ISSUES IN ONTOLOGY DESIGN FOR A CLINICAL DECISION SUPPORT SYSTEM

CHRISTOPHER S.G. KHOO & JIN-CHEON NA
Division of Information Studies
Wee Kim Wee School of Communication & Information
Nanyang Technological University
31 Nanyang Link, Singapore 637718
assgkhoo@ntu.edu.sg; tjcna@ntu.edu.sg

Introduction. Proposes that ontology design and development will be a major area for LIS theory and practice in the future. Summarizes the differences between ontologies and thesauri. Examines issues in the design and development of an ontology for decision support, and proposes some procedures and guidelines.
Method. Describes a case study of developing an ontology to represent the knowledge base for a clinical decision support system for wound management at a hospital.
Results. Classes in the ontology can be divided into medically related classes, decision-making classes and general utility classes. Guidelines adopted in the project include: preferring generic and reusable classes and relations, distinguishing between relations from a concept to different target concepts, specifying data entry form fields as relations and attributes, selecting a few important concepts to model as complex classes, and finally preferring a representation that is easier for the designer and user to understand and maintain.
Conclusion. Many of the issues in designing an ontology relate to the design of the relation types and the trade-off between designing more specialized relations or more specialized concepts. Choices also have to be made between simpler and easy to understand representations, or more complex, detailed representations that are more portable.

Introduction

Information and knowledge organization is one of the core areas of the library and information profession. Information/knowledge organization theory and practice has expanded in scope and variety in the past 1½ decades with the growth of knowledge management in organizations, information architecture for websites and portals, and Internet and Web applications including the semantic web. Various information/knowledge representation and encoding schemes have been developed including metadata schemas (e.g. Dublin Core), taxonomies, Resource Description Framework (RDF), and XML encoding. Librarians and information professionals are gradually getting involved in these expanded areas of information/knowledge organization, in addition to the traditional areas of thesauri, classification schemes and cataloging. Information professionals are developing taxonomies to support the information architecture of websites and enterprise portals, and metadata schemas for digital library applications.

An emerging area of knowledge organization for information professionals is in ontology design and construction. We forsee this to be a major area of work for information professionals in the future, with the growth of the semantic web and ubiquitous computing, and the push towards more intelligent systems. Today we are called catalogers, tomorrow we might be called ontologists or knowledge professionals.

This paper reports on an initial effort to develop an ontology as a knowledge base for a clinical decision support system at a hospital in Singapore. Wound management was chosen as the initial domain for the decision support system. We examine the issues faced in developing the ontology, and outline the procedure and process for designing and constructing it, compared to developing taxonomies, thesauri and classification schemes.

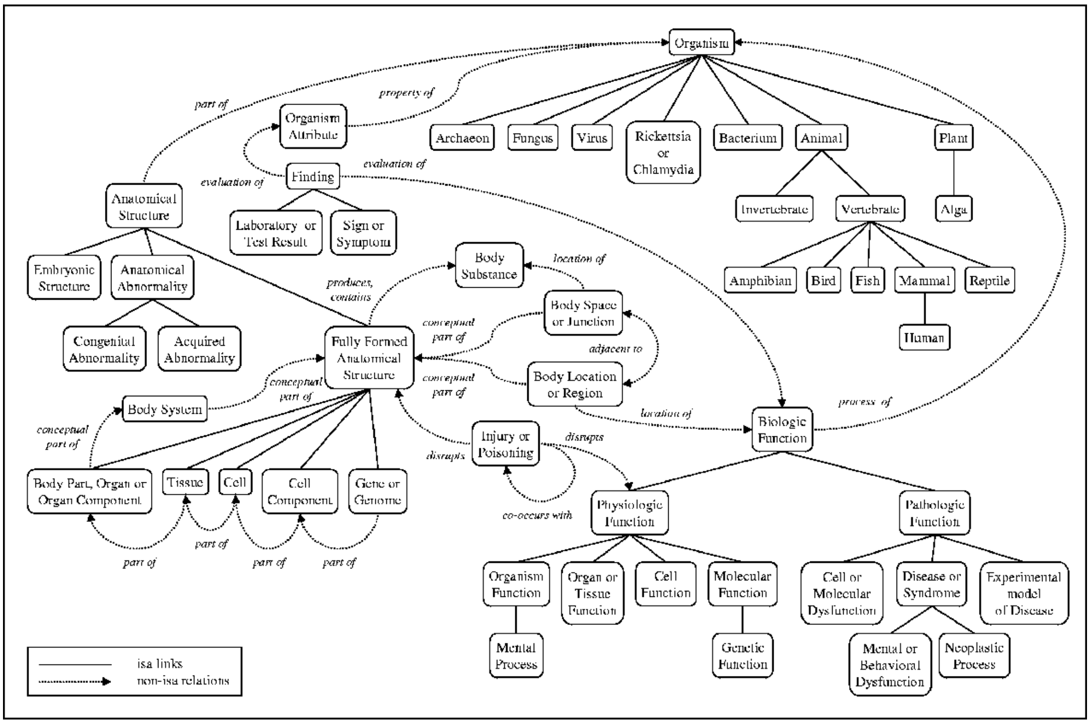

Fig. 1. UMLS upper-level ontology
(source: http://www.nlm.nih.gov/research/umls/META3_Figure_3.html)

What is an ontology?

Many definitions of *ontology* have been proposed, and papers have analyzed various aspects of the concept of ontology. This definition by Berners-Lee, Hendler & Lassila (2001) referred to the origin of the concept in philosophy and its adoption in artificial intelligence and semantic web:

> *In philosophy, an ontology is a theory about the nature of existence, of what types of things exist; ontology as a discipline studies such theories. Artificial-intelligence and Web researchers have co-opted the term for their own jargon, and for them an ontology is a document or file that formally defines the relations among terms. The most typical kind of ontology for the Web has a taxonomy and a set of inference rules.* (p. 40)

Perhaps the most often quoted definition is by Gruber (1993):

> *A body of formally represented knowledge is based on a conceptualization: the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them (Genesereth & Nilsson, 1987). A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose. ... An ontology is an explicit specification of a conceptualization.* (p. 199)

Guarino & Giaretta (1995), Guarino (1997) and Gómez-Pérez, Fernandez-Lopez & Corcho (2004) analyzed various definitions of ontology. A collection of definitions can be found at http://www.aaai.org/AITopics/html/ontol.html.

In practical terms, an ontology is a set of concepts linked by a set of semantic relations, sometimes complemented with a set of axioms or inference rules. Ontologies come in many types and flavors, depending on the application area and intended use, representation scheme, philosophical underpinnings, and the construction method. Those functioning as online search aids are more lexical in nature focusing on terms, have mostly hierarchical relations, and no axioms. Others supporting inferencing may be represented in a logic representation and have many axioms. Gómez-Pérez, Fernandez-Lopez & Corcho (2004) outlined the different typologies of ontologies that have been put forward by various authors, and said that even thesauri can be considered light-weight ontologies. One major difference between an ontology and a thesaurus or classification scheme, however, is the richer set of relations used in an ontology.

The most important relation is the *is-a* relation (also called *a-kind-of, superclass-subclass* or *genus-species* relation). This relation is used to link concepts together to form a taxonomy or hierarchy of concepts. For example, a collie *is-a* dog, a dog *is-a* mammal, a mammal *is-a* vertebrate, and a vertebrate *is-a* animal. In addition, other relations are explicitly defined to link concepts and taxonomies together. The set of relations is also organized into a hierarchy using the *sub-relation* relation. This is a special relation to relate different relations—a meta-relation if you will. Other meta-relations include *inverse-relation* and *same-relation*. Concepts can also have attributes, which are a kind of relation that links concepts to datatype values such as integers (e.g. age), real numbers (e.g. temperature), text strings (e.g. name), and dates (e.g. date of birth).

Fig. 1 shows the UMLS (Unified Medical Language System) upper-level ontology (http://www.nlm.nih.gov/research/umls/) as an illustration of an ontology. The *Organism* taxonomy is linked to the *Anatomical Structure* taxonomy by the *part of* relation, to the *Biologic Function* taxonomy by the *process of* relation. Notice that all the relations have a direction, indicated by an arrow head.

The *is-a* relation has some special properties. It is a transitive relation, meaning that if A *is-a* B and B *is-a* C, then A *is-a* C. Thus, since a collie *is-a* dog and a dog *is-a* mammal, then a collie *is-a* mammal. In addition, the *is-a* relation has the property of inheritance. A relation between two higher-level concepts is inherited by their descendants in the taxonomy. In Fig. 1, from the relation *Anatomical Structure* is *part of* an *Organism*, we can infer that *Congenital Abnormality* is *part of* a *Human*, since *Congenital Abnormality* is a descendant of *Anatomical Structure* and *Human* is a descendant of *Organism*.

In an ontology, instances are distinguished from concepts. Instances are entities or things that are assigned to concepts (denoting categories). Instances are often physical objects. For example, Snoopy is an instance of a dog. So an ontology can be seen as an abstract knowledge structure or schema (like a database schema) that is used to manage instances or things. However, the conceptual distinction between instances and concepts is fuzzy. What is considered an instance depends on the application domain. In a library loan application, the instances might be the physical copies of books identified by their barcode. In a bookstore system, the instances might be the book titles since the bookstore is not interested in tracking the individual copies of books—only the number of copies sold.

Some ontologies include IF-THEN rules to represent more complicated relations and to support reasoning and inferencing. For example,

IF A is-an animal and A *has-num-legs* 4, THEN A is-a mammal

An XML language called the Web Ontology Language (OWL) has been developed by the World Wide Web Consortium (2004) to encode or represent an ontology in XML format. In this study, we adopted OWL level 2 and F-logic as the encoding schemes. Since OWL does not represent rules, the F-logic language is used to represent them.

OWL level 2 imposes some constraints on the ontology design to make it easier for machines to process the ontology and perform inferencing. It imposes a strict separation of concepts (called classes) and relations (called properties). So a relation is not considered a class. Another constraint is the strict separation between instances and concepts. An instance is not allowed to be a concept as well. So instances are the bottom level of the ontology. Some thing cannot be an instance of another instance.

An ontology is different from classification schemes and thesauri in the following ways:

- In an ontology, the relations between concepts are specified. The relations are merely implied in a classification scheme. Thesaurus relations are limited to a small number of broad relations—broader-narrower term, related term, etc.
- An ontology focuses on concepts, whereas a thesaurus focuses on terms.
- Classification schemes and thesauri are user-oriented in the sense that they are designed to support user browsing and searching. Ontologies are meant to be processed by machines, which is why the relations and their properties (e.g. inheritance and transitivity) have to be explicitly specified.
- An ontology can contain IF-THEN rules to support inferencing.

An ontology is useful for representing knowledge in a complex domain to a level of rigor and consistency that allows intelligent systems to perform reasoning and inferencing accurately. Ontologies can be classified by their breadth of coverage. General purpose ontologies, such as CYC (http://www.cyc.com) and DOLCE (http://www.loa-cnr.it/DOLCE.html), have very abstract classes at the upper level which are derived from philosophical considerations. Industry-wide ontologies (e.g. travel ontology) is developed to support e-commerce and business-to-business communication and transactions. Application level ontologies are abstract ontologies designed to support a particular kind of application.

Our study has a narrow scope—to develop an ontology for a clinical decision support system for wound management in a hospital. However, it is hoped that parts of the ontology would be portable to other kinds of decision support systems.

Ontology-Based Decision Support System

The overall objective of the project is to develop a clinical decision-support system to assist nurses at a hospital in wound management. The system is meant to improve the quality of healthcare by:

- helping nurses make more accurate assessments and diagnosis.
- reminding them of the recommended procedures and treatments.
- alerting them to additional factors to consider in selecting procedures and treatments, e.g. cost, availability of drugs in the formulary, patient conditions and factors to consider.
- alerting them to alternative or new treatments/drugs.

However, it is ultimately the health professional's responsibility to choose an appropriate, effective and cost-effective treatment based on available information.

The system design and development is divided into 4 parts:

1. Knowledge acquisition and representation
2. Knowledge-based system, including inference engine
3. Image processing and retrieval
4. User interface, user studies and usability evaluation.

The focus of this paper is on knowledge representation and knowledge base design in part 1 of the system development. For the knowledge base, it was decided to use multiple storage and representation formats for the different types of knowledge—to take advantage of the strengths of the different formats and to reduce the complexity of the ontology. For example, detailed descriptions of procedures and treatments are stored either as documents or as relational database records. The ontology can focus on the knowledge needed for reasoning and making recommendations of treatments for different situations.

The technologies used to represent the knowledge base are:

- a relational database to store structured data containing details of treatments, drugs and dressings, as well as patient historical information
- a document management system to manage a repository of documents in various formats
- an image retrieval system to manage a repository of images
- an ontology for conceptual information needed for inferencing.

The heart of the knowledge base is the ontology which drives the reasoning, and determines what recommendations are presented to the user. An ontology server, Ontobroker provided by Ontoprise GmbH (http://www.ontoprise.de/de/en/home/products/ontobroker.html), was selected as the inference engine.

Ontology Development Process

The ontology development encompasses the following stages:

1. Determine the objectives and requirements of the ontology.
2. Identify the knowledge sources, and types of knowledge needed.
3. Analyze the environment and system in which the ontology will be deployed.
4. Carry out a user study to obtain information about
   a. user tasks to be supported by the system
   b. users' current decision making process
   c. users' knowledge
   d. current problems faced by users
5. Decide which types of knowledge will be stored as relational database records, documents, images or in the ontology.
6. Analyze the knowledge sources for concepts/terms, relations and rules.

7. Specify classes (i.e. concepts) and construct taxonomies of classes (i.e. concept hierarchies).
8. Specify relations between classes (especially across taxonomies). Construct relation hierarchies.
9. Construct IF-THEN rules
10. Evaluate, test and refine the ontology.

Though the stages are carried out in the order listed, a lot of backtracking is involved. For example, when constructing the IF-THEN rules, it may be necessary to backtrack to adjust the classes, taxonomies and relations.

Because the knowledge base is meant to support user tasks and user decision making, it is important to carry out user studies to find out more about the user tasks, the users' current decision making process and the relevant knowledge that nurses have. In this project, the user study was delayed because of the ill health of a project member, and thus detailed end-user input has not been taken into account in the ontology development. The requirements for the system were identified from documents and discussion with three representative nurses from the hospital in the project team.

Knowledge to be Represented

The knowledge to be represented in the ontology includes

- formal knowledge found in textbooks and clinical practice guidelines;
- informal (tacit) knowledge of expert wound nurses.

Formal domain knowledge given in textbooks and clinical practice guidelines issued by health departments, medical associations and hospitals are important to provide the main concepts and knowledge structures of the domain. It equips the knowledge professional with the basic knowledge and vocabulary needed to interview the domain experts for more in-depth knowledge.

The main knowledge sources used in developing the ontology:

- Wound management textbooks and review articles, e.g. *Wound Care Nursing* (Bale & Jones, 2006)
- Guidelines from health/medical organizations in other countries, e.g. *NHSSB Wound Management Manual* (Northern Health and Social Services Board, 2005)
- Guidelines from the country's Ministry of Health
  - Ministry of Health Nursing Clinical Practice Guidelines for *Prediction and Prevention of Pressure Ulcers in Adults*
  - Ministry of Health Nursing Clinical Practice Guidelines for *Nursing Management of Pressure Ulcers in Adults*
- The hospital's own wound treatment guidelines
- Recording forms used by the hospital to record information about the patient's condition, e.g. the wound chart for recording details of wounds and the Braden scale instrument for assessing the risk of pressure ulcer
- Experienced nurses and expert wound nurse (domain experts).

From the literature, it was gathered that the treatment decisions should take into consideration:

- The nature and features of the wound
- The patient's overall condition, including cause of wound and major health issues (e.g. diabetes)
- The potential efficacy of treatment, including possible interactions with other treatments
- The availability and cost of treatment.

The ontology should be compact and targeted at supporting clinical decision making. It should not be bloated with large amounts of theoretical knowledge and experimental information from the research literature.

Ontology Design Guidelines

The classes created in the ontology can be divided into the following groups:

- Medically related classes (domain-related concepts)—classes from the domain of medicine, wounds and wound management
- Decision-making classes—classes that focus on supporting decision-making functions

- General utility classes—low-level fundamental classes of a general nature, e.g. time and units of measure.

In developing an ontology, choices have to be made from alternative representations. A major consideration in ontology design is ease of understanding and maintainability. An ontology for a realistic application domain is likely to be quite complex and difficult to grasp and maintain. However, the knowledge professional needs to be able to examine and validate the ontology, and check that inferences draw from it will be sound. It should also be easy to map the ontology into a readable format so that domain experts can check and approve the rules. This is crucial in a clinical setting where wrong recommendations can cause lives and result in lawsuits. Thus an important design consideration is intuitiveness and ease of understanding.

Since most of the knowledge in the ontology comes from the literature (literary warrant) and domain experts (user warrant), the terms used and the structure of the ontology should be intuitive to the domain experts, and easy to map to and from the users' mental models. This will make it easier for the knowledge professional to convert the ontology into a format that allows the domain experts to check and endorse the ontology. It will also make it easier for the knowledge professional to convert the information provided by the domain experts into the ontological structures.

Since relations are more detailed in ontologies, compared to thesauri and classification schemes, many of the design issues in ontologies pertain to the specification of relations and attributes, and whether to have more types of relations/attributes or fewer. The choices include:

- Whether to represent something as a class or relation.
- Whether to have more detailed relations or more detailed classes—whether to specialize a class into more detailed subclasses (creating a class hierarchy), or to specialize a relation into subrelations (relation hierarchy).
- Whether to use class attribute values to specialize a class into subclasses. The specialized classes will then incorporate the attribute values.
- Whether to decompose a class (or relation) into its components (thus creating a complex class), or to represent it as a unitary simple class (or relation).

In addition, deciding whether to represent some thing as a subclass or an instance of an existing class can be tricky. Some of these issues will be discussed later.

We adopted the following principles to guide us in making decisions on alternative representations:

- Prefer reusable classes and reusable relations, i.e. construct generic classes and relations.
- Distribute the complexities among the classes and relations (so that the ontology is easier to read and understand). Don't create too detailed classes or too detailed relations.
- Distinguish between relations emanating from a class to different target classes.
- Select the representation that more closely reflects the domain experts' (user) perspective. For example, when deciding whether to specialize relations or classes, consider whether the users perceive the concepts as classes or relations. Usually, a hierarchy of classes is easier for the user to grasp than a hierarchy of relations.
- Specify data entry form fields as relations and attributes.
- Identify a few important or central concepts in the application domain, and design them as complex classes (with component classes).

Some Design Issues

Designing Relations and Attributes

A major consideration in designing an ontology is how detailed and specialized the relations will be—whether to use a small number of abstract relations or a large number of specialized relations organized in a relation hierarchy.

In many situations, the relation type can be inferred from the two classes that are related, for example:

Patient ->(relation)-> Cancer   implying the relation of *has disease*
Patient ->(relation)-> Dehydrated   implying *has nutritional state*

Patient ->(relation)-> Male implying *has gender*

In other situations, the relation type is ambiguous, for example:

Patient ->(relation)-> Drug	prescribed drug, recommended drug, or poison accidentally taken?
Patient ->(relation)-> Ankle	a disease of the ankle?
Patient ->(relation)-> Moderate amount	amount of which condition?

In this study, the majority of the relations represent input data to be entered by the nurses into the system. Many of the relations are transcribed from current recording/diagnostic forms used, e.g. the Braden scale used to assess the patient's risk of pressure ulcer, and the wound chart used to record detailed conditions of each wound. The attribute-value pairs taken from each field in the Braden scale and wound chart are represented as attributes and relations. This approach helps us to manage and identify all the input data.

We also specify different relations to distinguish between the associations from a particular class (e.g. Patient) to different target classes (i.e. Disease, Nutritional state, Gender). So the Patient class has the following relations:

Patient ->(has_disease)-> Cancer
Patient ->(has_nutritional_state)-> Dehydrated
Patient ->(has_gender)-> Male
Patient ->(recommended_procedure)-> Drug

Many of the relations are subrelations of *has*. Examples of the *has* sub-relations used in the ontology are given in Table 1.

Sometimes by specializing the relation, the target class can be left more general and reusable. Contrast the following examples:

1. Wound ->(has_exudate_level)-> High
2. Wound ->(has)-> High_exudate
3. Wound ->(has)-> Exudate ->(amount)-> Large

Version 1 might correspond to an entry in the wound chart (exudate level), whereas version 2 contains a commonly used term and concept ("high exudate"). In this case, we opted to use both versions 1 and 2. Version 1 is used for storing data from the input form. An inference rule then converts the information to version 2, which is used for inferring the recommended treatment.

Table 1. Selected subrelations and subattributes of *has*

has
- has authority
- has characteristic
    - has odour
    - has exudate
        - has bleeding
        - has serous_exudate
        - has purulent exudate
- has condition
    - has disease
    - has nutritional state
- has document
- has procedure
- has secondary procedure
- has instrument
- has location
- has gender

Divide a Class into Subclasses Incorporating Attribute Values

A class is sometimes divided into subclasses based on the values of a particular attribute. Thus, the knowledge professional sometimes has to decide whether to subdivide a class, or to represent this attribute explicitly as a relation/attribute.

Thus the *Pressure ulcer* class can be subdivided into subclasses:

Pressure ulcer

- Pressure ulcer stage 1
- Pressure ulcer stage 2
- Pressure ulcer stage 3

Or, the stages can be represented as separate concepts linked by a relation to the *Pressure ulcer* class:

Pressure ulcer ->(has_stage)-> Stage 1

In this case, we decide to follow user warrant, i.e. to create subclasses if the users use the specialized terms and view the concepts as subclasses.

Class versus Instance

It is sometimes difficult to decide whether a concept should be represented as a suclass or an instance of an existing class. Much depends on the application. If a concept is represented as a class, an instance of the class has to be created later in order to make use of the ontology for reasoning, for example:

Person:tom123 ->(has_disease)-> Wound:wound123 –
(has_characteristic)-> High_exudate:exudate123
(has location)-> Left_foot:foot123

An instance has to be created for every class that is involved in representing the situation (i.e. disease, its conditions, the patient, the treatment, etc.). This can be tedious if many concepts are involved in a situation. If it is not necessary to distinguish between different instances of a class or if the instance does not have any attributes or relations to other classes, then the class can be converted to an instance. In the above example, *Wound* (and other diseases) have to be represented as a class because every patient's disease is different and has different characteristics. However, "High exudate" and "Left foot" can be handled as instances (unless it is necessary to specify the attributes of the left foot).

In the wound management ontology, types of dressings and drugs are handled as instances. However, they can be related to instances of products (i.e. product names).

Representing Measures

Representation of quantified measures in an ontology is quite complicated. For example, the fact that "a wound has 30% granulation" can be represented as

Wound:* ->(has_characteristic)-> Granulation:* ->(quantity)-> * -
(value)-> 30
(measure)-> percent

The asterisk * represents an anonymous instance of a class. So "Wound:*" refers to a particular instance of wound. This representation is difficult to read. So we opted for the following simpler representation:

Wound:* ->(has_granulation_percent)-> 30

This representation makes use of a complex (specialized) relation that incorporates the concepts of *granulation* and *percent* measure.

Complex Classes Versus Simple Classes

Complex classes refer to concepts/situations that are modeled as a cluster of component classes. Thus complex classes are analyzed into lower-level or more basic component classes. In designing an ontology, one has to choose between representing a concept as a simple unitary class or as a complex class linked to a cluster of basic classes.

The complex representation is more powerful (i.e. can represent subtle variations of the situation) and flexible in that it can be adapted to other situations, applications and domains. On the other hand, complex representations are more difficult to understand and use, and need careful documentation. In this study, we select only the more important concepts/situations in the application to represent in detail as complex classes. The concepts selected to be represented as complex classes are *Person-Patient-Disease* and *Action.*

The *Person-Patient-Disease* situation is of course a central concern in the application. A *Person* has demographic attributes such as name, date of birth and gender. A *Person* may have several visits or stays at the hospital, and each of these visits have different attributes. We model these different visits as different *Patients*. There is a one-to-many relation from *Person* to *Patient*. A *Patient* can have many attributes. The basic ones that apply to all patients include case number (patient number), admission class, ward number and bed number.

A *Patient* can have one or more *Diseases*, such as a wound or pressure ulcer. However, patients with the same disease will have different disease conditions and attributes. Thus an instance of a disease related to a particular patient will have several attributes. The attributes depend on the type of disease.

If it is necessary to track the course of a disease (to represent the states of the disease over time), then another class *Disease_state* is needed. A *Disease* instance will have one or more *Disease_states* which naturally will have a date-time attribute. In this study, it was decided to store historical information of a patient's condition in a relational database to keep the ontology simpler. The ontology will contain only information necessary to recommend actions or treatments. The cluster of classes used to model patients and their diseases are thus:

Person ->(has_instance)-> Patient ->(has_disease)-> Disease

The *Action* class is used to specify recommendations for a particular disease condition:

Disease –>(recommended_action)-> Action –
(has_procedure)-> Procedure
(at_time_interval)-> Time_interval

The *Action* class has the following subclasses to indicate different kinds of actions:

- periodic_action
- treatment
- review.

Periodic_action is used to specify regular actions at specific time intervals. The *Action* class is complex because it may need to specify not just the *Procedure* (which may be a drug) but also the dosage and time interval to administer the procedure. *Procedure* has subclasses:

- Assessment
- Treatment_method
  - Surgery
  - Wound_treatment_method
    - Dressing

Procedure thus specifies the concrete action or treatment to administer.

Constructing Inference Rules

Most of the inference rules involve some kind of categorization, generalization or translation from one representation to another. The rules constructed for the wound management ontology can be divided into the following types:

- Rules for diagnosis (e.g. inferring a condition from a test score)
- Rules for scale conversion
- Rules for translation from one representation to another
- Rules used for recommending an action (e.g. treatment)
- Rules for supporting decision making functions, e.g. generating alerts

In hospitals, the Braden Scale is a questionnaire instrument used to assess a patient's risk of developing pressure ulcers. The risk of pressure ulcer development is classified according to the Braden scores:

- 5-11: High Risk
- 12-15: At Risk
- 16-23: Low Risk

Thus, the ontology has rules to map Braden scores to the degree of pressure ulcer risk, such as:

IF a Patient *has_braden_score* <= 11
THEN the Patient *has_condition* High_risk_of_pressure_ulcer

Categorization rules are also used to convert a measure from one scale to another (e.g. from a numeric scale to a categorical scale), e.g.:

IF a Wound *has_granulation_percentage* >= 20
THEN the Wound *has_characteristic* Granulation

The following is an example of rules that recommend an action or treatment:

IF a Wound *has_characteristic* Granulation
THEN the Wound *has_recommended_procedure* [Hydrocolloid AND Low-adherent_dressing]

Conclusion

Many of the issues in designing an ontology relate to the design of the relation types and the trade-off between designing more specialized relations or more specialized concepts. Choices also have to be made between simpler and easy to understand representations, or more complex and detailed representations.

We have suggested some guidelines for making decisions between alternative representations. However, different guidelines sometimes suggest different alternatives. In this study, we have preferred representations that are easier to understand both by the designers and the users, in order to make the ontology easier to validate and maintain. This often means that multiple concepts and relations are bundled into a specialized concept or relation that the user is familiar with. However, this decision is taken at the expense of portability and flexibility. Using more generic concepts and relations will result in bigger, more complex and detailed graphs that are harder to understand, but easier to adapt to new situations and domains. In this study we selected only a small number of important concepts to represent as complex and detailed graphs.

The representations that are selected also have implications for the inference rules to be constructed, since the rules refer to concepts and relations in the ontology. We have sometimes adopted multiple representations but this necessitates having inference rules that translate between the different representations. More discussion of inference rules construction is left to a future paper.

References

Bale, S., & Jones, V. (2006). *Wound care nursing* (2nd ed.). Edinburgh: Mosby Elsevier.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001) The Semantic Web. *Scientific American, 284*(5/May), 34-43.

Genesereth, M.R., & Nilsson, N.J. (1987). *Logical foundations of artificial intelligence*. Morgan Kaufmann. [As cited in Gruber, 1993]

Gómez-Pérez, A., Fernandez-Lopez, M., & Corcho, O. (2004). *Ontological engineering*. London: Springer-Verlaq.

Gruber, T. R. (1993). A translation approach to portable ontology specification. *Knowledge Acquisition, 5*(2), 199-220.

Guarino, N. (1997). Understanding, building and using ontologies. *International Journal of Human-Computer Studies, 46*, 293-310.

Guarino, N., & Giaretta, P. (1995). Ontologies and knowledge bases towards a terminological clarification. In N. J. I. Mars (Ed.), *Towards very large knowledge bases: Knowledge building and knowledge sharing* (pp. 25-32). Amsterdam: IOS Press.

Northern Health and Social Services Board. (2005). *NHSSB wound management manual*. Retrieved October 1, 2008, from http://www.nhssb.n-i.nhs.uk/publications/primary_care/Wound_Manual.pdf

World Wide Web Consortium. (2004). *OWL Web Ontology Language: Overview* (W3C Recommendation 10 February 2004). Retrieved 1 June 2008 from http://www.w3.org/TR/2004/REC-owl-features-20040210/.