Adaptive Novice Supporter

Sung-Pil Choi, Ho-Jin Choi, Sung-Hyon Myaeng
School of Engineering, ICU
119, Munjiro, Yuseong-gu, Daejeon, 305-732, Korea
{spchoi, hjchoi, myaeng}@icu.ac.kr

Abstract

In this paper, we implemented very simple plan generator which can help the novice researchers do their research efficiently. We have assumed that we should not make domain experts annoying when building the knowledge-base, and at the same time, we could provide users with the beneficial information made by them using very simple mechanisms. This is the main purpose of implementing ANOS (Adaptive Novice Supporter). We could realized that in order to overcome some limitations of ANOS, we should perform several further researches including the adoption of machine learning methods and the enhancement of knowledge-base providing much intensive user's surveying.

1. Introduction

Usually, a lot of new researchers who want to start their research in certain academic areas tend to have severe difficulty in planning or scheduling their initial research process. Some people luckily meet good academic advisors or capable seniors who are so attentive and polite that they are willing to help these novices. In many cases, however, they could spend much time to set up their starting point of study suffering from wandering over numerous wrong places. What is worse, in spit of those painful efforts, they could not find even their incipient places of studying, such as fundamental textbooks, thesis, web resources which might be very beneficial for them to conduct their research very efficiently. Even though they barely manage to find their starting points, they should waste additional time and efforts to settle down their [1]ways of good research.

As known to every scientific researchers or practitioners, in the world of science, each of almost all the scientific domains can be seen as a large lattice structure composed of its sub-domains and academic resources related to the sub-domains. To some high-level experts of the domain, it might not be difficult to define and construct a reasonably specialized scientific domain as this lattice structure. And this resulting structure can help above newcomers very effectively.

Then, how could those experts express their expertise easily and conveniently? What is the most important aspect in describing the lattice structure? Can we create and support the tools or other methodologies to these active experts? There have been so many efforts to answer these questions for a long time especially in AI community. Among them, expert system is one of the solutions which has been considered as the most reasonable and effective one.

The primary goal of expert systems research is to make expertise available to decision makers and technicians who need answers quickly. There is never enough expertise to go around - certainly it is not always available at the right place and the right time. Portable with computers loaded with in-depth knowledge of specific subjects can bring decade's worth of knowledge to a problem. The same systems can assist supervisors and managers with situation assessment and long-range planning. Many small systems now exist that bring a narrow slice of in-depth knowledge to a specific problem, and these provide evidence that the broader goal is achievable. [2]

Due to the invention of XML and semantic web, many people have become familiar with describing structured information and even well-organized knowledge. The most attractive feature of the framework of semantic web is that it can provide explicit mechanisms for creating and re-using knowledge. Because of this benefit, domain experts could write their own expertise into knowledge base using some functionality such as RDF, RDFS and OWL standards.

In this paper, we propose a very simple plan generator, named ANOS (Adaptive Novice Supporter) based on the knowledge base built by using RDF and RDFS. The main function of the system is to provide someone who is not familiar with a certain topic with useful information about the topic needed in starting his or her initial study or research. ANOS can help many new-comers in certain scientific and technological areas start and perform their research effectively. In addition to that, it will have a few fundamental functions related to suggesting useful information and related sub-topics, planning their research process with regard to the order of referencing the information. Finally, ANOS will provide experts with

[1] This research was performed for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Commerce, Industry and Energy of Korea.

efficient mechanisms to describe their expertise so that many new-comers will benefit by using it.

The remainder of this paper is organized as follows. In the next section, the problem descriptions and assumptions of this paper will be introduced briefly. Section 3 introduces ANOS in more detailed manner. The implementation issues and problems are discussed in section 4. In section 5, we present some limitations of the current version of ANOS and propose some enhancing mechanisms to overcome them. The last section concludes with the results of implementation of ANOS and the educational effects of this system.

2. Problem Descriptions and Assumptions

2.1 Difficulties of Launching Research

It is likely that many people engaged in scientific research areas tend to suffer great difficulty when they start studying or conducting new research areas which have never experienced yet before. To get out of this dark ignorance, they have to be at pains to locate some excellent initial points for their study. While doing that, they repeatedly feel heavy and anxious about whether their decisions to start doing a research in the field were good or bad, and eventually they have to give up their trials in spite of their good backgrounds and capabilities.

Due to the invention of World Wide Web (WWW), it firstly seemed that this difficulty can be settled down easily. However, owing to the tremendous scale of information on the web, people now have another difficulty of finding their best-fitted web resources. Although lots of people are now trying to enhance the information accessing mechanisms on the web, users' discontent is still so high. In order to give novice researchers more chance to feel comfortable and to make them more passionate when they start new trials, it is certainly necessary that seniors or experts who have experienced the difficulty too should be involved anyway.

2.2 Users of ANOS

In the previous sub-section, we mentioned newcomers or novices of some research domains who are willing to strive to spend their time and effort in studying. For the purpose of providing initial version of ANOS, we assume that the users of ANOS have some features or ability to evaluate their own level of knowledge. Table 1 shows the assumption about the users of ANOS.

There are two different kinds of user assumptions in ANOS. One is that a user can determine his or her level of knowledge of certain research topics and also can evaluation the estimation by specifying one of the three values, “low,” “middle,” and “high.”

Table 1. Users of ANOS (User Assumptions)

Category of Assumptions	**Details of Assumptions**
User's Current Level	The users of ANOS are well aware of their scientific and academic level of each research topics.
User's Ultimate Goal	The users of ANOS want to get one of the three kinds of knowledge and resources from ANOS ✓ Skin-Deep Knowledge ✓ Basic Knowledge ✓ Advanced Knowledge

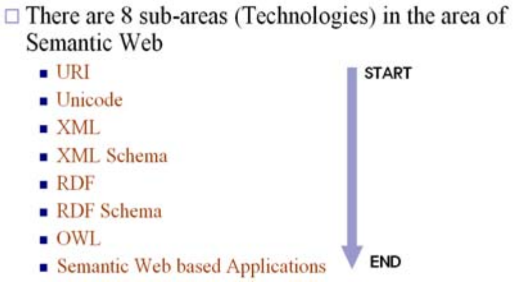
Another assumption is about their goals or expectations which they pursue using ANOS. It is assumed that every user of ANOS keeps their ultimate goals in mind, and the ultimate goal could become one of the three categories. First one is that a user wants to get skin-deep knowledge or superficial one to use for other purposes or to just glimpse the domain. The second goal is that a user wants to start to find their initial resources in earnest. The final and most high-level goal is that a user wants to be more an advanced researcher. As in the users' level assumptions, the underpinning assumption about those goal assumptions is that users are well aware of their final goal, and they can evaluate their goals in terms of three different values.

2.3 Target Domain

We chose to build our knowledge-base about “Semantic Web” research area. The reason why we chose the topic is, of course, its popularity and increasing interests. Almost every computer scientist and even those who are not experts of computer knows well about Semantic Web. If they approach this topic as one scientific research topic in some serious manner and if someone who has not even heard from it before try to study, however, they could hardly find good initial points of research. (As you know, of course, very famous site exists, www.w3.org)

Another reason why we have chosen the topic is that we can find much more resources on the web easily compared to other research topics, which seems natural because “Semantic Web” is a research topic about web itself!

Fig. 1 shows the sub-topics of “Semantic Web” used in the ANOS knowledge-base. Although “Semantic Web” has actually much more sub-topics, we assume that there are only eight sub-topics (technologies) in “Semantic Web” to narrow our scope of research. This figure also shows that the order exists in the sub-topics. When you study a certain topic, you usually first define the composed sub-topics to be studied or you should find the resources containing these composed sub-topics and its details. It is similar to the case when you read a book. A book is composed of several chapters each of which contains rather independent content. In many cases, the author of a book lets you know about the order of reading the entire chapters. Also each chapter contains the information about prerequisite chapters or knowledge needed to master the current chapter. Furthermore, the author says that you could skip certain chapters if you are enough familiar with the contents of the chapter. The reason why the author of a book can let you know about the information is that he or she is already in the level of experts of some scientific areas.



□ There are 8 sub-areas (Technologies) in the area of Semantic Web

- URI
- Unicode
- XML
- XML Schema
- RDF
- RDF Schema
- OWL
- Semantic Web based Applications

Fig. 1. The sub-areas (domains, technologies) of "Semantic Web" assumed in this paper.

Now let's assume that building a knowledge-base about "Semantic Web" is the same as writing a book named "Introduction to Semantic Web." Initially, our experts will have to write the table of contents about "Semantic Web" technologies. There might be so many ways to organize this contents list. One would give some conceptual level of contents list such as "1. What is Web?", "2. What is Semantic?", "3. What is Semantic Web?" and so forth. In this paper, we decided to stick to the specialized sub-areas or technologies when we organized the contents list of our knowledge-base.

Table 2 shows two different tables of contents of the books related to "Semantic Web." As we mentioned before, according to the viewpoint of the authors of the books, the different lists can be generated. In this respect, we can define two kinds of approaches of organizing a table of contents of certain books. One is the ***logical flow-based approach***, and the other is the ***sub-topics-based approach***. By the logical flow-based approach, it means that when the experts of one domain organize the table of contents, they invent their own orders of explanations which could best accomplish their goals of writing, such as giving easy and pedagogical information, providing detailed contents of certain domains, and proposing their new ideas etc. The second approach is that the authors simply list the sub-areas or sub-topics and sort these sub-areas using their own knowledge about dependencies between each sub-topic. For example, an author might think that if someone who wants to know about XML Schema, he should be well aware of XML first. Therefore, he places XML Schema after XML part.

Table 2. Three Kinds of "Table of Contents" of the three different "Semantic Web" related books

A Semantic Web Primer [4]	The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management [5]
1. The Semantic Web Vision	1. What Is the Semantic Web?
2. Structured Web Documents in XML	2. The Business Case for the Semantic Web
3. Describing Web Resources in RDF	3. Understanding XML and Its Impact on the Enterprise
4. Web Ontology Language: OWL	4. Understanding Web Services
5. Logic and Inference: Rules	5. Understanding the Resource Description Framework
6. Applications	6. Understanding the Rest of the Alphabet Soup
7. Ontology Engineering	7. Understanding Taxonomies
8. Conclusion and Outlook	8. Understanding Ontologies
	9. Crafting Your Company's Roadmap to the Semantic Web

Semantic Web Technologies: Trends and Research in Ontology-based Systems [6]
1. Introduction
2. Knowledge Discovery for Ontology Construction
3. Semantic Annotation and Human Language Technology
4. Ontology Evolution
5. Reasoning With Inconsistent Ontologies: Framework, Prototype, and Experiment
6. Ontology Mediation, Merging, and Aligning
7. Ontologyies for Knowledge Management
8. Semantic Information Access
9. Ontology Engineering Methodologies
10. Semantic Web Services – Approaches and Perspectives
11. Applying Semantic Technology to a Digital Library
12. Semantic Web: A Legal Case Study
13. A Semantic Service-Oriented Architecture for the Telecommunications Industry
14. Conclusion and Outlook

In Table 2, two upper contents can be classified as ones created by using the sub-topics based approach, and on the other hand the lower table of contents seems to be made by applying the logical flow-based approach. Note that all the three tables of contents are quite different in spite of talking about the same topic, "Semantic Web." Also, we can detect that there is a common characteristics between the three, in which all the three tables of contents contain the chapter discussing the application of the main topic. Therefore, we accommodate this feature so that our knowledge-base can hold more advanced resources concerning the applied parts of "Semantic Web" as well as the fundamental facts of each sub-topic.

The fundamental reason why we made our table of contents as Fig. 1 is that this approach is more practical and explicit for users to access the resources of our knowledge-base. Users could detect the overall knowledge structure of "Semantic Web," and they also can realize what is familiar and well aware and what is not easily. As a matter of course, according to experts' characteristics, this contents list could be changed. Because of encapsulation between the knowledge-base and application codes of ANOS, we can handle the change effectively only by altering the knowledge-base parts of the entire system. Additional reason is that if the structure and semantics of the sub-topics are too complicated, experts cannot build corresponding fact elements of knowledge-base easily. It will be quite serious when the size of the knowledge-base becomes reasonably large.

3. ANOS (Adaptive Novice Supporter)

3.1 Functions

There are three functions of ANOS. First function is to suggesting beneficial information (resources) related to “Semantic Web” area chosen carefully by experts. These include technical papers, notable textbooks, web pages, and other famous senior researchers or faculties. The second function is to provide users with sub-topics (contents list) of “Semantic Web” as we mentioned previous section.

Finally, ANOS gives an ordering sequence of those sub-topics which users should follow using the pre-ordered table of contents in the ANOS knowledge-base. The application part of ANOS is so simple and flexible that we could replace the underpinning knowledge-base without any difficulty. Also, even while ANOS is at work, we can easily update or revise our knowledge-base.

Table 3. Functions of ANOS

Functions List	**Examples**
Suggesting Useful Information related to one special academic domain	✓ Technical Papers ✓ Fundamental Textbooks ✓ Useful Web sites ✓ Experts in certain sub-areas
Suggesting sub-topics related to the domain	Sub-topics of Semantic Web → OWL, RDF, RDFS, URI, Unicode, XML, XML Schema
Planning the sub-topics in terms of the current user’s academic level or degree and his goal.	Ordering Sequence of Sub-topics highly recommended to users

3.2 Overall Architecture

Fig. 2 depicts the overall structure of ANOS. If users want to get some information adapted to their own situation, they should input some background information about themselves such as their academic or knowledge levels of each sub-domain, and their goals as presented in section 3.2. We will discuss about the user profile information (user context) in detail in section 4.4. Given the user’s profile information and knowledge-base, ANOS generates user’s own plan information, which is the role of the naïve planner. All the access operations needed to generate planning information are performed only through using RDQL. Owing to the absence of reasoning functions in Redland RDF Framework, we implemented the naïve planner by means of just using RDF query. Although initially RDQL seems to have very limited facilities in order to implement the planner, we could detect that it was just sufficient to realize our system.

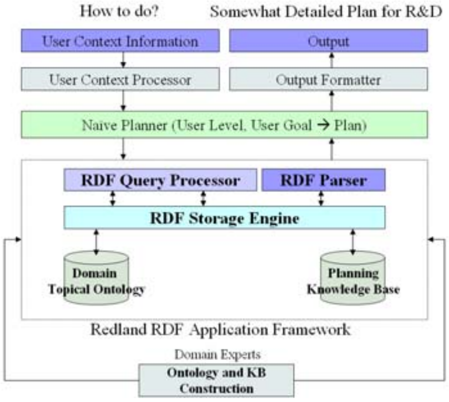


Fig. 2. The Architecture of ANOS

Because of the complete separation of the application codes and the knowledge-base of ANOS, we could easily update or extend our knowledge-base without any difficulty as mentioned before.

3.3 Construction of Ontology and RDF Instance

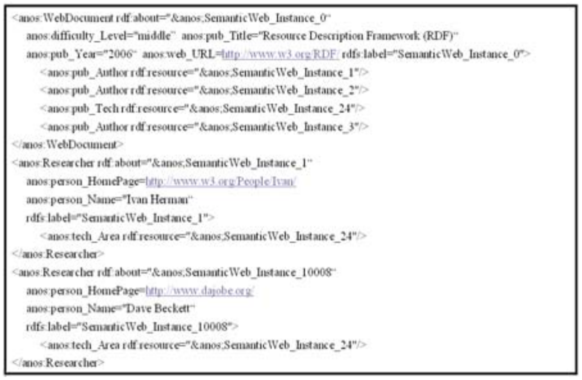
The ontology of ANOS is very simple and concise enough for any experts to express their own expertise as RDF documents. Many people think that ontology should be very expressive and complicate so that it can be used by reasoning engines to infer some knowledge. However, if we were too enthusiastic in building maximally expressive ontology without considering and analyzing the target applications using it, we could suffer from two problems. The first problem is that the ontology is too difficult for experts to use it. As I mentioned before, in many cases, experts do not know well how to express their expertise using ontology. The second one is about the reasoning engines. At this time, we could not take advantage of the full features of FOL (First Order Logic) and DL (Description Logic) when it comes to the conventional reasoning engines published freely. In this context, no matter how expressive the resulting ontology may be and no matter how effectively the ontology may have exploited the features above, the reasoning engines could handle just the fraction of it. It causes the inefficient consequences, which our efforts of building the ontology could become almost useless.

The problems mentioned above lead us to make very simple ontology for ANOS’ knowledge-base. The aim of our ontology is just to define vocabularies and its properties to describe the beneficial information about certain domains. Table 4 shows the brief information about our ontology.

Table 4. ANOS' Ontology

Super Classes	Sub Classes	Slots (Properties)
Person	Researcher	person_HomePage, person_Name, tech_Area
Publication	Book	pub_Title, pub_Author, pub_Tech, pub_Year, difficulty_Level
	Paper	pub_Title, pub_Author, pub_Tech, pub_Year, difficulty_Level, download_Link
	WebDocument	pub_Title, pub_Author, pub_Tech, pub_Year, difficulty_Level, web_URL
Technology		tech_Name, tech_Order

There are three main top-level classes in our ontology, "Person," "Publication," and "Technology." "Person" class has one sub-class named "Researcher." "Publication" class has three sub-classes, "Book," "Paper," and "WebDocument." "Technology" is a top-level class representing the sub-areas or sub-technologies of certain domains. Each class has its own properties (slots) describing the characteristics of the resources declared as the class type. Among them, "difficulty_Level" slot is the indication that shows the level or difficulty of a resource determined by an expert. Rather than using very complicated expressing mechanisms, we just let an expert to make a decision about the difficulty of one resource. This method seems to be very awkward and naïve, but if we can assume that experts could detect the difficulty level of one resource easily, it is probably the most straightforward and effective method.



```
<anos:WebDocument rdf:about="&anos;SemanticWeb_Instance_0"
   anos:difficulty_Level="middle"  anos:pub_Title="Resource Description Framework (RDF)"
   anos:pub_Year="2006"  anos:web_URL=http://www.w3.org/RDF/ rdfs:label="SemanticWeb_Instance_0">
      <anos:pub_Author rdf:resource="&anos;SemanticWeb_Instance_1"/>
      <anos:pub_Author rdf:resource="&anos;SemanticWeb_Instance_2"/>
      <anos:pub_Tech rdf:resource="&anos;SemanticWeb_Instance_24"/>
      <anos:pub_Author rdf:resource="&anos;SemanticWeb_Instance_3"/>
</anos:WebDocument>
<anos:Researcher rdf:about="&anos;SemanticWeb_Instance_1"
   anos:person_HomePage=http://www.w3.org/People/Ivan/
   anos:person_Name="Ivan Herman"
   rdfs:label="SemanticWeb_Instance_1">
      <anos:tech_Area rdf:resource="&anos;SemanticWeb_Instance_24"/>
</anos:Researcher>
<anos:Researcher rdf:about="&anos;SemanticWeb_Instance_10008"
   anos:person_HomePage=http://www.dajobe.org/
   anos:person_Name="Dave Beckett"
   rdfs:label="SemanticWeb_Instance_10008">
      <anos:tech_Area rdf:resource="&anos;SemanticWeb_Instance_24"/>
</anos:Researcher>
```

Fig. 3. One Part of ANOS' RDF document

Fig.3 shows a fraction of our RDF document. In this figure, there is one web document and two researchers who are the authors of this document. Using the native user interface of Protégé, experts do not have to write this RDF document by themselves.

Table 5. Result of Constructing RDF Instances

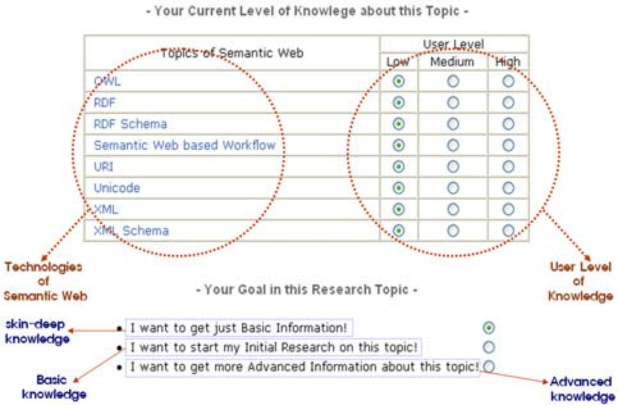
Super Classes	Sub Classes	Number of Instances
Person	Researcher	34
Publication	Book	10
	Paper	10
	WebDocument	39
Technology		8
Total		101

The table above shows the brief information about the number of instances in our RDF document. There are totally 101 instances in the RDF document, and we are still appending more instances to build some sizable knowledge-base about "Semantic Web." The sub-technologies (sub-areas) of "Semantic Web" discussed in section 3.3 are also expressed as resources in the RDF. These play the categorical role in describing the other resources.

3.4 User Profile Information

We have already mentioned that three strong assumptions exist in inventing and implementing ANOS. In these assumptions, there were two assumptions related to the target users of ANOS. The first one is that a user is well aware of his or her own level of knowledge of each sub-topic and furthermore he or she could evaluate the level by specifying one of the three values, "low," "middle," "high." The second assumption is that users usually have their own ultimate goals in mind with ANOS. Also, these goals could be categorized as three different levels of the academic achievements (skin-deep knowledge, basic knowledge, and advanced knowledge).

With these strong assumptions, we organized the user profile information as shown in Fig.5.



- Your Current Level of Knowlege about this Topic -

Topics of Semantic Web	User Level		
	Low	Medium	High
OWL	◉	○	○
RDF	◉	○	○
RDF Schema	◉	○	○
Semantic Web based Workflow	◉	○	○
URI	◉	○	○
Unicode	◉	○	○
XML	◉	○	○
XML Schema	◉	○	○

- Your Goal in this Research Topic -

- I want to get just Basic Information! ◉
- I want to start my Initial Research on this topic! ○
- I want to get more Advanced Information about this topic! ○

Fig. 5. User Profile Information Interface

The upper portion of the figure shows the input boxes in which users can select one of the three different kinds of academic level or knowledge level in each sub-topic. All the sub-topics of “Semantic Web” are automatically extracted from the RDF knowledge-base of ANOS. So if the RDF document is changed, the list of sub-topics is also altered. Users should choose one of the three values, “low,” “middle,” “high,” with regard to their own level of each sub-topic.

The lower part of this figure depicts the selection menu of users’ academic goals. Users should select one of the three goals which they want to achieve in ANOS. “Skin-deep Knowledge” means the elementary and minimum information needed to understand the topic. “Basic Knowledge” means the fundamental and conceptual information which is necessary to conduct the initial research on the topic. “Advanced Knowledge” denotes rather high-level and applied information needed when someone who has more or less deep knowledge about the topic wants to step further in the research.

4. Implementation

4.1 RedLand RDF Application Frameworks

Redland is a general-purpose Resource Description Framework (RDF) library providing APIs to the RDF model and it's concepts. It is also a flexible and efficient implementation of RDF that complements this power and provides high-level interfaces allowing instances of the model to be stored, queried and manipulated in C, Perl, Python, Tcl and other languages. [7]

Before implementing ANOS, we considered three candidate frameworks that can be used in managing our knowledge-base. The first one is “Jena” which is a very famous Java-based RDF/OWL frameworks being used by numerous developers. It provides a powerful reasoning mechanism based on SWRL rules. The second one is “3store” which is a persistent RDF knowledge-base toolkit developed by Steve Harris, AKT Consortium. This package is also known as having rule-based reasoning methods.

However, the above two frameworks have its own disadvantage when we considered them as our backbone system. First, Jena is a Java based application, which has been reported that the performance of it can be degenerated as the size of a knowledge-base becomes larger and larger. Because of the vulnerable point, many applications using “Jena” are also employing other supporting frameworks such as DBMS and file systems. Additionally, “3store” seems to be imperfect when it comes to the usage convenience. It is very difficult to install and operate it without any support of the distributors.

With these reasons, we decided to use RedLand which supports very flexible APIs for managing and querying RDF database. It is a combined C library providing a high-level RDF interface allowing the RDF graph to be parsed from XML, stored, queried and manipulated. RedLand contains three independent sub-packages related to RDF parsing, querying, and storing. Raptor is one sub-package which provides the functions related to RDF parsing. Rasqal supports two different RDF query standards, RDQL and SPARQL. In spite of its compact and efficient structure, one flaw of RedLand is that it does not support the reasoning facilities explicitly. As mentioned before, however, this disadvantage caused little problems because we built a very simple ontology structure and our reasoning mechanism is also simple so that experts could easily generate the knowledge-base.

4.2 Suggestion Algorithm and Service Scenario

Fig.6 shows the suggestion procedure of ANOS using its knowledge-base and user profile information.

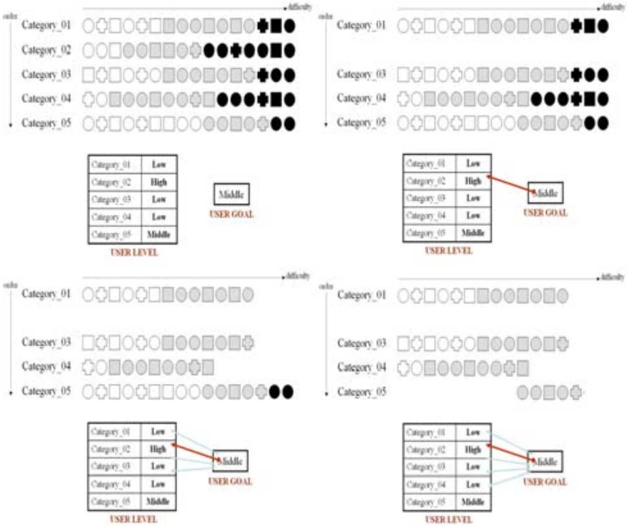
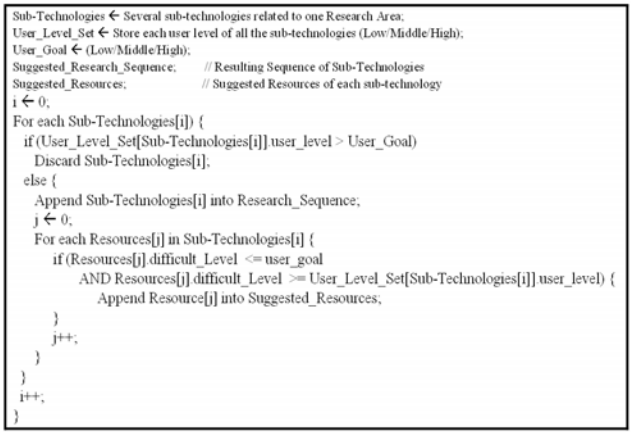


Fig. 6. Suggestion Procedure, the sequence is top-left, top-right, bottom-left, bottom-right.

Each resource in our knowledge-base is depicted by three different shapes of figures with respect to the type of each resource such as papers, web documents, and books. Category means the sub-areas or sub-technologies of one certain topic. Therefore, every resource of our knowledge-base belongs to a category. Experts have specified the difficulty level of each resource, which is expressed by the color of each figure. In the Fig.6, there are five categories representing sub-areas, and each category contains exactly the same number of resources for simplicity. Now, let’s suppose that a user’s current level and his goal are inputted as shown in the tables of each figure. This table shows that the goal of this user is “middle,” which means that he wants to get some basic information needed to conduct initial research on the current topic. Given this user profile information, ANOS will filter both categories and resources. As shown in the top-right figure, because the user level of category_02 is “High,” and the goal of this user is “Middle,” this user does not have to stick to all the resources in that category. So ANOS discards this category entirely. Given the remaining resources, ANOS now will perform filtering process on them. In category_01, it is also not necessary for the user to see the most difficult resources depicted as black figures because although the user level of this category is “Low,” the goal of this user is just “Middle.” Finally, in the category_05, the user’s level of this category is “Middle,” and the user’s goal is also “Middle.” In this case, the user does not have to be given the easy resources as well as the difficult ones.

Fig.7 depicts the pseudo code of the above algorithm. As you may have captured previously, it has a very simple logical structure.



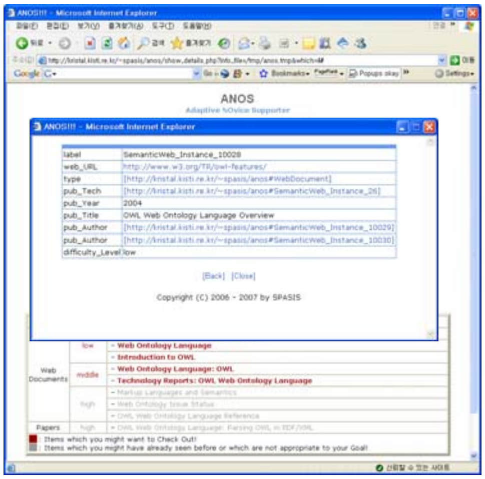
```
Sub-Technologies ← Several sub-technologies related to one Research Area;
User_Level_Set ← Store each user level of all the sub-technologies (Low/Middle/High);
User_Goal ← (Low/Middle/High);
Suggested_Research_Sequence;       // Resulting Sequence of Sub-Technologies
Suggested_Resources;               // Suggested Resources of each sub-technology
i ← 0;
For each Sub-Technologies[i]) {
  if (User_Level_Set[Sub-Technologies[i]].user_level > User_Goal)
    Discard Sub-Technologies[i];
  else {
    Append Sub-Technologies[i] into Research_Sequence;
    j ← 0;
    For each Resources[j] in Sub-Technologies[i] {
      if (Resources[j].difficult_Level  <= user_goal
          AND Resources[j].difficult_Level  >= User_Level_Set[Sub-Technologies[i]].user_level) {
            Append Resource[j] into Suggested_Resources;
      }
      j++;
    }
  }
  i++;
}
```

Fig. 7. The Suggestion Algorithm Pseudo Code

Based on this algorithm, ANOS provides a suggesting service which is composed of three scenario elements, domain selection, user input, and ANOS output. Table 6 shows the details of the service scenario of ANOS.

Table 6. Service Scenario of ANOS

Service Components	Detailed Scenario
Domain Selection	1. A user chooses one of the scientific topics provided by ANOS.
	2. ANOS analyzes a RDF document and extracts sub-topics (sub-technologies) of this topic.
User Input	1. A user should input his or her profile that contains the information about his or her degree of academic specialty of each sub-topic.
	2. A user should input his or her ultimate goal level in the ANOS system.
	3. A user should select one of the sub-topics provided by ANOS to narrow his or her ultimate goal of research
Generation of Adaptive Information	1. ANOS will present scheduling information (a sequence of research process composed of various kinds of resources such as technical paper, web documents and books.)


ANOS!!! - Microsoft Internet Explorer
ANOS!!! - Microsoft Internet Explorer

ANOS
Adaptive NOvice Supporter

label	SemanticWeb_Instance_10028
web_URL	http://www.w3.org/TR/owl-features/
type	[http://kristal.kisti.re.kr/~spasis/anos#WebDocument]
pub_Tech	[http://kristal.kisti.re.kr/~spasis/anos#SemanticWeb_Instance_26]
pub_Year	2004
pub_Title	OWL Web Ontology Language Overview
pub_Author	[http://kristal.kisti.re.kr/~spasis/anos#SemanticWeb_Instance_10029]
pub_Author	[http://kristal.kisti.re.kr/~spasis/anos#SemanticWeb_Instance_10030]
difficulty_Level	low

[Back] [Close]

Copyright (C) 2006 - 2007 by SPASIS

Web Documents	low	- Web Ontology Language
		- Introduction to OWL
	middle	- Web Ontology Language: OWL
		- Technology Reports: OWL Web Ontology Language
	high	- Markup Languages and Semantics
		- Web Ontology Issue Status
		- OWL Web Ontology Language Reference
Papers	high	- OWL Web Ontology Language: Parsing OWL in RDF/XML

🟥 : Items which you might want to Check Out!
▨ : Items which you might have already seen before or which are not appropriate to your Goal!

Fig. 8. The Service of ANOS, Domain Selection, User Input, and Generation of Output

In Fig.8, the captured images of running ANOS are given. This system implemented as a web service by adopting simplified user interface. Given the users’ profile information, ANOS generates a research order which should be followed by the users and the detailed resource information which is well-organized list of constituting resources. In the suggestion result image of Fig.8, you can see that the colors of resources vary from each other. The red-colored resources are the ones which the current user might want to study or read. The light grey-colored resources are what the current user can skip.

If you click one of the resources in the list, you can see the detailed meta-information about the selected resource. According to the type of the resource, different items will be given to the user; for web resources, additional web URL information will be shown, for papers, we can detect the download link information in which you can download.

5. Some Limitations and Further Research

As you might have already captured, there are many limitations of ANOS when we consider this system as an expert system. Out of these limitations, the most serious one is the function of generating research orders. ANOS should be given a pre-defined research order composed of sub-topics (sub-technologies, areas) made by experts. Then, it filters inappropriate sub-topics comparing with user profile information. This process is not exactly what an automatic plan generator based on AI planning algorithms does. It just discards certain sub-topics whose user level values are higher than the goal level values.

Therefore, we are just using the ordering information in the ANOS' knowledge-base specified by domain experts.

How can ANOS automatically generate a research order without any explicit ordering information? There can be two kinds of approaches in achieving that, machine learning based approach and expert systems of conventional AI. In machine learning approaches, there are two reasonable methodologies which can be used to generate planning information. These are both reinforcement learning and Bayesian networks. The common feature of the two methods is that they can capture the stochastic dependencies between two objects. If we assume these objects are the sub-topics of our knowledge-base, and if we have sufficient training data concerning the user's behavior and goals in doing research in a certain domain, we can find out the ordering data as well as the inter-relationships between two sub-areas.

The second major limitation is that we cannot directly compare the user level value with the values of user's goal and the difficulty level of each resource specified by domain experts. In other words, the values we have defined ("low," "middle," "high") cannot represent the actual meanings of our purpose. For example, when we compare the values of a user level and those of a user's goal and difficulty level of each resource, we cannot guarantee that the sources of the criteria of the three values are the same. This problem is caused by the absence of the objectivity in each value. So the experts' evaluation of each resource is not consistent with the users' evaluation of their own academic levels. Consequently, it could cause severe mismatch between users' expectation and ANOS' outputs.

How can we solve this problem? We propose one approach to cover this flaw. Before starting ANOS, an intensive evaluation test should be performed by users. The problems of the evaluation test could be generated by domain experts based on the same criteria used in the knowledge-base when they were specifying the difficulty level of each resource. The result of this test could be the canonical user level which can be compared directly with the difficulty level of each resource. It may not be so difficult for domain experts to make the test sheet. According to the result of the test, ANOS could estimate and compare the current level of users in each sub-topic more appropriately.

6. Concluding Remarks

In this paper, we implemented a very simple plan generator which can help the novice researchers do their research efficiently. We have assumed that we should not make domain experts annoying when building the knowledge-base, and at the same time, we could provide users with the beneficial information made by them using very simple mechanisms. This is the main purpose of implementing ANOS (Adaptive Novice Supporter).

Because of the simplicity and naïve approach of ANOS implementation, there are many limitations of our system. One of them is the inability to generate planning information automatically, and the other is the absence of the underpinning criteria of some values in our knowledge-base. We proposed some approaches to solve these two problems. Based upon the proposed techniques, we will have to enhance our system.

7. References

[1] Dave Beckett, Dave Beckett's Resource Description Framework (RDF) Resource Guide, www.planetrdf.com/guide/

[2] Expert Systems, www.aaai.org/AITopics/html/expert.html

[3] Expert System, http://en.wikipedia.org/wiki/Expert_system

[4] John Davies, Rudi Studer, and Paul Warren, Semantic Web Technologies: Trends and Research in Ontology-based Systems, John Wiley & Sons (July 11, 2006)

[5] Grigoris Antoniou, Frank van Harmelen, A Semantic Web Primer (Cooperative Information Systems) (Hardcover), The MIT Press (April 1, 2004)

[6] Michael C. Daconta, Leo J. Obrst, Kevin Smith, The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management (Paperback), Wiley (May 30, 2003)

[7] David Beckett, The Design and Implementation of the Redland RDF Application Framework, Tenth International World Wide Web Conference, May 1-5, 2001

[8] Resource Description Framework (RDF) / W3C Semantic Web Activity, www.w3.org/RDF/

[9] XML.com: What Is RDF, www.xml.com/pub/a/2001/01/24/rdf.html

[10] Planet RDF, www.planetrdf.com/

[11] Redland RDF Application Framework, http://librdf.org/

[12] RDQL - A Query Language for RDF, www.w3.org/Submission/RDQL/

[13] Jena Tutorial – RDQL, jena.sourceforge.net/tutorial/RDQL/

[14] Tom Gruber, The Pragmatics of Ontology as Language, Contract, and Content, Bio-Ontologies 2000 Workshop, 2000

[15] Tutorial on developing OWL ontology using the Protégé-OWL editor, Protégé-OWL Tutorial

[16] http://jena.sourceforge.net/, Jena Developer Page

[17] http://sourceforge.net/projects/threestore/, 3store

[18] http://www.rdfdata.org, Storage of RDF Documents