

Desiderata for a Clinical Terminology Server

CG Chute¹, PL Elkin¹, DD Sherertz², MS Tuttle²

(1) Mayo Foundation, Rochester, MN; (2) Lexical Technology, Alameda, CA

ABSTRACT

Clinical terminology servers are distinguished from more broadly based terminology servers intended for nomenclature development or mediation across classifications. Focusing upon the consistent and comparable entry of clinical observations, findings, and events, key desiderata are enumerated and expanded. These include 1) word normalization, 2) word completion, 3) target terminology specification, 4) spelling correction, 5) lexical matching, 6) term completion, 7) semantic locality, 8) term composition and 9) decomposition. Comparisons of this functionality to previously published models and specifications are made. Experience with a clinical terminology server, Metaphrase, is described.

INTRODUCTION

The emergence of electronic medical records has fostered a new recognition of clinical terminologies and nomenclatures. Increasingly, an emphasis upon consistently entered information, that is comparable across care providers and care settings, is being emphasized. To achieve this, developers and implementers of Electronic Patient Records (EPRs) are discovering that clinical information needs to be expressed using comprehensive and well-structured terminologies.

To achieve efficiently this goal, human interfaces to computer applications that serve up terminology contents need to be developed. Figure 1. illustrates a high level relationship between such a server and a user application. As proposed during the first CPRI sponsored National Terminology Summit¹ the problem can be partitioned among Entry, Reference, and Aggregate Classification functions. The invocation of "Terminology Server" could arguably apply to all three functions, indeed we review below efforts and proposals which purport to do all these and more. Yet from a clinical application perspective, one cares most about what the user actually sees, and how well a terminologically naïve person might interface against standardized descriptions of clinical findings, diagnoses, and interventions.

This paper attempts to state desiderata for the focused task of term entry by clinical users in a workstation context. These applications may vary substantially in their detailed tasks, as illustrated in Figure 2. Their common trait, however, is that they depend upon shared intermediary code against a common terminology database to support term entry and user composition of complex or composite concepts.

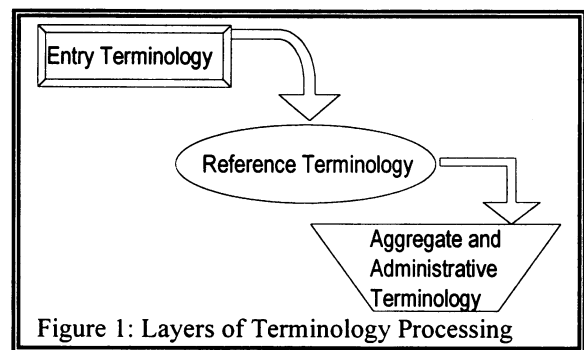


Figure 1: Layers of Terminology Processing

TERMINOLOGY SERVERS

Major Models and Proposals

The notion of a terminology server is not new. Indeed, many of the early UMLS descriptions broached the notion that terminologies would require databases to mediate translations among concepts shared across disparate terminologies². The reality of the UMLS is now more than a decade old. These were expanded to entertain the general problems of using terminologies in applications, as well as the challenge of supporting terminology development and maintenance³. Indeed, the distributed terminology development project within Kaiser Permanente designed by Campbell⁴ is fundamentally dependent upon a common terminology server.

The first generic description of general terminology servers *per se* was produced by the European Galen project⁵. In this seminal paper, Rector *et al.* outlined the functional attributes of the Galen Terminology Server. They specified two modes of use, those envisioned for operational and runtime systems, and those intended for terminology

development and maintenance. Within this context, they enumerated performance tasks:

- Managing external references.
- Managing internal representations.
- Mapping natural language to concepts.
- Mapping concepts to classification schemes.
- Management of extrinsic information.

Upon these, the authors expand in Socratic fashion by posing questions and tasks that a well behaved terminology server should answer. The vast majority of these tasks correspond to exploring and detailing the intrinsic properties and relationships of and among terms. In this, many have less to do with providing behaviors requisite to underpinning clinical applications and concept data entry.

Other papers on terminology servers have explored web infrastructures to support functional delivery of standard nomenclatures⁶. Le Beux's group extended this notion, to accommodate the specific functional needs of medical procedures⁷. The realization of a practical terminology server was most extended by the work of Tuttle *et al.*⁸ in their development and refinement of the Metaphrase™ engine, described in further detail below.

CORBAMED Lexicon Query Service

Perhaps the most ambitious definition document about terminology servers remains the tour de force by Solbrig and colleagues, in their detailed specification of a terminology IDL (Interface Definition Language)⁹. Responding to a CORBAMED RFP, their 175-page document outlines in unprecedented detail definitions, functional scenarios, a terminology reference model, the IDL interface, explicit meta-terminology, and conformance criteria for the specification.

This response owes much of its intellectual organization to the RFP that prompted its creation. In that document, mandatory and optional requirements for the CORBAMED terminology server were specified. Briefly, a simplified paraphrase of these requirements is:

Mandatory:

1. Enumeration of lexicon concepts and their associated information.
2. A means of retrieving a unique external identifier for each concept.
3. A means to retrieve attribute values which support translation to other coding schemes.

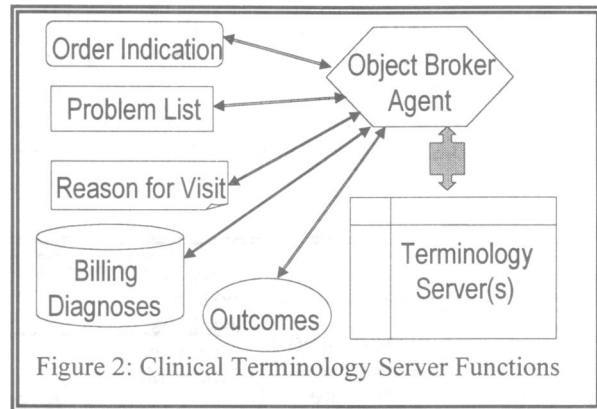


Figure 2: Clinical Terminology Server Functions

4. Enumeration of attribute types and relationships supported within a lexicon.
5. Enumeration of concepts which participate in specified relationships (*e.g.* hierarchical) with respect to a concept.
6. Enumeration of attributes and relationships for a given concept.
7. Enumeration of concepts corresponding to a specified attribute value.
8. Enumeration of concepts that satisfy multiple relationships and attribute value combinations.

Optional:

9. Support partial pattern matching or generic selection on matching criteria.
10. Support traversal of relationships within a lexicon.
11. Represent concepts as coordinated terms or composite entries.

Criteria 4, 5, and 10 seem oriented to an infrastructural support of terminologies. Items 1, 3, 7, and 8 seem less oriented to the clinical interface, whereas 2, 6, 9, and 11 correspond most closely to the focus of the present manuscript.

The response outlines functional scenarios, under which differing combinations of the requirements pertain. These scenarios include: Information Acquisition, Information Display, Mediation among Representations, Indexing and Inference, Browsing, and Composite Concept Manipulation. Of these, the first and last seem most relevant to our focus of a Clinical Terminology Server.

Clinical Terminology Server Distinguished

The major premise of a clinical terminology server is that it is used by clinicians to enter patient observations, findings, and events such as

procedures. It does not need to carry the weight of terminology updates, maintenance, or development, and thus might be regarded as a server "lite." However, unique demands and functional behaviors arise in what might be regarded as a data entry role of

Similarly, the scope of appropriate semantic types may differ substantially between say, an Indication for Order vs. a Chief Complaint or Reason for Visit; another negotiation managed by a middleware layer. These behaviors are schematized in Figure 3. and

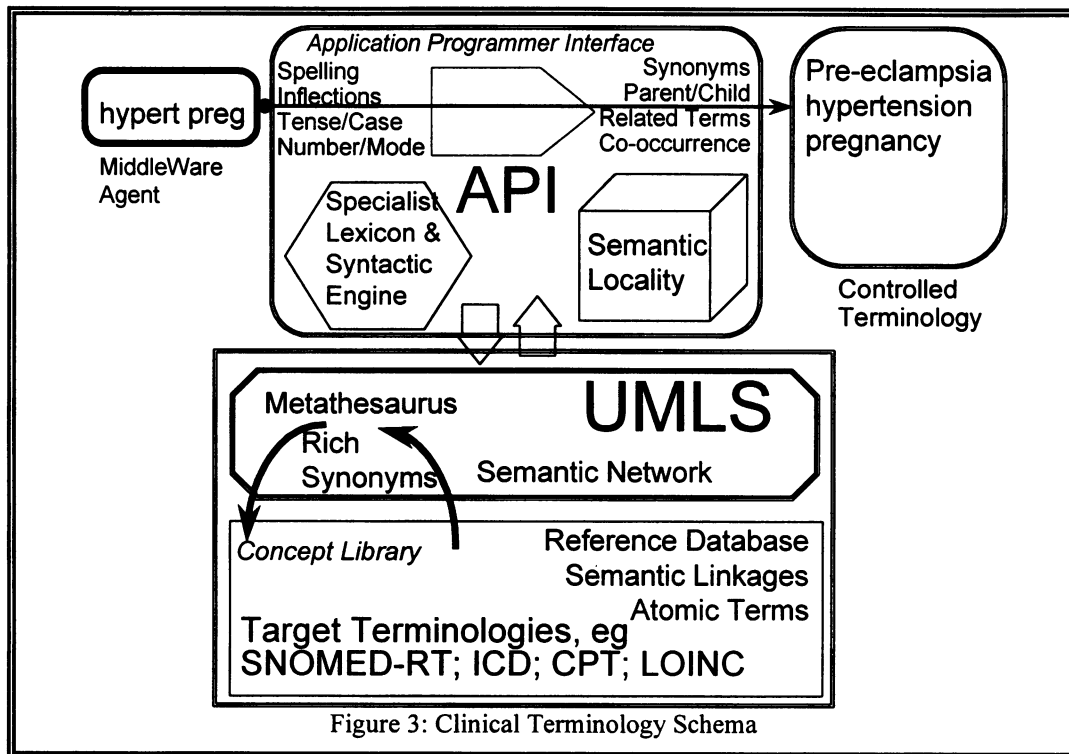


Figure 3: Clinical Terminology Schema

clinical servers that do not surface for development servers. These have more to do with helping a poor user find an imprecisely entered term than addressing elegant structures and relationships within a terminology.

Figure 2 outlines the schematic function of a clinical terminology server. Fundamentally, it can serve a variety of functions within a clinical enterprise. The designations such as Problem List and Reason for Visit merely designate a few. Importantly, these disparate functions can and should share an underlying terminology base server, though they need not specify a common target terminology. For example, Problem Lists may be in a highly detailed nomenclature such as SNOMED RT, yet Billing Diagnoses will be in classifications such as ICD-9-CM or CPT.

Partly to negotiate such functional variations, the insertion of a "broker" layer, typically in an object environment such as Java, facilitates these differentiating calls against a common server.

expanded in functional behavior by the specific desiderata.

DESIDERATA

1. Word Normalization

Humans, regardless of their mode of entry (typing or speech recognition), are prone to use lexical variants of words that may not match their corresponding representations in a nomenclature. Common variations include gender, tense, case, pluralization, possessive inflection, or punctuation. Such variations can impair the matching between entered words and phrases with target terminology entries. This problem was historically addressed by invoking word stemming algorithms which would strip word endings using patterned methods such as the familiar "ies" change to "y."

More elegant approaches to word normalization have arisen, including commercial "stemmers" with associated lexicons. Within the domain of biology and medicine, the emergence within the UMLS

framework of the Specialist Lexicon¹⁰ has fundamentally transformed this task. The excellence of the lexical knowledge base of word specific variants, coupled with highly efficient normalization and “lexical variant generator” code, comprise an outstanding resource which presently constitutes a “best of breed” mechanism for normalization.

Regardless of the software source, the functional requirement for intelligent lexical normalization is an undeniable criterion of a terminology server. Extension of this kind of functionality to multiple languages (e.g. French or Japanese) is clearly a next step¹¹.

2. Word Completion

This is an old notion, and is best regarded as a “wild card” word ending. For example, pn* would capture *pneumococcal* as well as *pneumocystis*.

3. Target Terminology Specification

Because an enterprise would benefit from a common terminology server, it must support different nomenclatures and classifications likely to be employed within the enterprise, such as SNOMED RT or ICD-9-CM. Thus, these targets need to be specifiable by a calling application, which is likely to have application specific needs for one target vs. another.

4. Spelling Correction

While word normalization and lexical matching offer substantial flexibility and advantage, there are some of us who remain challenged by the idiosyncrasies of spelling. The logical alternatives provided by INSO CorrectSpell™ products embedded within many modern office-suite computer products (including MS Word) have made us accustomed to intelligent spell checking and recommendations. While this level of functionality may not be economically feasible, behavior akin to the old UNIX `spell` program is a minimal expectation.

5. Lexical Matching

While obvious, the requirement that normalized strings, with or without spelling variants, be matched against a library of indexed words must not be overlooked. These words in turn map to term or synonym entries, which correspond to the entered text. Heuristics that optimize the number of matched words entered to coordinate phrases within a target lexicon become the heart of this criterion.

6. Term Completion

A non-obvious extension to lexical matching is the weighted consideration of terms likely to be dropped by knowledgeable application users. For example, seeking the concept *Turner's Syndrome* a user might enter simply “Turner,” expecting the system to complete the full term by adding the “syndrome” ending (note the possessive form is dispatched by the lexical normalization algorithms.) On the other hand, the obverse case of entering “syndrome” and expecting Turner's to surface is not reasonable. Thus, the term specificity of words needs to be computerized and maintained by the server to intelligently engage term completion.

7. Semantic Locality

The work of the Galen Terminology Server and the CORBAmed Lexicon Query Server emphasized the needs to understand and leverage relationships between terms. This becomes invaluable when leveraging the functional behavior of term entry applications by making visible closely related terms or concepts. We have previously demonstrated in usability laboratory¹² that users find terms displayed from the semantic neighborhood (child and sibling terms) helpful, and will often select such recommended terms.

8. Term Composition

This is a deceptively simple concept, but deviously difficult to engage. In essence, it assumes recognition that many terms are a composite of smaller elements; most composites comprise modifiers or qualifiers¹³ and a kernel concept. A server should propose coordinated standard terms that in combination capture the full notion intended, e.g. *Anterior* and *Myocardial Infarction*¹⁴.

9. Term Decomposition

Closely related to the notion of term composition is the breaking apart of complex phrases into its atomic components. Debate continues over exactly what constitutes an atomic component, e.g. whether *Colon Cancer* should be decomposed into the atomic elements of *Colon* and *Cancer*, or itself constitutes an atomic notion. Nonetheless, this functionality is important to the functional behavior of a clinical terminology server¹⁵.

DISCUSSION

Mayo has substantial experience prototyping the Metaphrase⁸ engine in a variety of clinical applications¹²⁻¹⁵. Architecturally, we have distributed the desiderata enumerated in this manuscript over an

n-tiered object environment, ranging from the lexical database, through middleware, and client applications. Nevertheless, the Metaphrase component functionally addresses many of the characteristics we have described, specifically Nos. 1-7. It explicitly incorporates the Specialist Lexicon resources¹⁰ within its word normalization components.

The issues of term composition and decomposition remain more experimental. While interim versions of Metaphrase had incorporated these capabilities, we have found most of this functionality more useful within the client. At this layer, the moderate CPU resources of modern workstations can be focused upon a difficult task, without compromising system performance across the network. More pertinently, iterative user feedback can be more readily incorporated as an optimal composition is built from standard lexicon elements.

The historical notion that a single terminology server, or even a singular design, might serve the needs of clinical applications and terminology developers has not been borne out. We attempt to articulate the functional needs of a terminology server oriented toward the clinical needs of care providers using applications in an operational environment. The criteria specified emphasize navigational ease over conceptual or structural elegance. We envision terminology server products emerging which target this need, as increasing numbers of providers and vendors recognize the critical need for consistent clinical descriptions in the now exploding EPR marketplace.

ACKNOWLEDGEMENTS

This work was supported in part by grants from the NLM (LM05416) and jointly funded by NLM and AHCPR (HS08751). We thank Kevin Keck and Phil Ogren for substantial help with the Metaphrase server and Mayo's Problem List application. Karen Elias has been invaluable with manuscript and administrative support.

REFERENCES

- 1 www.cpri.org/terminology
- 2 Lindberg D, Humpheys B, McCray A. The Unified Medical Language System. In: van Bommel J. Ed. *1993 Yearbook of Medical Informatics*. Stuttgart: Schattauer Verlag, 1993: 41-53.

- 3 Mays E, Weida R, Laker, M, et al. Scalable and expressive medical terminologies. *JAMIA Sympos Suppl* 1996:259-263
- 4 Campbell KE, Cohn SP, Chute, CG, et al. Scalable methodologies for distributed development of Logic-Based Convergent Medical Terminology. *Meth Inform Med* 1998; 37(4-5):426-439.
- 5 Rector AL, Solomon WD, Nowlan WA, et al. A Terminology Server for Medical Language and Medical Information Systems. *Meth Inform Med* 1995;34(1-2):147-57.
- 6 Gennari JH, Oliver DE, Pratt W, et al. A web-based architecture for a medical vocabulary server. *JAMIA* 1995;SympSuppl:275-9.
- 7 Burgun A, Denier P, Bodenreider O, et al. A web terminology server using UMLS for the description of medical procedures. *JAMIA* 1997;4(5):356-63.
- 8 Tuttle MS, Olson NE, Keck KD, et al. Metaphrase: An Aid to the Clinical Conceptualization and Formalization of Patient Problems in Healthcare Enterprises. *Meth Inform Med* 1998;37(4-5):373-83.
- 9 Solbrig H, Brinson T. *Lexicon Query Service: RFP Response*. OMG TC Document CORBamed/98-03-22. Murray, UT: 3M Health Information Systems, Protocol Systems Inc., 1998. <ftp://ftp.omg.org/pub/docs/corbamed/98-03-22.pdf>
- 10 McCray AT, Srinivasan S, Browne AC. Lexical Methods for Managing Variation in Biomedical Terminologies. *Proceedings of the 18th Annual Symposium on Computer Applications in Medical Care*, 1994: 235-239.
- 11 Wagner JC, Solomon WD, Michel PA, et al. Multilingual natural language generation as part of a medical terminology server. *MEDINFO'95* 1995;8(1):100-4.
- 12 Elkin PL, Mohr DN, Tuttle MS, et al. Standardized Problem List Generation, Utilizing the Mayo Canonical Vocabulary Embedded within the Unified Medical Language System. *JAMIA* 1997;Symp Suppl:500-504.
- 13 Chute CG, Elkin PL. A Clinically Derived Terminology: Qualification to Reduction. *JAMIA* 1997;Symp Suppl:570-574.
- 14 Elkin PL, Bailey KR, Chute CG. A Randomized Controlled Trial of Automated Term Composition. *JAMIA* 1998;Symp. Suppl.:765-769.
- 15 Elkin PL, Bailey KR, Ogren PV, et al. A Randomized Double-Blind Controlled Trial of Automated Term Dissection. *JAMIA* 1999;Symp. Suppl.: submitted.