Chapter 2

ONTOLOGY DEVELOPMENT METHODOLOGIES

Here we analyze the state of the art of various existent methodologies that can be used for ontology construction process.

2.1 METHONTOLOGY

METHONTOLOGY [21] is a methodology that is based on software engineering and knowledge engineering. It is structured methodology used to build ontologies from scratch. It also gives a life cycle to build ontologies based on evolving prototypes. METHONTOLOGY is divided in to management activity, development activity. These activities are integrated to work together. The lifecycle model of METHONTOLOGY is waterfall but uses prototyping strategy. METHONTOLOGY consists of the following tasks:

2.1.1 Specification

The goal of the specification phase is to produce either an informal, semi-formal or formal ontology specification document written in natural language, using a set of intermediate representations or using competency questions, respectively.

2.1.2 Knowledge Acquisition

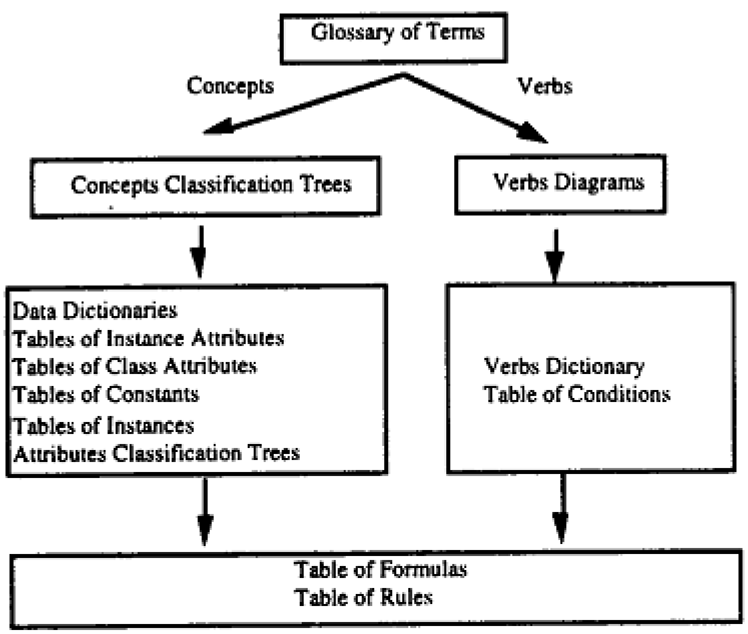
It is coincident with other activities. Most of the acquisition is done simultaneously with the requirements specification phase, and decreases as the ontology development process moves forward. Experts, books, handbooks, figures, tables and even other ontologies are sources of knowledge from which the knowledge can be elucidated using in conjunction techniques such as: brainstorming, interviews, formal and informal analysis of texts and knowledge acquisition tools.

2.1.3 Conceptualization

In this activity, you will structure the domain knowledge in a conceptual model that describes the problem and its solution in terms of the domain vocabulary identified in the ontology specification activity. The first thing to do is to build a complete Glossary of Terms (GT). Terms include concepts, instances, verbs and properties. So, the GT identifies and gathers all the useful and potentially usable domain knowledge and its meanings. Note that you do not start from scratch when you develop your GT.

For each set of related concepts and related verbs, a concepts classification tree and a verbs diagram is built. After they have been built, you can split your ontology development process into different, but related, teams. Figure 2 graphically summarizes the intermediate representations used in the conceptualization phase.

Fig. 2: Set of Intermediate Representations in the conceptualization phase.



2.1.4 Integration

With the goal of speeding up the construction of your ontology, you might consider reuse of definitions already built into other ontologies instead of starting from scratch.

2.1.5 Implementation

Ontologies implementation requires the use of an environment that supports the meta-ontology and ontologies selected at the integration phase. The result of this phase is the ontology codified in a formal language such as: CLASSIC, BACK, LOOM, Ontolingua, Prolog, C++ or in your favorite language.

2.1.6 Evaluation

Evaluation means to carry out a technical judgment of the ontologies, their software environment and documentation with respect to a frame of reference (here it the requirements specification document) during each phase and between phases of their life cycle. Evaluation subsumes the terms Verification and Validation.

The output proposed by METHONTOLOGY for this activity is many evaluation document in which the ontologist will describe how the ontology has beenevaluated, the techniques used, the kind of errors found in each activity, and the sources of knowledge used in the evaluation.

2.1.7 Documentation

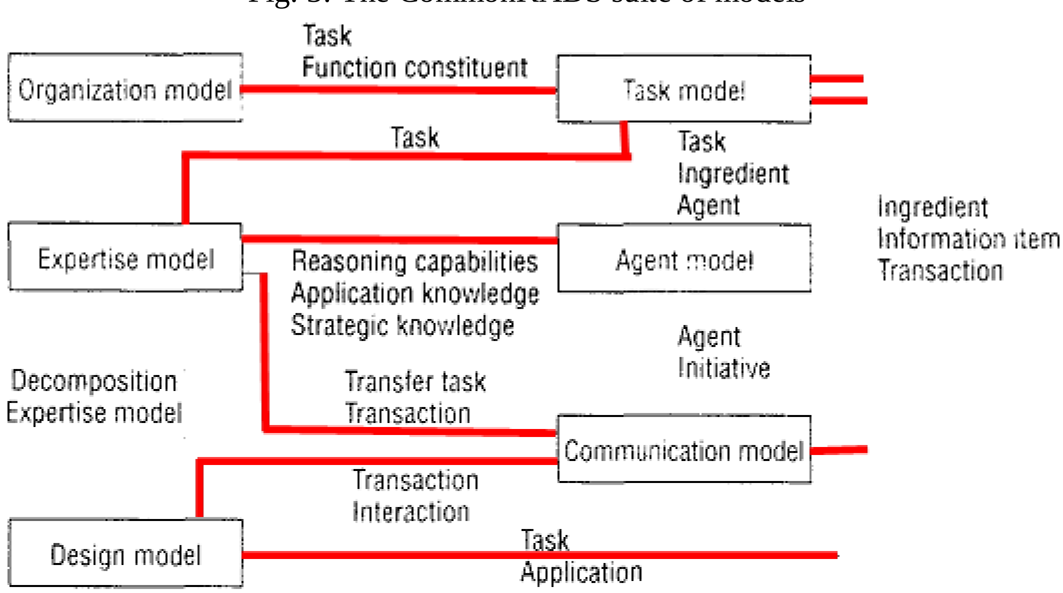
There are no standard guidelines on how to document ontologies. In many cases, the only documentation available is in the code of the ontology, the natural language text attached to formal definitions, and papers published.

METHONTOLOGY pretends to break this circle including the documentation as an activity to be done during the whole ontology development process. In fact, after the specification phase, you get a requirements specification document; after the knowledge acquisition phase, a knowledge acquisition document; after the conceptualization, a conceptual model document that includes a set of intermediate representations that describe the application domain; after the formalization, a formalization document; after the integration, an integration document; after the implementation, the implementation document; and during the evaluation, an evaluation document.

2.2 CommonKADS

CommonKADS [22] is widely used ontology development methodology. CommonKADS is focused on the work product-based rather than process-oriented. It supports every aspect of knowledge engineering including project management, group analysis, knowledge acquisition, conceptual modeling, user interface, system integration and architecture [23]. Fig. 3 summarizes the suite of models involved in a ComrnonKADS project. A central model in the CommonKADS methodology is the expertise model, which models the problem solving behavior of an agent in terms of the knowledge that is applied to perform a certain task.

Fig. 3: The CommonKADS suite of models



The CommonKADS model set provides four models that are specifically geared to modeling the organizational environment of a KBS: the organization, task, agent and communication models.

The *organization model* supports the analysis of the major features of an organization to discover problems and opportunities for KBS development, as well as possible effects a KBS could have when fielded. A template that defines object and relation types is associated with each model in the model set.

The *task model* describes, at a general level, the tasks that are performed or will be performed in the organization where the expert system will be installed. The tasks it covers are those that help realize an organizational function. The task model is represented as a hierarchy of tasks. In addition, aspects like inputs and outputs of tasks, task features, and task requirements can be modeled. The task model also specifies the distribution of tasks over agents.

An *agent* is an executor of a task. It can be human, computer software, or any other "entity" capable of executing a task. In the *agent model*, the capabilities of each agent are described. The model can also be used to represent constraints on an agent, such as norms, preferences, and permissions that apply to the agent.

Because several agents are usually involved in a task, it is important to model the communication between agents. This is the purpose of the CommonKADS *communication model*. The transactions here are modeled at a level that is still independent of a computational realization.

2.3 Methodology by Farooq et. al.

In their paper [24], they proposed a technique for ontology design during SW application engineering process. By incorporating their technique, existing web applications design methods may easily be upgraded for semantic web (SW) applications. They made some deliberate efforts in design phase of ontology development which they found missing in other methods.

2.3.1 Adaptation at Specification Level

For ontology, requirements should be specified accordingly, in specification phase. A preliminary web-ontology model should be prepared at specification level. The activities involved, are as below:

a) Domain Vocabulary Declaration
b) Identifying resources and assigning them to proper groups
c) Identifying Axioms
d) Identifying relationships and assigning them proper names
e) Identifying data-characteristics and assigning them, proper names
f) Applying constraints
g) Verification

2.3.2 Adaptation at Design Level

The processing of design phase mainly uses the report generated by specification phase, and transforms it into some algorithmic or pseudo form so that it can be coded easily in any computer language in order to make it executable. Since ontology (schema and document) is based on Resource Description Framework model, therefore they designed a model, so-called RDF model, from preliminary ontology model generated in previous phase. This model consists of triples. A triple contains three components: (i) subject (ii) predicate and (iii) object. Each name in RDF model is a URI reference or a literal.

2.4 Methodology by Gaoyun et. al.

In their paper [19], they firstly analyzed four reasons for occurrences of inconsistencies in ontologies. Then based on the reasons they propose a methodology for ontology construction to minimize the distinctions as follows:

2.4.1 Clarifying domain

In order to clarify the domain, following three problems should be solved: (1) the scope the domain covers, (2) the contents the domain describe and (3) the viewpoint of the domain uses. All the answers of the problems depend on the users of ontology. Consequently the following questions should be considered as we are users: Who would be the users of ontology? What can we do by this ontology? What problems can we solve by this ontology?

2.4.2 Building domain model

After clarifying domain, the domain model construction is separated into two phases: building physical model of domain and building concept model of domain to minimize inconsistencies brought by subjective factors.

2.4.2.1 Building physical model of domain

As above analysis ontology construction highly depends on subjective factors, which is a major reason why ontologies built have such inconsistencies from one another. To construct consistent ontologies, the influences affected by subjective factors should be reduced and the process of concepts extracting should be suspended. To model the domain we should begin with objective description of real world, describe objects and relations between them and then build physical model of domain. This less abstract physical model is easier to reach agreement among creators, and then abstract it unceasingly to higher concept model.

2.4.2.2 Building concept model of domain

What's more based on a widely recognized physical model, in concept model if there are inconsistencies which are caused by the degree of abstraction, we just need to unify the inconsistencies by identifying a proper granularity for concept. Attribute of concept can aid this identifying process which would be introduced in next subsection.

2.4.3 Identifying attributes

By attribute two concepts are contacted with each other, and thanks to attribute the simple concept tree which has inherited relations only could be converted to concept web which has abundant relations and could describe complex relations of the world more exactly.

2.4.4 Layered vocabulary

Another problem that may lead to inconsistencies is caused by the case that several terms aim at one meaning. To unify the terms used in the ontology, requiring creators to use the controlled vocabulary is a good choice. But it is not the same good to compel users to use words in the controlled vocabulary either. For this reason the vocabulary is divided into three grades: controlled vocabulary, domain special vocabulary and common vocabulary. The terms in controlled vocabulary which is used in ontology construction originate from various metadata of domain to guarantee consistency of terms using. The terms in domain special vocabulary and common vocabulary come from daily words used by experts and ordinary people respectively to ensure richness of words that can be used.

2.4.5 Merging ontology

Reusing existed ontologies could reduce the work of construction and improve efficiency. Even developing ontology by one community independently, ontology division is also a good choice. These two cases are both involved in ontology merging finally. This process would bring inconsistencies if the chosen ontologies are inappropriate.

Therefore, when choosing ontologies the following two points should be ensured. Firstly ensure the viewpoints of ontologies should be the same. Secondly, part concepts of the one chosen ontology should be equivalent to or subsumption of some concepts of the other one that is parts of ontologies should be compatible without conflictions. Moreover for ontology reuse it is crucial to transform terms with controlled vocabulary.

2.4.6 Ontology formalization

The fundamental purpose of ontology is to aid machine comprehending the meaning of information which is coded by ontology language. OWL has been accepted as the ontology description language by the authors.

2.5 MADRE

The authors, Zhang et. al., realized that because of the differences between domains, there was no existing technical route as the standard method for ontology construction at the present time. The paper [25] offers an improved ontology construction method MADRE (Method of Analysis, Design, Representation and Evaluation) based on IDEF5 and seven steps methodologies. This method uses graphic language to explicit represent domain knowledge for research domain. Moreover, it verifies correctness of ontology construction on relation and hierarchy in the phase of ontology evaluation.

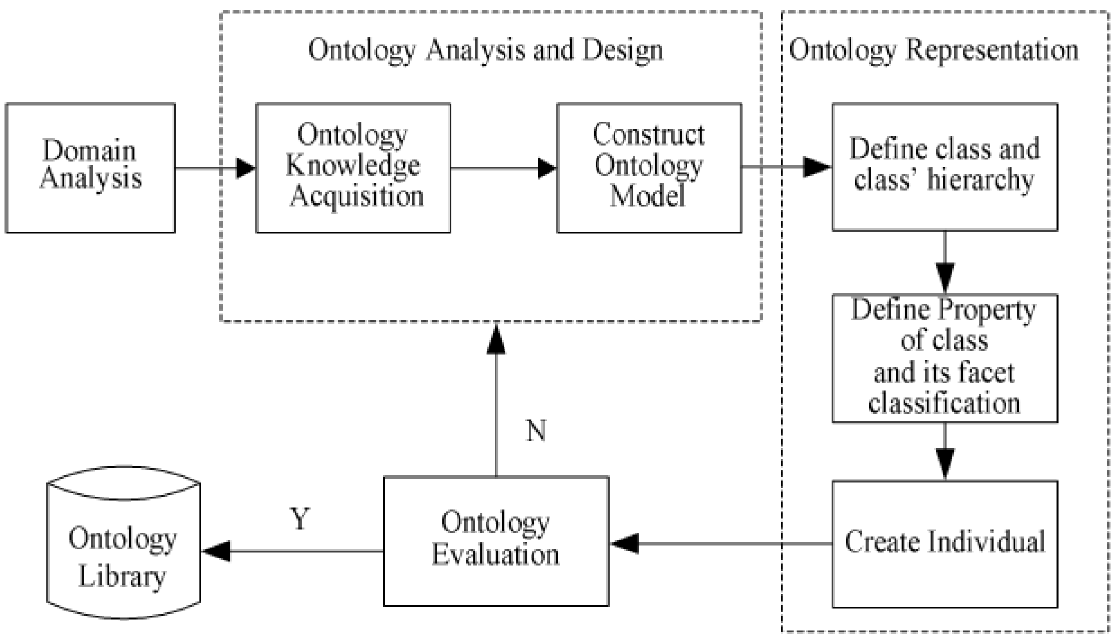
Following the Seven Steps method, the construction is divided into four steps, which are domain analysis, ontology analysis and design, ontology representation and ontology evaluation. Meanwhile, in the phase of ontology analysis and design, MADRE adopts expression of the graphic language of IDEF5. The phase of ontology evaluation in MADRE is able to make an accurate judgment of created ontology. The specific flowchart to construct ontology with MADRE is as follows:

- Use Phase of Domain Analysis: determine the professional field extension and reusability of domain ontology.
- Phase of Ontology Analysis and Design: acquires semantic information about core concept, relations, actions and so forth, define the hierarchy of concepts and relations between different activities, formally translates the professional

knowledge and raw data of the current domain to commonly used information based on ontology, and therefore establishes the ontology model.

- Phase of Ontology Representation: according to the ontology model, detailed specific the class and properties of the domain ontology, and creates individual. Meanwhile, represents the domain ontology with structural language-OWL.
- Phase of Ontology Evaluation: establishes rule set, evaluates the accurateness and coincidence of ontology by the First-Order Logic inference. If it is consistent with the domain knowledge, the construction of ontology succeeds; otherwise, go back to Phase of Ontology Analysis and Design.

Fig.4: The process of MADRE ontology construction

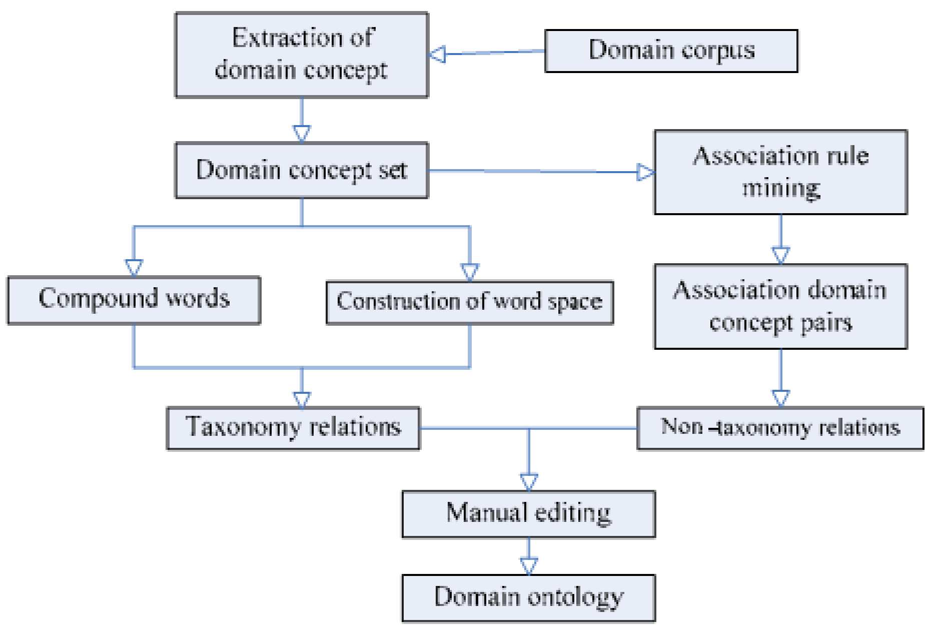


2.6 Semi-automatic Domain Ontology Construction methodology by Dan et. al.

In their paper [26], an approach to semi-automatically constructing domain ontology based on Chinese word partition and data mining is proposed.

The semi-automatic domain ontology system designed in this paper mainly consists of three parts: the extraction module of domain concepts and the extraction module of taxonomy and non-taxonomy relations among domain concepts. Statistical analysis method, generalized suffix tree and clustering method and association rule mining method are respectively adopted in the above three parts. The system framework is illustrated in Figure 5.

Fig.5: The system framework



2.7 Automatic Ontology Construction Approach by Jia et. al

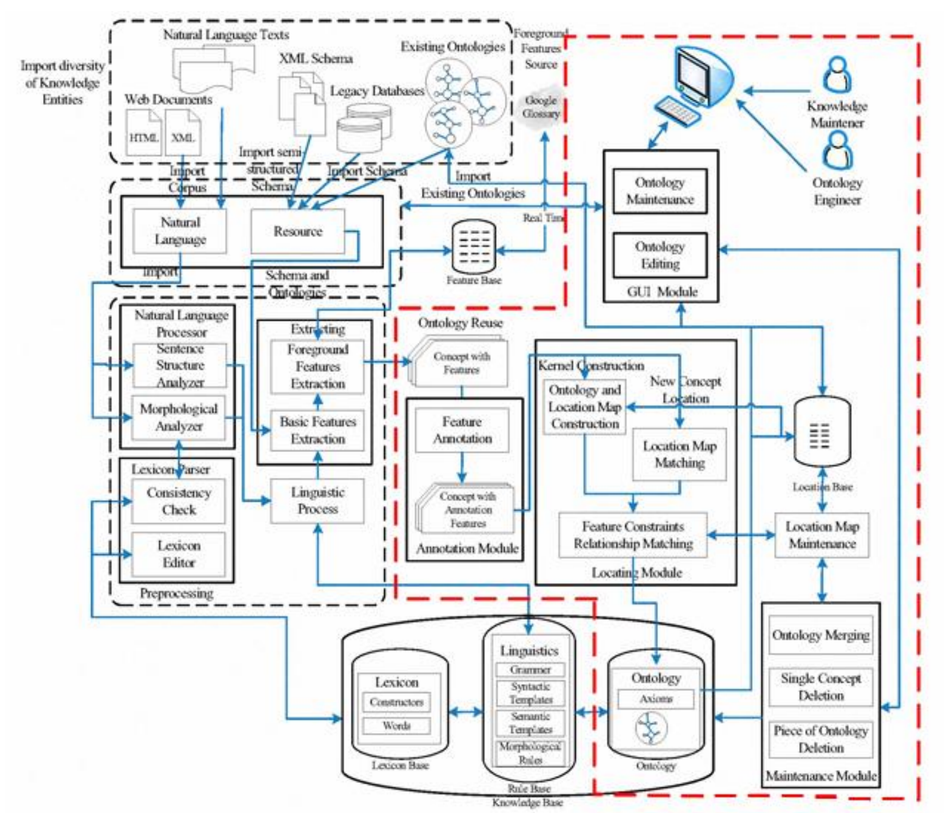
In this paper, thesaurus and oracle database are selected as the existing resources in the military intelligence. The thesaurus is relatively complete and accurate knowledge resource, and it contains 61895 terms and 3 kinds of relationships between concepts. The classes of the ontology mainly are obtained from thesaurus, while the database provides the instances for the ontology. Protégé is used as the development tool and OWL as ontology language.

The authors combine the characteristic thesaurus. First, they select the standardized thematic terminology as the classes of ontology and give up the non-thematic terminology; second, semantic relations should be adjusted in order to obtain accurate semantic relations, which is the core job needed to be done in the thesaurus. In order to resolve this problem, all of the concepts in the thesaurus are classified by the semantic categories, and then the relations will be converted based on the concepts of each category. At the same time, considering the characteristic of relational database, the authors propose a series of rules to achieve the target that append the instances into the ontology automatically. Finally, following some certain steps as they show4, the ontology is constructed completely.

2.8 Concept Feature-based Ontology Construction and Maintenance

The study [15] develops a concept feature-based mechanism for constructing and maintaining ontology. In addition to assisting enterprise knowledge engineers in constructing and classifying knowledge more precisely to increase knowledge maintenance efficiency, the proposed mechanism assists knowledge users in accurately searching for required knowledge based on use of concept features. They designed a concept feature-based ontology construction and maintenance framework, developed techniques related to the concept feature-based ontology construction and maintenance, and implemented a concept feature-based ontology construction and maintenance mechanism. The proposed framework includes six modules of import, preprocessing, annotation, locating, maintenance, and graphic user interface (GUI), as shown in figure 6.

Fig.6: Concept feature-based ontology construction and maintenance framework.



2.9 A structured ontology construction by using data clustering and pattern tree mining

The paper [14] proposes an automatic system of ontology construction to link the relations between concepts. The system contains two major functions: document clustering and ontology construction. The former mainly groups related documents by clustering method. The latter discovers inter-concept relations from each group of related documents and integrates all the generated conceptual relations in light of sequence patterns to construct ontology. The system is hence suitable for domains with a larger scope. It can construct an ontology which is able to describe inter-concept relations in more detail as well as more finely and reasonably structured.

2.9.1 Ontology construction

Traditionally, ontology construction usually uses the following methods for ontology construction: *relational analysis*, *clustering*, and *formal concept analysis (FCA)*.

Relational analysis discovers and clusters the relations of keywords, such as synonyms, roots, hypernyms, and hyponyms. It then constructs the ontology by manmade or other methods. *Clustering* usually groups keywords of documents into clusters and constructs ontology by selecting representative concepts from each cluster. Formal concept analysis uses the binary relation matrix between documents and vocabulary to generate the supremum concept set. The inter-conceptual hierarchical relation and a complete partial sort is formed by the sets of all the concepts. After constructing the inter-concept level relation by means of *FCA* construction, the concept figure of ontology is thus constructed.

2.9.2 System Architecture

The architecture of the structured ontology construction system contains two modules: *document clustering* and *ontology construction*. Basically, document clustering is responsible for document vector computation and document clustering. Ontology construction establishes the concept tree of the clustered documents and integrates them into complete document ontology.

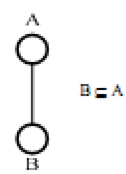
Figure 7 gives the document vector algorithm. Figure 8 gives the concept tree construction algorithm and Figure 9 gives the sequence pattern mining algorithm used in ontology construction. Finally figure 10 represents the ontology construction algorithm.

Fig.7: Document vector algorithm

1. Establish the matrix of latent semantic singular value decomposition: $R \circ S' \circ M^T = A$
2. Set up the dimension of document importance and establish the dimension-reduction matrix (S) of new documents and keywords in the latent semantic space.
3. Calculate the feature vector of documents: $F = R \circ S$.

Fig.8: Concept tree construction algorithm

1. Establish the two-dimensional matrix of documents and keywords for each group of documents. The documents to which Keyword A corresponded were more than those to which Keyword B corresponded, and no other keywords corresponding to documents existed between A and B, so a link was established between A and B, the result of which constructed a concept lattice.



2. Delete the empty concept nodes which were not root nodes in the concept lattice in order to form concept trees.

Fig.9: Sequence pattern mining algorithm

1. Take all the root-to-leave path patterns of the concept trees and sequence the concepts in all the patterns to generate candidate concept sets.
2. In the candidate concept sets, all the single concepts construct a single concept set. In single concept sets, find out the set which is greater than or equal to the minimum support, namely the highly frequent single-concept set. Repeat the steps to discover the highly frequent two-concept set, the highly frequent three-concept set, the highly frequent four-concept set, etc. until no highly frequent concept set can be discovered, so that the referentially valuable sequence pattern structure will be found.
3. Delete sub-sequences from the mined longest sequence pattern. For example, the longest 4-concept set is <1,2,3,4>. If the 3-concept set includes <1,2,3> and <1,2,4>, delete them, so the rest is the sequence pattern set.

Fig.10: Ontology construction algorithm

1. Select the longest pattern structure as the ontology skeleton and add the mined patterns to the structure tree according to the pattern length and occurrence frequency.
2. Through WordNet, combine with the Jaccard similarity (Sim) to obtain the distance (Dis) between root nodes. That is, the more similar they are, the shorter the distance is. The closest root nodes are adopted to generate common hypernyms; the common hypernyms of the keywords of the root nodes in the ontology skeleton are established, and the new root nodes of hypernyms are created.

$$Sim(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

$$Dis(A,B) = 1 - Sim(A,B)$$

where A and B are two different root nodes, and S is between 0 and 1. The greater the value is, the more similar A and B are.
3. Repeat Step 2 until all the sub-trees in all the ontology skeletons find their common hypernyms.
4. Link the hypernyms with the structure trees to form the ontology structures.