Soil Knowledge-based Systems Using Ontology

Tongpool Heeptaisong and Anongnart Shivihok

Abstract—**The aim of this paper is to develop a soil knowledge-based system by using ontology (SOKS). We applied XPath algorithm and automatic term weighting of ontology to improve the performance of this system. The system provides functions to search from various sources such as HTML, databases and digital libraries on the Internet and support knowledge sharing and knowledge reuse which is the important process in knowledge management. The system can improve performance of document retrieval in term of precision value is 0.9 which outperform the traditional system.**

***Index Terms*— Soil, Ontology, Agriculture Knowledgebase, Semantic Web, Knowledgebase System.**

I. Introduction

KNOWLEDGE of soil is very important for farmers because it is one significant factor which effects the plant farming and production. Up to now, soil knowledge and information is stored extensively in various formats on the internet such as HTML, databases and digital libraries. The various formats are the main problem when searching or retrieving information. Thus, researchers continually try to find new methods to improve the performance of their system. The ontology is the one popular method which uses semantic and knowledge representation with other information systems to enhance performance of their retrieval system.

The objective of this study is to develop a soil knowledge-based systems using ontology to assist the search of soil knowledge which is stored in various sources.

This paper is organized as follows: Section 2 describes related works and tools which are used to create an ontology. Section 3 is the proposed framework of the soil knowledge-based system. Section 4 covers the experimental results and discussion. Finally, the conclusions and future work are drawn in section 5.

II. Related Works

A. What is an Ontology

Ontology is formed in terms format and their relationships. Furthermore, it is grouped into classes and subclasses of relationships [1]. Ontology represents knowledge and sharing in a specific domain. Moreover, it enables reuse of the domain knowledge such as WordNet [16], ARGOVOC [17] etc. Recently, ontology is widely used for semantic web searches with information systems. Ontology building can be developed via two methods: by experts [7, 13-15] and by terms of a document extracted automatically [3, 8, 11]. Normally, the first method is more successful and found to be the most popular.

Ontology software is very important when building a successful ontology. Popular software for this task are Hozo [16] and Protégé [17] because they can be used to develop ontology structure easily and can support Resource Description Framework (RDF), Ontology Web Language (OWL), Extensive Markup Language (XML) and a standard of W3C. This research used Hozo editor to create soil ontology.

B. Using Ontology with Information System

Ontology has an important role with semantic information retrieval as ontology is used to improve the performance of retrieval processes and can be used to access different sources. Nowadays, ontology is widely used in various areas such as medical retrieval systems [5, 9], e-learning systems [2, 12], Thai succession law systems [3], agricultural knowledge management [10] and personalized systems [4]. Furthermore, ontology is used to define knowledge domains.

C. Related Articles

Bhavani and coworkers used the topic ontology mapping method with e-Learning system to access different sources in each chapter. This research designed ontology relationship with Protégé software and exported it as an OWL file [2].

Boonchom and Soonthornphisaj proposed an automatic ontology building method referring to Thai succession law using a new algorithm called Ant Colony. This method can create automatic ontology and is accepted by Thai law specialist, afterwards they used this ontology with information retrieval systems [3].

Buranarach and coworkers used ontology combined with healthcare systems to predict chronic disease patients because it predicts with high accuracy. Moreover, this system can support knowledge rules from knowledge engineers [7].

Xiuqin and Jun proposed the key steps to implement semantic web to management agricultural knowledge which is broadly distributed on Internet [10].

III. Soil Knowledge-based Systems Framework

This system includes three processing, which has been designed to be automatic and easy-to-use for user satisfaction, as shown in Fig. 1.

Manuscript received January 10, 2012; revised January 18, 2012. This work is supported by Thai Government Science and Technology Scholarship, Ministry of Science and Technology, Department of Computer Science, Faculty of Science, Kasetsart University, Thailand. Special thanks Assoc.Prof. Dr. Yongyuth Osotsapar and Dr. Marut Buranarach who are soil science and ontology specialist.

T. Heeptaisong is a PhD's student at Department of Computer Science, Faculty of Science, Kasetsart University, Bangkok, Thailand (e-mail: g5184040@ku.ac.th).

A. Srivihok is a lecturer at Department of Computer Science, Faculty of Science, Kasetsart University, Bangkok, Thailand (e-mail: fsciang@ku.ac.th).

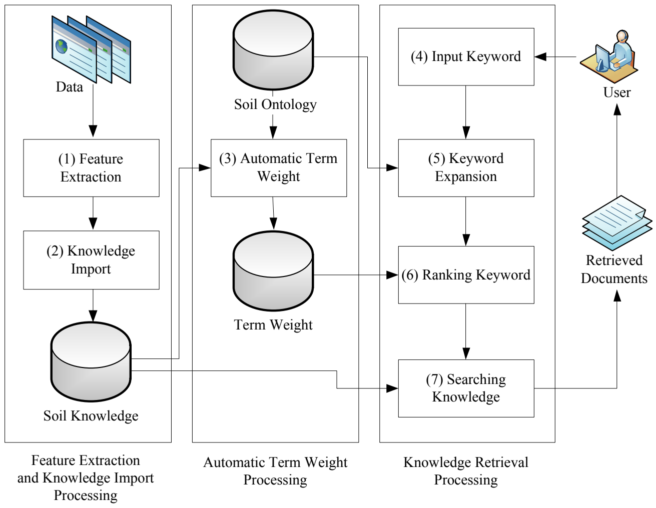


Fig. 1. Soil Knowledge-based Systems Framework.

A. *Feature Extraction and Knowledge Import Processing*

Feature extraction processing is cleaning process to transform unstructured data to structure data. In this case, we used data from two sources in [18, 20] as shown in Fig. 2. Most of data is web pages format therefore it must was arranged to be standard knowledge. Data may be in different format such as databases, web pages or digital libraries. For instance, digital libraries and databases format are a title, author, email, abstract, detail, publish year and etc., but web pages format is unstructured. After feature extraction processing, these structure data are imported to soil knowledge.

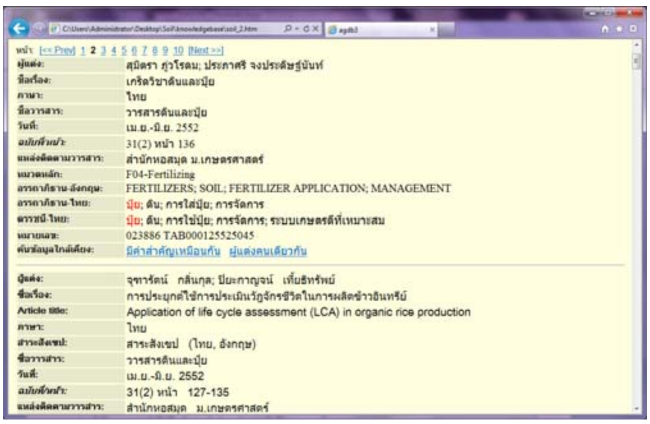


Fig. 2. Unstructure data from web pages.

Feature extraction and knowledge import processing are described as follows:

Load data
Do While (data is not empty)
*Identify format from input data (digital libraries, web pages or database)***.**
Delete HTML tag from the data.
Change to standard format.
Import to soil knowledge repository.
Loop

This process involves four steps:

- Step 1, we input unstructured data, the feature extraction processing identifies input data formats such as digital libraries, web pages or databases.
- Step 2, the process eliminates HTML tag from the data for example <html />, <body />, <table />, <tr />, <td />, <h1 /> and etc.
- Step 3, the process replaces standard format to extracted data.
- Step 4, these data import to soil knowledge repository.

B. *Automatic Term Weight Processing*

Automatic term weight processing extracts terms from soil ontology and use those terms for calculating term weights. After soil ontology development in [7], we have found that the ontology includes all 84 nodes and 83 relationships as shown in appendix A. For relation building we use both "Is-a" and “Part-of”. The soil ontology must be stored in the OWL file as shown in Fig. 3. We edited some tag of OWL file to be easy for finding label or term in ontology using XPath algorithm.

```
<soil>
<rootClass resource="soil" ID="100">
        <subClass SubClassOf="soil">genesis</subClass>
        <subClass SubClassOf="soil">property</subClass>
        <subClass SubClassOf="soil">chemical</subClass>
        <subClass SubClassOf="soil">microbiology</subClass>
        <subClass SubClassOf="soil">organic</subClass>
        <subClass SubClassOf="soil">fertility</subClass>
        <subClass SubClassOf="soil">fertilizer</subClass>
        <subClass SubClassOf="soil">morphology</subClass>
        <subClass SubClassOf="soil">survey</subClass>
        <subClass SubClassOf="soil">classification</subClass>
        <subClass SubClassOf="soil">erosion</subClass>
        <subClass SubClassOf="soil">conservation</subClass>
        <subClass SubClassOf="soil">management</subClass>
    </rootClass>
    <subClass SubClassOf="soil" resource="genesis" ID="101">
      <label>material</label> <label>formation</label>
      <label>weathering</label></subClass>
    <subClass SubClassOf="soil" resource="property" ID="102">
          <label>texture</label><label>density</label>
          <label>porosity</label><label>structure</label>
          <label>color</label><label>moisture</label>
          <label>temperature</label><label>aeration</label>
</subClass>
</soil>
```

Fig. 3. Example of soil ontology in OWL file.

In this paper, we used TF-IDF technique to define term weight. Given K is a set of keyword in soil ontology, TF is term frequency, DF is document frequency and IDF is log (N/DF). We describe about defining automatic term weight as following:

Load all terms in OWL file to memory.
N = Total number of document in soil repository.

Do While (K is not empty set)
DF_i *= Total number of document frequency retrieved soil repository by* K_i

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TF_i = Total number of term frequency appeared in soil repository by K_i
IDF_i = log (N/DF_i)
$TF\text{-}IDF_i = TF_i * IDF_i$
Insert K_i and $TF\text{-}IDF_i$ to Term Weight
Loop

This process involves four steps:

- Step 1, this process loads all terms in OWL file to memory such as soil, genesis, property, fertilizer and etc.
- Step 2, while step 1 finished, the process counts total number of document in soil repository.
- Step 3, the process calculates DF_i, TF_i, IDF_i and $TF\text{-}IDF_i$ values.
- Step 4, the process inserts K_i and *TF-IDF* value to Term Weight.

C. Knowledge Retrieval Processing

Keyword expansion step, after the user inputs a keyword, this sub-processing will expand the keyword with soil ontology and use entity mapping [6] which is an easy and accurate method. For example keyword expansion of organic keyword will include humus, humic acid, immobilization, immobilization and mineralization of the keyword from the ontology. Keyword expansion from ontology, we have applied XPath algorithm of XML to accomplish this, as following:

Receive a keyword from user

Load Soil Ontology

For each keyword in Soil Ontology

If keyword is found then

Keyword += Child of keyword in Soil Ontology

End if

Loop

Return Keyword

This process involves three steps:

- Step 1, this process receives a keyword from user.
- Step 2, while step 1 finished, the process loads soil ontology file.
- Step 3, the process will match the keyword with ontology and keep keyword expansion.

Ranking keyword step, this step will add weight to each keyword. Each keyword will be matched with term weight which used automatic tf * idf to define keyword weight in automatic term weight processing. For instants, organic keyword after expansion from an ontology file will result in organic (196.39), mineralization (34.68), immobilization (8.82), humus (2.94) and humic acid (2.94) as shown in table 1.

TABLE I
AUTOMATIC RANKING KEYWORD BY TF * IDF TECHNIQUE

Keyword Expansion by Ontology	
Non term weighting	Automatic term weighting
organic	organic (196.39)
humic acid	mineralization(34.68)
humus	immobilization (8.82)
immobilization	humus (2.94)
mineralization	humic acid (2.94)

Searching knowledge is the final step for knowledge retrieval processing. This process will search soil knowledge with ranked keywords. Furthermore, it will rank documents accessing to keyword term weighting as shown in Fig. 4.

Searching knowledge algorithm.

*Query = "SELECT TOP 50 * FROM SoilKnowledge WHERE "*

For each Keyword

Query += " $Att_{i..N}$=$Keyword_{i..k}$ "

Loop

Query += " order by DocSore "

Return Query

This process involves two steps:

- Step 1, Search knowledge with ranked keywords.
- Step 2, Show knowledge from the top 50 document scores.

IV. EXPERIMENT RESULTS

A. Soil Knowledge-based Systems Using Ontology

This system was developed using the PHP programming language and MySQL as the database management systems, as shown in Fig. 4-6.

Fig. 4. Soil ontology knowledge-based systems by using soil as keyword.

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

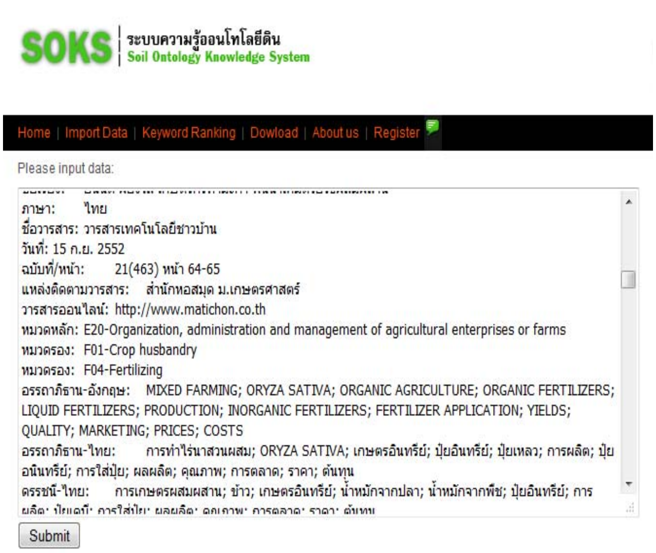


Fig. 5. Unstructure documents from digital library.

No.	Term	DF	IDF	TF	TF*IDF
1	acidity alkalinity	1	2.94	1	2.94
2	actinomycets	1	2.94	1	2.94
3	aeration	3	2.47	4	9.88
4	algae	1	2.94	1	2.94
5	bacteria	5	2.25	7	15.75
6	biofertilizer	88	1	89	89
7	biological method	1	2.94	1	2.94
8	calcium	2	2.64	3	7.92
9	chemical	42	1.32	60	79.2
10	chemical fertilizer	24	1.56	38	59.28
11	classification	5	2.25	8	18
12	colloid	1	2.94	1	2.94
13	color	1	2.94	2	5.88
14	conservation	11	1.9	19	36.1
15	density	1	2.94	1	2.94
16	erosion	5	2.25	10	22.5
17	essential nutrient element	1	2.94	1	2.94
18	exchange	2	2.64	4	10.56

Fig. 6. Keyword raking by SOKS.

B. Results

We applied soil ontology to the system and tested it with information retrieval. The result shows the improvement of performance retrieval processes at 90 percent compare with traditional system, as shown in table 2. We use precision equation from (1) to measure the performance of this system.

$$\text{Precision} = \frac{\text{number of relavant document retrieved}}{\text{number of document retrieved}} \quad (1)$$

TABLE II
COMPARISONS OF EFFICIENCIES IN INFORMATION RETRIEVAL USING TRADITIONAL SYSTEMS AND SOKS

English keywords	Precision	
	Traditional System	**SOKS**
soil	0.34	0.86
genesis	0.10	0.70
property	0.74	0.94
chemistry	0.90	0.96
microbiology	0.03	0.90
organic	0.95	0.97
fertility	0.67	0.98
fertilizer	0.32	0.99
morphology	0.25	0.5
survey	0.37	0.75
classification	0.57	0.71
erosion	0.71	0.71
conservation	0.53	0.73
management	0.42	0.95
Average	**0.53**	**0.90**

From table 2, we can summarize that precision of SOKS. This system can improve the performance of keyword search. It can support user's requirements with high accuracy (precision=0.9) when searching for soil knowledge.

V. Conclusion

The aim of this paper was development of a soil knowledge-based systems (SOKS) using ontology. We easily applied XPath of XML and PHP language to improve the performance of this system, therefore we added term weight by automatic TF*IDF. This system can search knowledge from multiple sources such as HTML, documents or databases on the Internet and supports knowledge sharing and knowledge reuse which is an important process in knowledge management. As a result, the system can improve performance of document retrieval in term of precision value is 0.9 which outperform the traditional system.

For future work, the application of multiple ontologies with knowledge retrieval is an interesting issue. The keyword expansion and term weight algorithms should be concerned.

References

[1] Noy, N.F., Mcguinness, D.L. (2001). Ontology Development 101: A Guide to Creating Your First Ontology. Standford Knowledge Systems Laboratory Technical Repoort KSL_01_05.

[2] Bhavani, S., Hepu, D. and Brian C. (2009). An Ontology-Driven Topic Mapping Approach to Multi-Level Mamagement of e-Learning Resources. Journal 17th European Conference on Information Systems.

[3] Boonchom, V. and N. Soonthornphisaj (2010). Thai succession and family law ontology building using ant colony algorithm. Proceedings of the 2009 international conference on New frontiers in artificial intelligence. Tokyo, Japan, Springer-Verlag: 19-32.

[4] Castells, P., M. Fernández, et al. (2005). Self-Tuning Personalized Information Retrieval in an Ontology-Based Framework. Proceedings of the 1st IFIP WG 2.12 & WG 12.4 International Workshop on Web Semantics (SWWS 2005). LNCS 3762: 977-986.

[5] Dridi, O. and M. Ben Ahmed (2008). Building an Ontology-Based Framework For Semantic Information Retrieval: Application To Breast Cancer. Information and Communication Technologies: From Theory to Applications.

[6] Gal, A. and P. Shvaiko (2009). Advances in Ontology Matching. Advances in Web Semantics I. S. D. Tharam, C. Elizabeth, M. Robert and S. Katia, Springer-Verlag: 176-198.

[7] Heeptaisong, T. and A. Srivihok (2010). Ontology Development for Searching Soil Knowledge. The 9th International Conference on e-Business (iNCEB2010), November 18-19, 2010, Bangkok, Thailand.

[8] Imsombut, A. and A. Kawtrakul (2008). Automatic Building of an Ontology on the Basis of Text Corpora in Thai. Language Resources and Evaluation 42(2): 137-149.

[9] William Philip Taylor II (2010). Creating a Biomedical Ontology Indexed Searching Engine to Improve the Semantic Relevance of Retrieved Medical Text. A Dissertation Presentation to the Graduate School of Clemson University.

[10] Xiuqin, Q. and Y. Jun (2010). Ontology Based Distributed Agricultural Knowledge Management. Fuzzy Systems and Knowledge Discovery (FSKD), Seventh International Conference on.

[11] Yilmazel, O., C. M. Finneran, et al. (2004). MetaExtract: An NLP System to Automatically Assign Metadata.Digital Libraries. Proceedings of the 2004 Joint ACM/IEEE Conference on.

[12] Zhu, H., Q. Tian, et al. (2009). Domain Ontology Component-Based Semantic Information Integration. Proceedings of the 2009 First International Workshop on Education Technology and Computer Science - Volume 03, IEEE Computer Society: 101-104.

[13] Thunkijjanukij, A., A. Kawtrakul, et al. (2009). Ontology Development: A Case Study for Thai Rice. Kasetsart Journal (Natural Science) 43(3): 594-604.

[14] Fortuna, B., D. Mladenić, et al (2005). Semi-automatic construction of topic ontology. SIKDD 2005 at multiconference IS 2005. Ljubljana, Slovenia.

[15] Buranarach M, Chalorthum N, et al (2009). An Ontology-based Framework for Development of Clinical Reminder System to Support Chronic Disease Healthcare. The 4th International Symposium on Biomedical Engineering.

[16] Hozo Ontology Editor, Available source http://www.hozo.jp

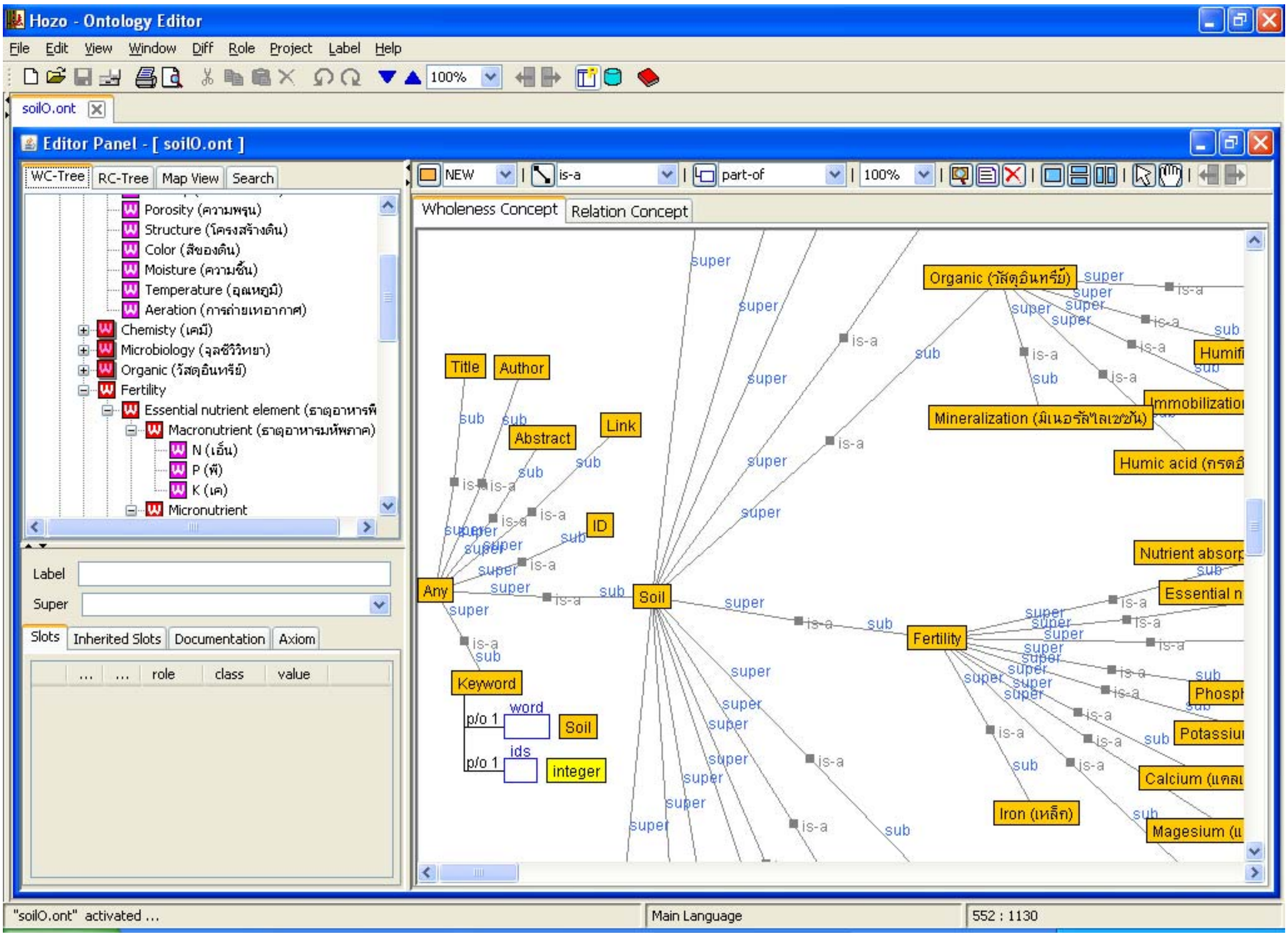
[17] Protégé Ontology Editor and Knowledge Acquisition System, Available source http://protege.stanford.edu

[18] Knowledge Ariculture Available source http://pikul.lib.ku.ac.th/agre

[19] WordNet - a lexical database for English, Available source http://wordnet.princeton.edu

[20] AGROVOC Thesaurus, Available source http://www.icpa.ro/AgroWeb/AIC/RACC/Agrovoc.htm

Appendix A. Example of soil ontology development using Hozo editor.



ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)