentation
8) IS-A parents
9) Non-IS-A hierarchical relationship parents
10) Defining/differentiating characteristics

<u>Concept unique identifier</u>

A meaningless numeric or alphanumeric identifier (or code) that is not case-sensitive is assigned to each concept. The identifier is unique. Once an identifier is assigned to a concept, it is always associated with the meaning of that concept. If the meaning truly does not change, the identifier does not change. If a unique identifier is retired from use because a concept with that meaning is no longer useful, it is never used again for a concept with a different meaning.

<u>Concept preferred name</u>

A preferred name is assigned to each concept. The meaningless unique identifier is not immediately comprehensible to a human reader and therefore an English name must be available. The concept name should be nonvague and as unambiguous as possible. The preferred name is in English. The preferred name must be unique for the vocabulary at that time. No other concept with a different meaning can share the preferred name. If one is tempted to assign two different concepts the same preferred name, one or both of the names should be modified so as to indicated the difference in their meanings. A preferred name could be reused for a concept with a different meaning if it was previously retired from use although this practice is to be discouraged if possible. This is because the preferred name is not the unique identifier that stays constant over time. The meaningless code is the unique identifier over time.

English definition

Every concept has an English definition. The English definition helps human readers understand the meaning of the concept. The definition is a chunk of readable text. Thus, the vocabulary serves as a dictionary.

Synonyms

Synonyms are supported as a means of further explaining what the meaning of a term is. Synonyms are alternate names in English commonly used for the same concept. Thus, the vocabulary serves as a thesaurus. A term designated as a synonym does not have to be a unique name with respect to the rest of the vocabulary. Synonyms may include acronyms and abbreviations.

Foreign language translations

Each concept can be assigned names in other natural languages such as Spanish, French, and German. The name of the language is specified for each translated concept. The mapping between the concept and the name in another language should be a synonym mapping. That is, they should have the same meaning.

External coding scheme translations

Each concept can be assigned names or codes in other external coding schemes such as the UMLS, ICD-9-CM, and SNOMED International. The name of the external coding scheme is specified for each translated concept. If each concept in the vocabulary has a UMLS code assigned to it if one exists, then access to the UMLS would allow for translation to the source vocabularies in the UMLS.

Free text documentation

Each concept has a block of free text associated with it for any additional annotations about the concept that the authors wish to include.

IS-A parents

Each concept must be placed in the IS-A type hierarchy. To specify the concept's location in the hierarchy, the concept's parents must be specified. A concept can have more than one parent. Each child concept is a specialization of its parent concepts, and each parent concept is a generalization of its child concepts.

Non-IS-A hierarchy parents

There are other relationships beside IS-A that organize concepts into hierarchies. Examples in medicine include ANATOMIC-PART-OF, ARTERIAL-BRANCH-OF, VENOUS-TRIBUTARY-OF, and LAB-TEST-COMPONENT-OF. Although every concept in the vocabulary must have its IS-A parents specified, this is not true of non-IS-A relationships. However, a concept in a non-IS-A hierarchy that has children must have parents as well. This assures that the hierarchy will not be discontinuous.

Defining/differentiating characteristics

Each concept must have characteristics specified that define it and differentiate it from its parents. The characteristic has a name and a value. Thus, a concept would be defined by saying that it "IS-A such-and-such" where such-and-such is the parent and therefore by inheritance the concept takes on the characteristics of its parent, but it also has additional characteristics that further specialize the concept and differentiate if from its parent. There is no limit to the number of characteristics that can be specified for a concept, but each characteristic should be a "defining" characteristic. That is, to the extent that it is possible, the goal is for only characteristics that are definitional to be included. Other non-definitional knowledge about a concept would be more appropriately stored in a separate knowledge base.

Some concepts may not easily be specified by defining characteristics and are often called primitives.

Defining characteristics frequently take on values that are in turn other concepts stored in the vocabulary. This provides links between related concepts in the vocabulary and specifies what the link is.

B. Domain-Specific Vocabulary Model

The domain-specific vocabulary model specifies the principles about the hierarchical structure or defining characteristcs that must be adhered to in order for consistency to be maintained in the vocabulary.   This requires specification of some of the high-level concepts . It is therefore domain-specific and requires domain expertise and content development.

In addition, definition templates provide constraints that allow consistency to be maintained. Some kinds of concepts will be similar enough that they share the same set of characteristics. Such a set of characteristics specified for a particular kind of concept is called a definition template. It is a template or pattern that is to be followed for concepts of that kind. To illustrate this idea of a definition template, two examples are given below. The sample DISEASE definition template contains up to six characteristics that could be specified to describe a particular disease, and the sample LABORATORY-TEST template contains six characteristic that must all be specified in order to describe a particular laboratory test.

DISEASE

- Etiology
- Involves-site
- Functional-process
- Morphologic-change
- Temporal-quality
- Severity-level

LABORATORY-TEST

- Analyte-measured
- Property-measured
- Timing
- Specimen
- Precision
- Method

Because a structured vocabulary contains domain knowledge, there are always boundaries that must be defined that draws the line between what should be stored in the vocabulary and what should be stored in separate knowledge bases. A structured medical vocabulary cannot be expected to contain all of the knowledge of medicine. However, some of these boundaries must be agreed upon and made explicit in order to assure consistency. For example, information that should be stored in the vocabulary for a drug would likely include the active chemical ingredient and classification information such as whether it is an antibacterial drug, and antiviral drug, an antifungal drug, or an antiparasitic drug. Reasons for including such knowledge are that it helps to organize the large number of drugs, helps users find the drugs they are interested in, and allows the user to retrieve sets of similar concepts. However, it would probably be too much to expect for the vocabulary server to include information about which antiviral drugs have efficacy against which viruses and which drugs are currently recommended for the treatment of which diseases. It would be more appropriate to store such knowledge in knowledge bases that make use of the nomenclature provided by the vocabulary but that are maintained separately by domain experts who are responsible for keeping this kind of knowledge up-to-date. Constraints and rules that draw the line between the vocabulary and other knowledge bases should be stated as explicitly as possible so that both users and maintainers clearly understand what the vocabulary contains and what it does not.

Thus, high-level concepts, definition templates, and rules that describe the extent of vocabulary content are all part of the domain-specific vocabulary model. The InterMed group is not attempting to fully specify the domain-specific vocabulary model for clinical medicine but emphasizes that the evolution of a domain-specific model is crucial for ensuring consistency in a large vocabulary to which multiple experts contribute.

C. Vocabulary Maintenance Model

The maintenance model specifies the information that must be stored to record the changes made to the vocabulary. Every change, including the initial entry of a concept to the vocabulary, must have an author and a timestamp associated with it. Rules that constrain the types of changes that can be made and specify how those changes are handled to maintain conistency of the vocabulary are also part of the maintenance model.

1. ADDITIONS

When a new concept is added, the appropriate information about the concept as described above in the general vocabulary model must be specified. It must be placed in the IS-A hierarchy in at least one place. That is, it must have at least one IS-A parent specified. It may have more than one IS-A parent specified. The union of all the characteristics of all its parents must be valid for the new concept.

2. MODIFICATIONS

Modification to existing concepts are divided into two groups: 1) changes that can be made to a concept without altering its meaning, and 2) changes that are made to a concept that do alter its meaning.

Changes that do not alter the meaning of a concept are those that modify names or labels that identify the concept. These include concept preferred name, English definition, synonyms, foreign language translations, external coding scheme translations, and free text documentation.

Changes that do alter the meaning of a concept must have a new concept unique identifier assigned. If the old concept is still useful, then this type of concept modification is actually an addition of a new concept. If the old concept is no longer useful and is to be retired, the new concept can be placed in the hierarchy in the same place as the old concept if that location is still valid. If it is not appropriate to place the new concept in the old location, then the concept is placed in the most appropriate place, and the old concept is deleted. Examples of changes to a concept that will alter the meaning are changes of IS-A parents, changes of non-IS-A parents, and changes of defining characteristics and their values. If a new concept is brought in to replace an old concept, then a reference to the obsolete concept is recorded.

3. DELETIONS

When a concept is deleted, it is retired from use but is accessible to future users. The information on the deleted concept is kept but the concept is designated as no longer active or not used. The unique identifier for the deleted concept can never be used again for a concept with a different meaning.

III. Functional Specifications for a Vocabulary Server

The functional specifications for a vocabulary server describe the inputs to and outputs from the server for users or applications that have READ capability and the inputs to and changes made to the contents of the server for users or applications that have WRITE capability.

A. READ Queries

For users or applications that have READ capability, the following queries are supported:

1. For given"concept-string" get concept preferred name(s) that match most closely lexically
2. For given "concept-string" get concept preferred name(s) of concept that matches most closely in meaning
3. For given concept preferred name, get concept identifier (ID)
4. For given concept ID, get concept name
5. For given concept ID, get English definition
6. For given concept ID, get synonyms
7. For given concept ID, get foreign language translations
8. For given concept ID, get external coding scheme translations
9. For given concept ID, get free text documentation
10. For given synonym, get concept preferred name(s)
11. For given external coding scheme ID, get concept ID(s)
12. For given foreign language translation, get concept ID(s)
13. For given concept ID, get all direct IS-A parents
14. For given concept ID, get all IS-A ancestors
15. For given concept ID, get self plus all IS-A ancestors
16. For given concept ID, get all IS-A direct children
17. For given concept ID, get all IS-A descendents
18. For given concept ID, get self plus all IS-A descendents
19. For given concept ID, get all parents for a particular non-IS-A hierarchical relationship
20. For given concept ID, get all ancestors for a particular non-IS-A hierarchical relationship
21. For given concept ID, get self plus all ancestors for a particular non-IS-A hierarchical relationship
22. For given concept ID, get all direct children for a particular non-IS-A hierarchical relationship

23. For given concept ID, get all descendents for a particular non-IS-A hierarchical relationship
24. For given concept ID, get self plus all descendents for a particular non-IS-A hierarchical relationship
25. For given concept ID, get defining characteristic names and values (including characteristics inherited from ancestors)
26. For given concept ID and given defining characteristic name, get value of characteristic
27. For given defining characteristic name, get all concept IDs whose value is "some-value"
28. For given concept ID, get defining characteristics names and values (not including defining characteristics inherited from ancestors) plus IS-A parents and non-IS-A parents
29. For given concept ID, get maintenance information

B. WRITE Queries

For users or applications that have WRITE capability, the following queries are supported:

The following changes can be made without changing a concept unique ID:

1. Add new concept
2. "Retire" concept that has become obsolete
3. For given concept ID, change preferred name
4. For given concept ID, change English definition
5. For given concept ID, remove a synonym
6. For given concept ID, add a synonym
7. For given concept ID, add a foreign language translation
8. For given concept ID, remove a foreign language translation
9. For given concept ID, change a foreign language translation
10. For given concept ID, add an external coding scheme translation
11. For given concept ID, remove an external coding scheme translation
12. For given concept ID, change an external coding scheme translation
13. For given concept ID, add free text documentation
14. For given concept ID, remove free text documentation
15. For given concept ID, change free text documentation

The following changes can only be made by changing the concept unique ID as well:

1. For given concept ID, add new IS-A parent
2. For given concept ID, remove an IS-A parent
3. For given concept ID, add new parent for a particular non-IS-A hierarchical relationship
4. For given concept ID, remove parent for a particular non-IS-A hierarchical relationship
5. For given concept ID, add a new defining characteristic and its value
6. For given concept ID, remove a defining characteristic and its value
7. For given concept ID and defining characteristic, change the value of that characteristic

IV. Application Software Provided to Vocabulary Server Users

The vocabulary server itself is a collection of concepts stored on a server that can accept inputs and return outputs through an application programming interface over a network. Many users however will be interested in viewing the concepts and maintainers will be interested in changing the concepts without writing their own software to send queries to the server. Therefore, the two most important applications that users will want to have available to them will be a browser and an editor.

A browser allows a person to view concepts and information about the concepts in a read-only mode. It should have an easy-to-use interface that allows the user to perform tasks such as navigating hierarchies, expanding and contracting subtrees in the hierarchies, viewing unique identifier codes as well as preferred names, viewing the characteristics of each concept, and tracing back the evolution of the concept over time and document authorship. The browser should make it easy for users to download the entire vocabulary or subsets of the vocabulary to their local computer.

An editor allows a person not only to view concepts and information about the concepts but also to make additions, modifications, and deletions. It should have basically the same easy-to-use interface that allows viewing in read-only mode but it should provide extra functions that allow the user to make changes to the vocabulary. The editor is also responsible for managing the log of changes over time.

An editor used by a single user does not have to deal with problems of concurrency control that occur with multiple users trying to send queries simultaneously. As in traditional databases, if the vocabulary was in a consistent state before a transaction, then it must be in a consistent state at the end of the transaction. A single-user editor must only ensure that consistency is maintained before and after each individual transaction where transactions occur sequentially. A multi-user editor however must ensure that transactions initiated by multiple users will be not conflict and corrupt the vocabulary. In the short term, the easiest solution is to require that only one user be allowed to make changes at a time. However, in the future, it will be important to have mechanisms in place that allow multiple users at different sites to work on maintaining various portions of the vocabulary at the same time.

If local vocabularies are expected to persist either because legacy vocabularies must be maintained or because different sites have different local needs and therefore must maintain concepts that others do not need, it would be desirable to provide translation support software along with the browser and editor. Translation support software would assist a user in making mappings between a local vocabulary and the server vocabulary. It would also provide assistance in coordinating changes that occur in the local vocabulary with changes that occur in the server vocabulary and keeping them synchronized over time. This is a separate research problem however.

The InterMed group considers browser and editor software as essential applications for a vocabulary server and will include them in the prototype implementation.

V. Conclusion

Development of the InterMed vocabulary model and server specifications has been a collaborative effort by the InterMed Collaboratory. The goal is to provide a general approach for management of large structured vocabularies that are shared by many users over the Internet, but a particular emphasis has been placed on the needs of users and applications in clinical medicine. The model provides a framework within which concepts can be stored, retrieved, and kept up-to-date. Full implementation of the model for clinical medicine would require consensus about the domain-specific model that would support the electronic medical record, and comprehensively populating the system with concepts would be necessary. However, the first step is to design the model. The second step is to build a small prototype system that demonstrates how the model could be implemented, what a browser and editor might look like, and how applications could interact with the server over the Internet. From this prototype, we hope to provider a clearer understanding of what a vocabulary server should be.