From Terminology to Terminology Services

WA Nowlan* AL Rector§ TW Rush* WD Solomon§
*Medical Products Group, Hewlett-Packard Ltd, Bristol, BS12 6QZ, UK
§Medical Informatics Group, Department of Computer Science, University of Manchester, Manchester M13 9PL, UK.

ABSTRACT

Terminologies have traditionally been considered as static datasets held in books or databases. The GALEN Terminology Server presents a prototype for a new view of terminologies delivered as a set of functions and services provided to other applications. This facilitates their development and integration as part of a strategy for sharing and re-using information and knowledge. The essential features of the Terminology server are the functions which it can perform; questions which it can answer and statements which it can be told. The GALEN Terminology Server supports these operations through a modular architecture and uniform applications programming interface which allows client applications to ignore the internal structure and simply use the Server for terminological, coding, and linguistic functions.

INTRODUCTION: THE IDEA OF A 'TERMINOLOGY SERVER'

The development of 'vocabulary servers' has been mooted by various organisation in both the United States and Europe, but there is no general consensus on what such a server should be and how it should be used. This paper describes one vision of a slightly broader concept — a *Terminology Server* — and the progress towards its realisation in the GALEN[1] project sponsored by the European Union's AIM programme. The Terminology Server is seen as a key component in a broad strategy for developing and integrating clinical and related systems. It provides terminological services to applications, ranging from acting as a repository for vocabularies and lexicons, to the dynamic transformation and encapsulation of complex concepts, to support for sophisticated user interfaces.

Structure versus function

The authors [1-3]and others [4-6] have argued that if computer systems are to play a significant role in clinical care, then concept systems or 'ontologies', formally modelled, which can be manipulated by computer systems, are essential. The requirement to manage those complex concept systems leads to a choice between two options: either every application will itself have to deal with the concept model, or the services will have to be 'subcontracted' to a separate server that is general enough to support a wide variety of applications. GALEN aims to demonstrate the feasibility of the server option. The belief behind GALEN is that a wide range of applications have a common need for formal concept systems, linguistic support, and translations to and from existing coding schemes.

This Terminology Server option leads to a different view of terminologies. Traditionally terminologies have been seen as *static data structures* which could be written down or at least stored in a straight forward database Large schemes such as SNOMED-III, the READ Codes, and ICD-10 all provide one degree or another of prescriptive advice about how the coding system is to be used, but they are defined in terms of the structure rather than the functions performed. The belief within GALEN is that this is no longer adequate and the Terminology Server is described in terms of the *functions* and *services* it delivers to applications through its applications programming interface. These functions will require the manipulation of highly complex conceptual, linguistic and coding structures, but the application is deliberately protected from knowing the detail of these. The Terminology Server provides a high level uniform view of concepts, language, and codes.

Aims of the Terminology Server

The GALEN Terminology Server aims to:

- facilitate the development and integration of systems for patient care that can effectively deal with the detail, complexity, and heterogeneity of clinical information and clinical usage
- mediate between systems to integrate information sources, including conversion between coding schemes, transformation between the conceptual structures of different medical record and database schemata, encapsulation of complex data structures

1 General Architecture for Languages Encyclopædias and Nomenclatures in Medicine. The members of the GALEN consortium are: University of Manchester (UK, Coordinator), Hewlett-Packard Ltd (UK), Hôpital Cantonal Universitaire de Genève (Switzerland), Consiglio Nazionale delle Ricerche (Italy), University of Liverpool (UK), Katholieke Universiteit Nijmegen (Netherlands), University of Linköpking (Sweden), The Association of Finnish Local Authorities (Finland), The Finnish Technical Research Centre (Finland), GSF-Medis Institut, (Germany), Conser Systemi Avanzati (Italy)

0195-4210/94/$5.00 © 1994 AMIA, Inc.

in forms suitable for storage in relational databases, and linkage of knowledge-based and patient care systems
- provide natural language generation and, eventually, understanding, in multiple languages and facilities to simplify the development of multilingual clinical systems.
- support knowledge editing and acquisition, including the compilation and extension of coding and classification systems, support for the terminological needs of editors for knowledge based systems, and the developers of information systems models and user interface designs.

The server is designed to act as a repository of terminological and linguistic information both as a reference during system development and as a run-time service for existing systems. It is also aimed at supporting information sharing both within and across sites. Sharing and mediation will, of course, be easiest between applications developed using tools linked to the server. However facilities for mapping to pre-existing 'external representations' such as coding systems are provided. As regards the medical content of the server the medium term goal is to achieve collaborative distributed development of the large fund of detailed conceptual knowledge which will be needed to support the next generation of clinical applications.

Defining the functions required to support these applications and an architecture in which such development is feasible has been a major part of the GALEN project. The remainder of this paper describes this functionality, architecture, and briefly the implementation.

A FUNCTIONAL DESCRIPTION OF THE GALEN TERMINOLOGY SERVER

The basic description of the server is simple. Requests are made of the Terminology Server by an application and answers returned. There are three kinds of request: questions it can be asked; statements it can be told; and global checks it can perform. All share a common structure.

Structure of Requests

A request is specified in three parts: a named operation, its input arguments, and the form of the required output(s). The input or output types for the Terminology Server may be any of the forms:

- **Concept references** to entities in the server's own representation. References uniquely identify concept entities and can be either simple (e.g. a number) or compound expressions.
- **Linguistic expressions** in a supported natural language. Linguistic expressions do not, in general, identify internal concepts uniquely, and answers may include information on ambiguities which applications may either deal with themselves or treat as the basis for further questions to the terminology server. In the current implementation, linguistic expressions are only available as an output, but work on more than rudimentary natural language understanding for input is just beginning.
- **External expressions** such as codes from existing coding and classification systems, which have been mapped into the internal representation. An external expression can always identify a unique concept entity, but a concept entity may map to several codes in an external representation, possibly with further information indicating the differences between them, closeness of match, etc.

In general the Terminology Server will accept input objects and produce answers in any of these forms. Any necessary coercion between types is performed internally.

Internal Representation of concept entities: The CORE Model and GRAIL

Internally, the primary representation for concepts is the COncept REference (CORE) model, which is a compositional model expressed in the GALEN Representation and Integration Language (GRAIL) Kernel. The GRAIL Kernel is described in detail elsewhere[1, 7]. It is a strongly constrained compositional formalism. The model contains simple entities and the constraints which govern how these entities can be combined. Hence an indefinitely large number of concepts can be represented using a compact and efficient model. Because of the strong constraints, it is possible to verify whether proposed composite entities are 'sensible' and to generate lists of all 'sensible' compositions involving any given concept entity.

Requests: Things the Terminology Server can be Asked.

All these produce no change in the conceptual knowledge contained within the CORE Model.

What does this reference or expression mean?

- Is this a legal expression, and what is its simplest corresponding concept entity in the CORE Model (e.g. with any redundancies removed)?
- If the expression is legal, how is the corresponding concept entity classified— what more general concept entities subsume it? What more specialised concept entities does it subsume?
- What is known about this concept entity from the CORE Model? What other extrinsic information has been said about this concept entity?

<hasSpecificLocation UrinaryBladder>

This it returns as a simple reference and a French natural language phrase to display to the user. A specific lexical entry is found by the Multilingual Module and hence the Server returns 'cystite'. The user goes on to describe the cystitis in more detail by choosing to say it is *acute*. The Server thus produces

InflammatoryProcess which *<hasSpecificLocation UrinaryBladder hasChronicity acute>*.

This is again returned by the Server as a concept reference and a French phrase 'cystite aiguë'. The application then asks for a corresponding expression in SNOMED and receives via the Code Conversion Module 'D7-21110'. Throughout the interaction the application has simply received sets of pointers, and the user has read natural language phrases.

Requests: Things the Terminology Server can be Told

One of GALEN's major goals it to support local extensions and flexible development within an overall coherent framework provided by the CORE Model. Local sites and applications must therefore be able to add information to the Terminology Server in a number of different ways.

To extend the existing model

- Give new local names to existing or potential concept entities. Adding local names does not increase the range of things which can be expressed by the model, but it can make the model easier to use by simplifying what would otherwise be complex expressions.
- Add new primitive concept entities. The range of primitive concept entities may not include things which are important locally. For example, a surgical system might not include names for all of the surgical instruments used at a particular site.
- Add new statements so that existing attributes and concept entities can be used in new ways. It is often the case that the sanctions in the CORE model are too specific for local use and may have to be extended.
- Add new attributes and associated sanctions so that new things can be said. The range of attributes may not support sufficient detail for local use.

In general, additions of fine detail can performed locally, although central communication allows coordination of the model

To add to the other information sets supported.

- Add or modify the linguistic information
- Add or modify the mappings to an external representation such as a coding system or database schema.
- Add or change editorial information about items
- Add or modify the additional extrinsic information the model.

Information can be added to or modified in any of the data sets related to the linguistic or external representations. It also provides for maintaining editorial and version information about all items stored. If some datasets require special operations — e.g. to cope with special features of coding systems such as the dagger-asterisk structure in ICD-9 — these can be packaged in a standard form and exported through the Server's applications programming interface. In addition, the Server allows applications to store closely related information attached to the terminology structure, e.g. drug information, diagnostic criteria, and triggers to decision support.

Requests: Global Operations on the Model

- Coherence checking
- Managing updates
- Generation of local coding schemes

The Server provides a range of tests on the overall coherence of the model and a range of facilities for managing version control and updates are under development. One important function to potential users is the ability to 'compile out' fixed special purpose coding systems from the CORE model. These are simple coding systems for special purposes, and only contain a subset of the possible concepts and relationships within the CORE model. Nonetheless they are guaranteed to be coherent with the overall system and interchangeable with others using the Terminology Server or systems based on it.

ARCHITECTURE

Internally, the overall task has been modularised into different aspects - conceptual, linguistic, coding, and extrinsic - which are implemented by separate modules within the Terminology Server (figure 1). The Terminology Server combines these modules, adds reference and coercion mechanisms, and exports it's services via the API, to applications. The Terminology Server's reference management makes it easy for external applications to manipulate and store concept entities, for example as part of a patient record system. The Terminology Server's coercion mechanism provides efficient ways of combining multiple module services and relieves applications of needing to know how specific requests are handled. A flexible interface has been developed so that individual modules may 'export' their services, via the API, to external applications, so additional functionality can be made available very quickly.

The 'meaning' of a concept reference is its complete expansion and classification as a concept entity in the CORE Model, including all of the essential characteristics which can be inferred from its definition.

What can be said about this concept entity?

- What statements can sensibly be made about this concept entity? What are its sensible modifiers and relations?
- How can this concept be specialised according to given criteria? — *e.g.* anatomically, functionally, according to clinical indications or effects.
- What are the 'sensible' ways in which this set of concepts can be combined into a single larger concept?

A major function of the Terminology Server is to tell applications what further can be sensibly said about a concept entity — to support a user interface to help clinicians enter the information; to assist a bibliography system refine a query; or to assist a natural language system to disambiguate candidate phrases.

What are the nearest representations to this concept entity in some other representation?

- What are the expressions for this concept entity in a particular external system such as ICD-9? What is the preferred term for this concept entity in that system? If no exact match is possible provide the 'best' matches according to given criteria and supplementary information to assist the application in choosing among them. What information is lost or added in the conversion?
- What are the natural language expressions for this concept in a particular language? What is the preferred form for a particular 'clinical linguistic group'.
- Are these two concept entities derived from two different external representations the same? If not, how do they differ? What information would have to be added or removed from each to make them the same?
- Find all of the expressions in a given external representation which correspond to children of this concept entity, i.e. all of the codes which this concept entity subsumes. This is a particularly important question for information retrieval. It allows the Terminology Server to compensate for the deficiencies in the organisation of external coding systems. For example, forms of heart disease are found in at least five different chapters of ICD-9.

Answers to these questions provide a comprehensive service for translating between the concepts used by various representations — coding systems, database schemata, internal representations of knowledge based systems, simple linguistic 'rubrics', etc.

What is the encapsulated form of this expression?

- Provide a volatile or persistent fixed length reference for this concept entity.
- Encapsulate these concept entities according to a given format for an application as a set of references or a set of external expressions.

These functions provide fixed length 'handles' to complex concepts which can be easily manipulated by applications and relational databases. The application is protected from the problems of potentially unbounded compositional structures.

What other extrinsic information has been attached to this concept besides the indefeasible terminological knowledge?

- Find the most specific information in a given category about a concept entity.
- Find all of the information in a certain category about a concept entity and all of its parents.
- Find all the children of a particular concept entity such that a particular piece of extrinsic information holds.

Strictly speaking, the CORE Model contains only concrete conceptual knowledge which is indefeasible and true 'by definition'. However, a major function of the CORE model is to provide a framework with which to organise other, more general information - *e.g.* concerning drug interactions, clinical procedures, or diagnostic methods. Holding such information and using the conceptual framework to retrieve the most specific information in a certain category is so useful that additional operations are provided to support these functions directly. However the Terminology Server makes no attempt itself to reason with this additional information.

An example

A simple example will illustrate the flow of requests to the Server. The Terminology Server is not itself an end user application but assume a clinical data entry application holds a reference to the concept *UrinaryBladder* as a starting point. During the course of its dialogue with the user the application asks the Server for those attributes concerned with locations that can sensibly apply to this concept, and the possible values for these. One of the replies is

isSpecificLocationOf-InflammatoryProcess.

This is selected by the user who asks to go on and describe the inflammation in more detail. The Server is thus asked to combine the components to give the concept

InflammatoryProcess which

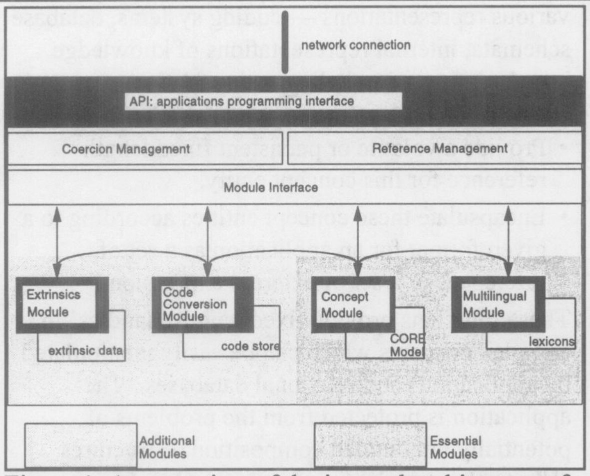

Figure 1: An overview of the internal architecture of the Terminology Server.

The central task of concept modelling is addressed by the 'Concept Module' which interprets the 'Concept Reference' (CORE) Model The CORE Model serves as an *inter lingua* amongst medical nomenclatures, vocabularies, and the terminological aspects of database schemata.

The Multilingual Module provides lexicons and grammatical information for expressing, and eventually understanding, phrases in natural languages. External representations can depend on the Multilingual module to translate the CORE Model expansion of their representation, in addition to any 'official' translation.

The Code Conversion Module maintains the external representations, along with special information related to their structure and browsing, *e.g.* information on the cross referencing in SNOMED or the dagger-asterisk mechanism and exclusions in ICD-9/10. The Code Conversion Module also provides the functionality concerned with resolving ambiguities and conflicts when there is not an immediate one-to-one correspondence between the GALEN CORE Model and the target external representation, for example when the expansion of a term from one external coding system has no direct representation in a different external coding system.

The Extrinsic Information Module provides a repository in which applications or sites can store detailed information about the clinical criteria for using concepts in the Terminology Server. These definitions are appended to the classification structure of the CORE Model but are not part of it.

CURRENT STATUS

All the modules described above have been implemented using Parc Place Smalltalk™ (VisualWorks™) with Sybase™ for database support. Client ends are available in Smalltalk™ and C. The Server is currently being used within the GALEN project to support clinical user interfaces for test ordering in nosocomial infections and entry of arthroscopic findings. Applications to support a knowledge editor for Medical Logic Modules and a Classification Manager for assisting in the development of coding and classification systems are under way. The architecture and interface language have been demonstrated including the ability to operate multiple client application and a server on separate machines linked either by local area networks or across the Internet.

The current applications have allowed an analysis of the functions required which has so far proved robust. The modular architecture provides a smooth means of expansion. Full evaluation must await maturation of the applications, and above all of the demonstration of multiple coherent applications sharing the same server. Even at this early stage the Terminology Server provides a vision of new possibilities - of a shift in emphasis from providing terminology *per se* to providing terminological services.

REFERENCES

1. Rector A, Nowlan W, Glowinski A. Goals for concept representation in the GALEN project. 17th Annual Symposium on Computer Applications in Medical Care (SCAMC-93). McGraw Hill, 1993: 414-18.
2. Rector A, Nowlan W, Kay S. Foundations for an electronic medical record. Meth Inform Med 1991;30:179-86.
3. Rector A, Nowlan W, Kay S. Conceptual knowledge: the core of medical information systems. In: Lun K, Degoulet P, Pierre T, Rienhoff O, (ed). Seventh World Congress on Medical Informatics, MEDINFO-92. Geneva: North-Holland Publishers, 1992: 1420-1426.
4. Evans DA, Cimino J, Hersh WR, Huff SM, Bell DS. Toward a medical concept representation language. J Am Med Informatics Assoc 1994, 1 (3) 207-217.
5. Masarie Jr F, Miller R, Bouhaddou O, Giuse N, Warner H. An interlingua for electronic interchange of medical information: using frames to map between clinical vocabularies. Comp Biomed Res 1991;24(4):379-400.
6. Cimino JC, Hripscak G, Johnson S. Knowledge-based approaches to the maintenance of a large controlled medical terminology. J Am Med Informatics Assoc 1994;1(1):35-50.
7. Rector A, Nowlan W. The GALEN Representation and Integration Language (GRAIL) Kernel. GALEN deliverable 6. (Medical Informatics Group, University of Manchester), 1993.