

## Association for Information Systems AIS Electronic Library (AISeL)

---

PACIS 2005 Proceedings

Pacific Asia Conference on Information Systems  
(PACIS)

---

December 2005

# A Web Services Approach to Model Management in DSS

Ou Liu

*University of Hong Kong*

Jian Ma

*University of Hong Kong*

Follow this and additional works at: <http://aisel.aisnet.org/pacis2005>

---

### Recommended Citation

Liu, Ou and Ma, Jian, "A Web Services Approach to Model Management in DSS" (2005). *PACIS 2005 Proceedings*. 31.  
<http://aisel.aisnet.org/pacis2005/31>

This material is brought to you by the Pacific Asia Conference on Information Systems (PACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in PACIS 2005 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# A Web Services Approach to Model Management in DSS

Ou Liu

Department of Information Systems  
City University of Hong Kong, Hong Kong  
isliuou@cityu.edu.hk

Jian Ma

Department of Information Systems  
City University of Hong Kong, Hong Kong  
isjian@cityu.edu.hk

## Abstract

*Model management is an important research topic in the Decision Support Systems. Current research focuses on methods and theories to improve shareability, and effective utilization of decision models in an organization. This paper presents a Web services approach to the representation and design of model base in organizational model management. Ontology is used for the model base design, and “decision ontology” is proposed for model management. XML is used for the exchangeable representation of decision ontology. The proposed methods and theories are used in the construction of a DSS for R&D project selection at the National Natural Science Foundation of China (NSFC). The proposed approach makes use of the Web services technologies, it allows the sharing and reuse of decision models in the Internet environment.*

**Keywords:** Model Management, Ontology, XML, Web Services

## 1. Introduction

A DSS is an information system that supports decision-making process of all levels of management from top executives to line managers. Model management is very important for the creation, storage, manipulation and effective utilization of decision models in DSS (Blanning 1993). Because of the always-changing decision environment and non-professionalism of decision makers (Wu 2000), ease to use and reuse decision models is a basic requirement for model management (Marakas 2002).

Several model management frameworks have been classified by the model representation method (Tsai 2001), e.g., Relational Model (Blanning 1987), Structured Modeling (Geoffrion 1996), OO Modeling (Lenard 1993; Ma 1997). Besides representation problem, some of them also address the problem of model base design, e.g., Structured Modeling builds up a model base by construction of elemental structure, generic structure, and modular structure; OO Modeling design a model base according to the inheritance hierarchy.

Currently, with the support of Internet technologies, DSS can be used for decision-making tasks across multiple departments inside an organization (Huh et al. 2004; Tian et al.). In these circumstances, decision models are required to be represented in a standard and unified format so that they can be interchanged and interoperated among different systems or different parts of a system (Dolk 2000; Schneeweiss 2003). It is also important for decision models to be understood, shared and reused consistently and correctly by different users (Huh et al. 2004; Tian et al. 2002a). However, there is still lack of design theory to support model representation and sharing in an organization (Krishnan et al. 2000; Tsai 2001). Also, previous framework can hardly be mapped to an implementation that supports collaborative organizational model utilization (Huh et al. 2004; Krishnan et al. 2000).

Recent advancement in artificial intelligence and software engineering has brought new chances to improve model management in organizational environment. Ontology engineering is proposed to achieve knowledge sharing and reuse in distributed environments (Elst et al. 2002; Noy et al. 2001). An ontology refers to an engineering artifact, consisting of a specific vocabulary used to describe a certain reality, and a set of explicit assumptions about the intended meaning of the vocabulary (Elst et al. 2002; Fensel 2001). It is used for sharing a common understanding of the information structure among people or software agents and enabling the reuse of domain knowledge (Fensel 2001). Since the specification of decision models are a specific kind of knowledge (Suh et al. 1995), and it is important to structure and define decision-related terminology systematically in model management (Geoffrion 1996), ontology engineering will be useful in model management (Tian et al. 2002b).

Another trend in software engineering for meta-data/knowledge/component sharing and reuse in implementation level is to apply XML-based technologies (Klein et al. 2001; McIlraith et al. 2003). For example, XML is used for open information interchange (Kim 2001; Klein et al. 2001), and Web services provide a standard means to share and interact with software modules/components over the Internet (Gibbins et al. 2004; IBM 2000; Wang et al. 2004).

In this paper, we make use of XML-based technologies and ontology engineering in efficient model management. Their application is not the same as in software engineering because the context is different in model management. Model management in DSS is to make managers (usually non-computer-experts) easy to use and construct models in an always-changing environment, while software components are relatively stable and used by computer experts.

This paper presents a Web services framework for model management. Inspired by the research in artificial intelligence community, decision ontology is designed for representing modeling knowledge in model management. An XML-based language, i.e. Decision Ontology Markup Language (DOML), is designed and used to specify the decision ontology. DOML can be processed by XML parsers for exchanging information with other systems over the Internet. A Web services implementation architecture is also proposed to build the model management system. Thus the collaborative parties can connect and access the decision models of the DSS through a standard interface.

The proposed methods and theories have been used for the development of the internet-based science information systems (ISIS, <http://isis.nsf.gov.cn>). ISIS includes a DSS component for R&D project selection, which is used in the National Natural Science Foundation of China (NSFC). As the largest government-funding agency in China, NSFC received more than 43,000 proposals from over 1,400 universities and research institutions nationwide in China before end of March in 2004. Over 200,000 external review forms were assigned and processed electronically through ISIS three months after proposal submission deadline in March. The decision support components of ISIS are used to facilitate the overall decision-making process from proposal submission, selection of external reviewers, peer evaluation and panel evaluation, to aggregation of the decision outcomes. Many decision models are designed to support these tasks (Tian et al. 2002a). These decision models and the domain knowledge are represented in decision ontology and can be reused and shared by users with different knowledge backgrounds.

The rest of this paper is organized as follows. The proposed model management framework is presented in section 2. The application of the proposed framework at NSFC is illustrated in section 3. The last section is the conclusion.

## 2. The Proposed Model Management Framework

The proposed model management framework is based on ontology engineering and Web services technology.

### 2.1 Decision Ontology and Web Services for Model Management

Ontology engineering is used to process the terminologies/concepts in decision-making processes (Noy et al. 2001; Tian et al. 2002b). In a decision-making process, decision-makers usually invent and use a set of terminologies/concepts to talk about decision situations, decision models, and how the models can be applied in the situations. Defining decision-related terminology systematically is important in model management (Geoffrion 1996).

In this paper, decision ontology is designed and used to conceptualize the knowledge for decision-making processes. It includes the knowledge of the decision models as well as the meta-knowledge about them. Decision ontology can be divided into two parts, i.e. domain ontology and modeling ontology. The latter is further divided into the following three parts: the domain-related variables (*decision variables*), the domain-independent mathematical models (*model types*) and how these models can be applied in the domains (*model templates*). XML (Nambiar et al. 2002) also is used to represent the decision ontology in an Internet environment. The main reasons for applying decision ontology into model management in DSS are:

- 1) Ontology technology can enhance the sharability, reusability and interoperability of the decision models in model management of DSS.
- 2) Decision ontology extends the ideas in previous model management frameworks such as the modeling-in-the-large (Muhanna et al. 1994), concept definitional system for model management (Geoffrion 1996), and a domain-world for models (Hong et al. 1993), which intends to achieve definitional systems of concepts to describe models and their applications at a conceptual level. By clarifying the relationship among decision variables, model types and model templates, it also helps to achieves model schema/instance separation and model interface/solver separation (Tsai 2001).
- 3) XML-based representation of the decision ontology enables the sharing of decision-related knowledge over the Internet. It also provides a simple and standard format for storing and exchanging Web data and knowledge.

Web services technology is used to implement the decision models (e.g. model solver) at the implementation level. It involves a set of protocols and standards for Web implementation (Curbera et al. 2002; IBM 2000). Web services supports the communications between platform and language independent applications. Its implementation architecture allows decision models to be reused and interacted from anywhere in the Internet.

Figure 1 presents a model management framework that supports different levels of abstraction. At specification level, Decision Ontology is designed to describe the model specification; and an XML-based language, i.e. Decision Ontology Markup Language (DOML/XML), is designed to represent the Decision Ontology. At implementation level, a Web services-based architecture is proposed for implementation of model bases in DSS.

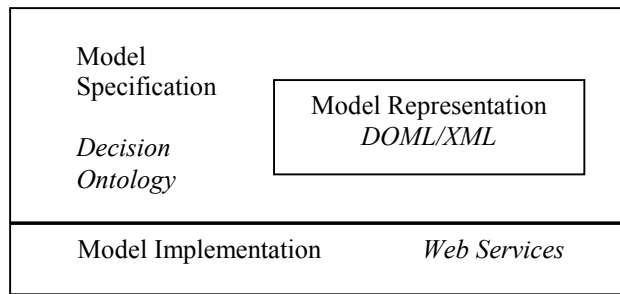


Fig. 1: Levels of abstraction in model management

## 2.2 Ontology-based Architecture for Model Management

The conceptual architecture of the proposed model management system is shown in Figure 2 (Liu et al. 2004). It contains a decision ontology, an algorithm library, an adapter and a user interface.

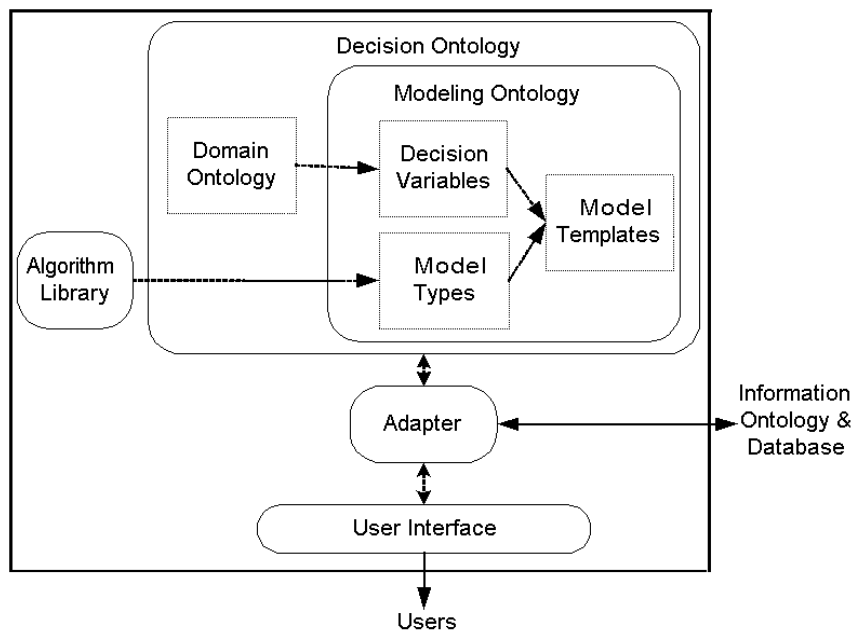


Figure 2. The Conceptual Architecture of the MMS (Liu et al. 2004)

The decision ontology conceptualizes the knowledge and meta-knowledge of decision models. It provides the knowledge related to decision models, rather than the model instances and algorithms for model solving. To instantiate and solve models, firstly, the adapter has to retrieve the model templates from decision ontology and data from the information ontology and database component. The adapter then combines them to formulate model instances, and call on the algorithms from the algorithm library to solve these model instances.

The decision ontology is the main component of the architecture. It consists of two parts: domain ontology and modeling ontology. The domain ontology refers to the terminology for decision-making, defining the concepts and terms used in the decision-making processes of a domain. It mainly includes the terminology from the information ontology (data schema), and some collective concepts such as *Proposals* (meaning a set of proposals). The modeling ontology is further divided into the following three categories of knowledge:

- Domain-related variables (*decision variables*). A decision variable usually appears in the form of a feature of a domain concept specified with a dimension and a unit. It can be instantiated with a mathematical value.

- Domain-independent mathematical models (*model types*). A model type is an interface of some algorithm stored in computers (Mannino et al., 1990; Hong et al., 1993, 1995). It specifies a relation among relevant *mathematical variables*.
- And the application of model types in the domains (*model templates*). Model templates describe how model types can be applied in specific decision-making tasks (Hong et al., 1993, 1995). To obtain a model template, model types are instantiated using *decision variables*.

Decision ontology can encapsulate the semantic information of decision models for models management. It can be viewed as the generalization of the constructs proposed in the structured model of structured modeling (Geoffrion 1996) and real-world model of the unified modeling language approach (Hong et al. 1993).

### 2.3 An Illustrate Example

An example in R&D project selection is shown to illustrate how model types, model templates and decision variables are defined and related to each other.

In R&D project selection in NSFC, when assigning external reviewers to proposals, we need to do a matching between proposals and reviewers according to the research disciplines (Tian et al. 2002a). In NSFC, each of the proposal and external reviewer are required to declare two keywords to indicate the discipline areas it belongs to. NSFC maintains a dictionary of keywords that forms a tree structure. The closer a key word node is to the root, the larger discipline area it represents. For example, key “A010103” stands for “Geometry”, and “A01010302” for “Algebraic Geometry”. The model template *ProposalsExpertsMatching* is design to compare the keyword lists of the research disciplines of proposals and experts, and find out how well they match in pairs. Its usage is like

*ProposalsExpertsMatching (ProposalsKeywordList, ExpertsKeywordList, MatchingResult)*

where *ProposalsKeywordList*, *ExpertsKeywordList* are input decision variables and *MatchingResult* is the output.

The model template applies the model type *FuzzyMatchingModel* by binding its parameters with the keyword lists of the proposals and experts. The model type, i.e. the interface of the model solver/algorithm (Tian et al. 2002a) is like:

*Double[][] FuzzyMatchingModel (String[][], String[][]).*

It compares two lists of words and return a matrix of degrees of how well the elements of the two lists match with each other. Notice that model types do not provide algorithms for model solving, but call the algorithms from the algorithm library.

In our definition, as a decision variable, the set of keyword lists of the proposals is defined as

*ProposalsKeywordList = <ProposalSequence, KeywordList, nil>*

The “nil” here implies a decision variable without unit. Other decision variables can be defined based on domain ontology similarly.

The model type for keyword matching is defined as

*FuzzyMatchingModel = <{s1, s2}, {m}, {s1, s2 □ List, m □ Matrix} >*,  
which specifies the input and output as well as their domain.

The model template is defined as

*ProposalsExpertsMatching = <FuzzyMatchingModel, Bx, By >*,

which make use of the model type “*FuzzyMatchingModel*”. One binding example in **Bx** is {*s1, ProposalsKeywordList* }, which binds *s1* in the model type with the decision variable “*ProposalsKeywordList*”.

#### 2.4 XML-based Representation of Decision Ontology

XML-based Decision Ontology Markup Language (DOML) is used to describe the specification of the decision ontology in this paper. DOML is formally defined by XML Schema, which defines the structure, content and semantics of XML documents. A fragment of the definition is shown in Fig. 3, which defines the domain concepts in the decision ontology.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="Concept" type="ConceptType"/>
  <xsd:complexType name="ConceptType">
    <xsd:sequence>
      <xsd:element name="Attr" type="AttrType" minOccurs="0"
maxOccurs="unbounded"/>
      <xsd:element name="Rel" type="RelType" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:string"/>
    <xsd:attribute name="parent" type="xsd:string"/>
  </xsd:complexType>
  <xsd:complexType name="AttrType">
    <xsd:attribute name="id" type="xsd:string"/>
    <xsd:attribute name="range" type="xsd:string"/>
  </xsd:complexType>
  <xsd:complexType name="RelType">
    <xsd:attribute name="id" type="xsd:string"/>
    <xsd:attribute name="target" type="xsd:string"/>
  </xsd:complexType>
  ...
</xsd:schema>
```

Fig.3 DOML defined by XML Schema

This fragment defines domain concepts and their attributes as well as the relations among them. “Concept” is an instance of “ConceptType”, which has zero to many attributes (“Attr”) and relations (“Rel”). Each of the concepts, attributes and relations has a unique “id” to identify itself. Besides, an attribute has a “range” to specify its data type. A concept can have relations with another one. It can inherit all the attributes and relations from its parent.

Using DOML, we can firstly define the domain ontology in the decision ontology. We use the case of R&D project selection to illustrate it. In Fig.4, one part of the concept hierarchy in the domain ontology is shown. The concept “ExternalMember” inherit from the concept “Person” all attributes and relations such as “FName”, “LName”; besides, “ExternalMember” has its own ones defined, such as the relation “BelongTo” which targeting another concept named “Organization”. And the concept “Investigator” inherits all those attributes and relations from “ExternalMember” also with some unique ones defined.

```

<Concept id="Person">
  <Attr id="FName" range="String"/>
  <Attr id="LName" range="String"/>
  <Attr id="Age" range="Int"/>
  <Attr id="Address" range="String"/>
  ...
</Concept>

<Concept id="ExternalMember" parent="Person">
  <Rel id="BelongTo" target="Organization"/>
  ...
</Concept>

<Concept id="Investigator" parent="ExternalMember">
  <Attr id="Bibliography" range="Text"/>
  ...
  <Rel id="HeadOf" target="Project"/>
  ...
</Concept>

```

Fig.4. Domain concept hierarchy defined in DOML

Based on the defined domain ontology, we can define decision variables and further the whole modeling ontology in Fig. 2. As stated before, model types define domain-independent models at the symbolic level, i.e., the interfaces of algorithms (the model solvers). And model templates describe how model types can be applied in specific decision tasks, by combining model types with some domain concepts and their attributes. In the following, the formal definitions of model type and model template are interpreted with DOML.

The sample model illustrated in last section can be described in DOML as Fig.5. We first define decision variables. One name “ProposalsKeywordList” is shown in Fig 5, referring to the attribute “keywordList” of the collective concept “ProposalSequence” in domain ontology. Then, the model type defines the access interface including input and output. The model template has the model type as its base model type, and binds mathematic variables in the model type with decision variables. A binding is shown linking the input “s1” to the decision variable “ProposalsKeywordList”.



```

<DecisionVar id="ProposalsKeywordList">
    <Concept id="ProposalSequence"/>
    <Attr id="keywordList"/>
</DecisionVar>
...

<ModelType id="FuzzyMatchingModel">
    <Input id="s1" range="List"/>
    <Input id="s2" range="List"/>
    <Output id="m" range="Matrix"/>
</ModelType>

<ModelTpl id="ProposalsExpertsMatching">
    <BaseType id="FuzzyMatchingModel"/>
    <binding var="s1">
        <DecisionVar id="ProposalsKeywordList">
    </binding>
    ...
</ModelTpl>

```

Fig.5. A Model Type and a Model Template defined in DOML

Model templates just describe the application of model types in certain situations, and they are not model instances, which need data from the information ontology and database. To instantiate models, the adapter (in Fig. 2) has to retrieve instance data from the information ontology and database component. And then the adapter will validate them and transform them into concept instances in DOML syntax, using the domain concepts as their data schema. Finally, the adapter combines the concept instances with model templates from the decision ontology to formulate model instances.

In a summary of this section, we have defined a Decision Ontology Markup Language (DOML) based on XML Schema. Using DOML, we can define domain concepts, decision variables, models (model types and model templates), as well as their instances.

### **2.5 Web Service-based Implementation for Model Management**

We have built up a foundation for model management by using DOML to represent all the decision resources, such as data and models, into decision ontology. The XML syntax of DOML makes it easy to share and interchange these resources, and also makes a family of XML-based protocols and standards available for processing the ontology and implementing model management systems. Here, Web services technology, an XML-based technology involving a set of protocols and standards, is applied for the implementation of model management systems, regarding each model as a Web service.

In the normal working mechanism, Web services are accessed/ interacted by users or software agents using SOAP (Simple Object Access Protocol) (Walsh 2002) messages. The functional interfaces of Web services are described with WSDL (Web Services Description Language) (Walsh 2002). We can retrieve the WSDL description that the needed service complies with, so as to learn how to interact with it using SOAP messages. Then we can send SOAP/XML messages by HTTP packages to interact with the Web service.

The Web service-based model management system architecture we proposed is shown in Fig.6. Each model (in fact, a model solver) is implemented as a Web service. Its WSDL description document, carrying the information of the interface of the model solver, can be easily transformed into an XML-based model type definition in Decision Ontology. The transformation is done by an XSL (Extensible Stylesheet Language) (Harold 2001) processor, which can transform any XML document into another (XML) document according to an XSL file.

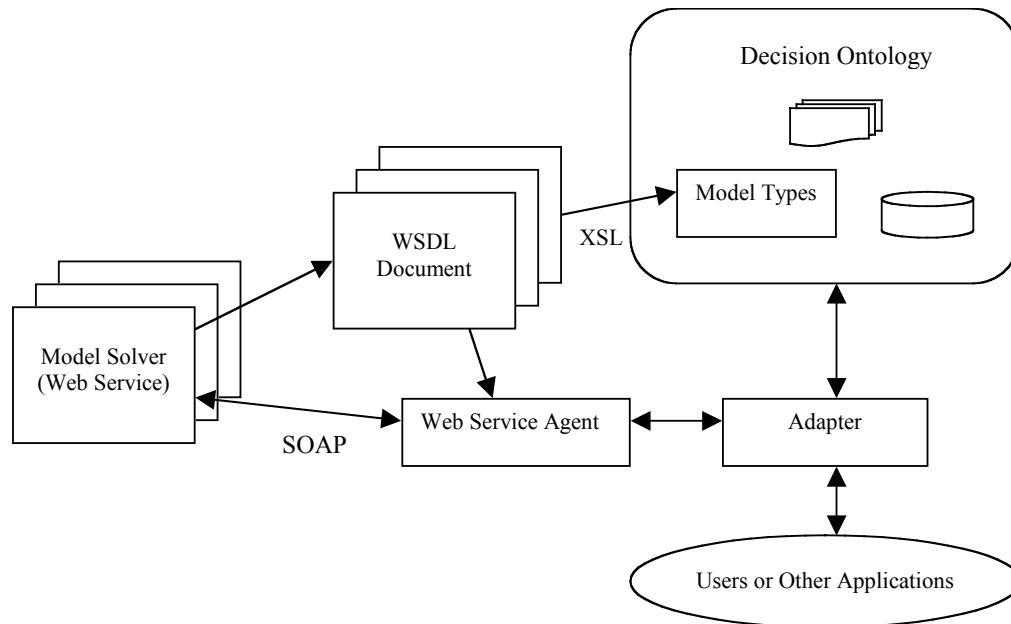


Fig.6. The proposed Web service-based system architecture for model management

In order to access those Web services (model solvers), a Web service agent is needed to take all the tasks of connecting them according to their WSDL description. To do this, the agent firstly achieves the WSDL document of the requested model, from which it learns how to create a correct SOAP message for interacting with this Web service. Then, it creates and sends/receives SOAP messages according to the requirements from outside (through the adapter in Fig.6), so as to accomplish the tasks. The adapter is still the console (like a broker) which accesses and manages model solvers according to Decision Ontology, when requests come from users or other application.

The advantages of the proposed Web service-based system architecture lie in the following:

- Firstly, Web service is open, platform-independent, and interoperable. This newly emergent technology involves a set of international standards supported by many major industry vendors with many products developed or under development. This makes the proposed model management framework very practicable.
- Secondly, with models implemented as Web services, it provides a standard means for users and other software systems (e.g., other models, agents, or even outside systems) to interact with decision models over the Internet. Some cases will be shown in the application in next section.
- Finally, Web service technology well fits the above framework for model base design and model representation: it is an XML-based technology consistent with the proposed DOML; the WSDL documents of models are easily matched to model types in Decision Ontology; and in fact, Fig.6 shows a detailed implementation of Fig.2.

### 3. Applications at NSFC

The proposed model management framework has been applied for R&D project selection at NSFC in Chinese Mainland. Some design aspects have been involved in previous sections. It in fact includes three categories of interactive systems (Fig.7): (1) the ISIS (Internet-based Science Information System) for National Natural Science Foundation of China (NSFC) (<http://isis.nsf.gov.cn>, English version can be found at <http://isis.nsf.gov.cn/default.asp?language=E>), which facilitates the overall decision process of R&D project selection involving many different categories of users all over the country; (2) a Web service-based model management system named “Onto-MMS” for managing the decision models used in ISIS (in fact, it can be regarded as a subsystem of ISIS); (3) many IRISs (Internet-based Research Information Systems, an Example is <http://www.cityu.edu.hk/cityu/research/iris.htm>) in universities, each of which manages the R&D project information in a single university, sums up and submits the information to NSFC, and can interact with decision models in ISIS.

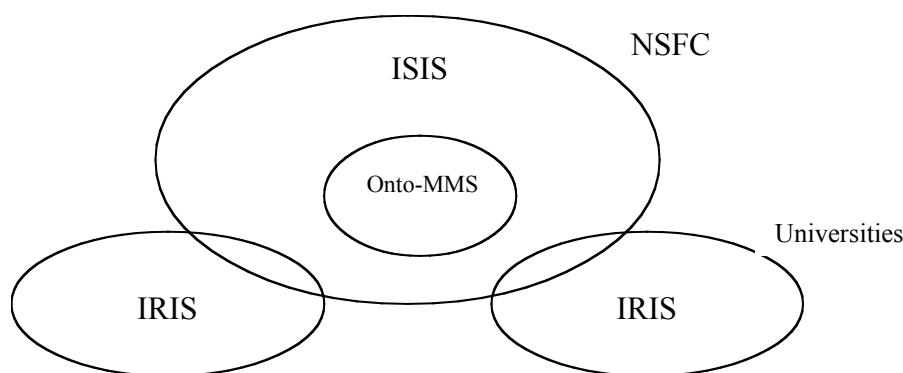


Fig.7. Official Systems for R&D project selection in Chinese Mainland

There are two major data sources: *Person* for human resource data and *Proposal* for proposal data. Human resource data consists of those for *Internal User*, *External User*, and *Applicant/Principal Investigator*. *Internal User* consists of *Top Manager*, *Department Manager*, and *Division Manager*. *External User* consists of *External Reviewer* and *Panel Expert*. Every *Person* belongs to an *Organization*. An *Applicant / Principal Investigator* may submit one or more proposals through his/her affiliated *Organization* to the funding agency, and a *Proposal* may include one or more *Applicant/Principal Investigator*. A *Proposal* may be reviewed by *External Reviewers* and *Panel Experts*.

Table 1. Decision Ontology for R&D Project Selection

<b>Concepts in Domain Ontology:</b> Proposal Reviewer ProposalSequence ReviewerSequence ...	<b>Decision Variables:</b> ProposalReviewerGrade ProposalPanelGrade ... ProposalsKeywordList ...
<b>Model Types:</b> FuzzyMatchingModel JobAssignmentModel MCDMAggregationModel PreferenceConvertModel SubjectiveObjectiveAggregateModel ...	<b>Model Templates:</b> ProposalsExpertsMatching ProposalsExpertsAssignment ProposalEvaluationAggregation PanelExpertsEvaluationAggregation DistributionAnalysisforFinalDecision ...

Table 1 shows part of the decision ontology content for R&D project selection. It includes concepts in domain ontology and decision variables, model types and model templates in modeling ontology. The concepts in domain ontology are defined based on the data schema. Decision variables are defined based on domain ontology. Some of the model types and model templates used in the system are also shown in Table 1.

In the following, we will more focus on the demonstration of the “Onto-MMS”. For security reason, it runs inside an Intranet in client/server mode. Its functions include:

- Edit the domain ontology: users can add new concepts, modify or delete an existing one, and re-organize concepts.
- Maintain decision variables: decision variables can be added, modified and related to information in the domain ontology.
- Maintain model types: existing model types can be modified and deleted. New ones can be added.
- Maintain model templates: the binding of model types with concepts can be changed using this function. In other word, this function enables users to define their usages of model types for their own decision situations.
- Retrieve model templates: users can search required model templates semantically using concepts.

To manage concepts in the domain ontology, the system maintains a term tree that represents the concept hierarchy. There is a node *root* in the tree which is the start point of all concepts. Each of other nodes is the sub-concept of its parent (upper-level node). Concepts can be edited by performing operations on them such as “find”, “add”, “delete” and “edit”.

The system supports operations on model types and model templates including “find” (by name), “add” and “delete”, which are similar to those on concepts. Figure 8 illustrates how to add a new model template called *ProposalsExpertsMatching*. First, the domain-independent model type *FuzzyMatchingModel* is chosen. Then, the correspondent input and output variables of the model type are shown in the first column of the mapping table. Next, concepts and their attributes are selected and mapped to the variables, in the format of *concept\_name.attribute\_name* shown in the second column in the mapping table (e.g., *ProposalSequence.keywordList* in Fig.8). If a correspondent decision variable has been defined before, its name will be shown in the third column; otherwise, user can name it there. Finally, a domain-dependent model template is built.

The systems also support flexible ways to retrieve model templates using concepts. Fig.9 is a screenshot of searching for model templates. First, decision variables are specified as the input or output of the expected model templates, which then form the searching conditions. If users cannot remember the decision variable name, they can select the concepts and attributes from domain ontology alternatively. Second, searching mode should be chosen. There are two searching modes: *Exclusive* and *Inclusive*. Mode *Exclusive* means that the input and output of the expected model templates should contain no other attributes but those selected in the first step. Mode *Inclusive* means that the input and output of the expected model templates could contain other attributes besides those selected in the first step. Finally, model templates satisfying the conditions are listed at the right bottom for viewing and editing. In Fig.9, two model templates are got as the searching result under the search conditions and mode *Inclusive*.

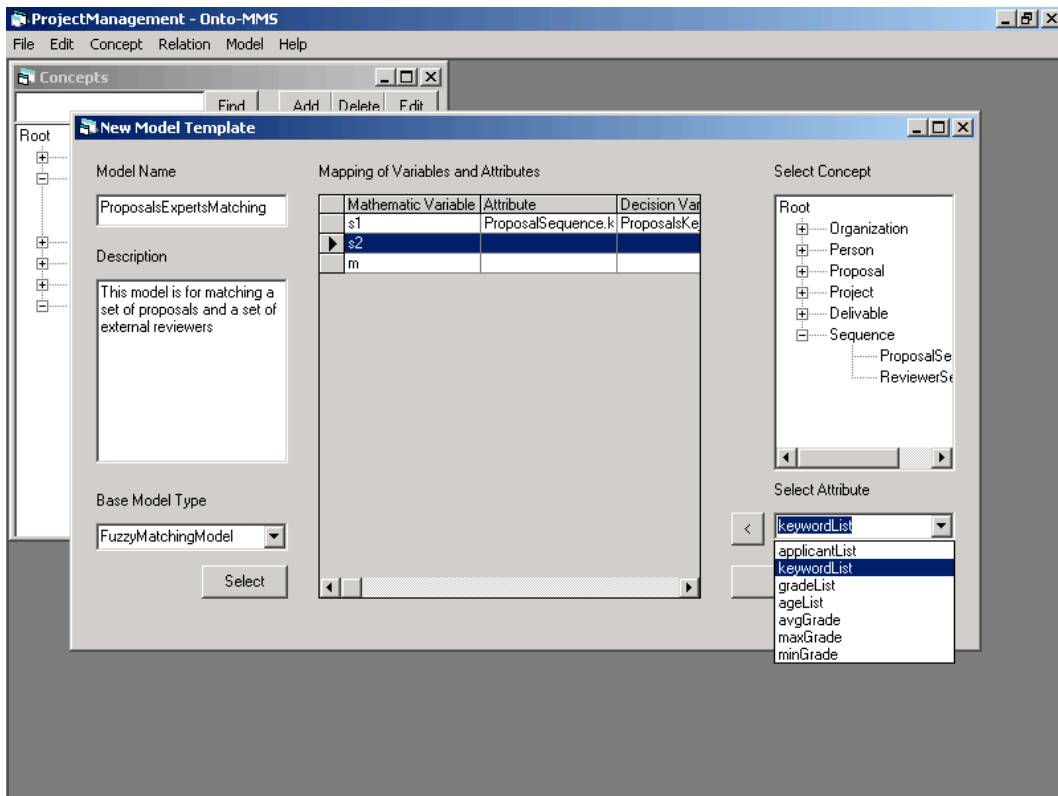


Fig.8. Creating New Model Templates

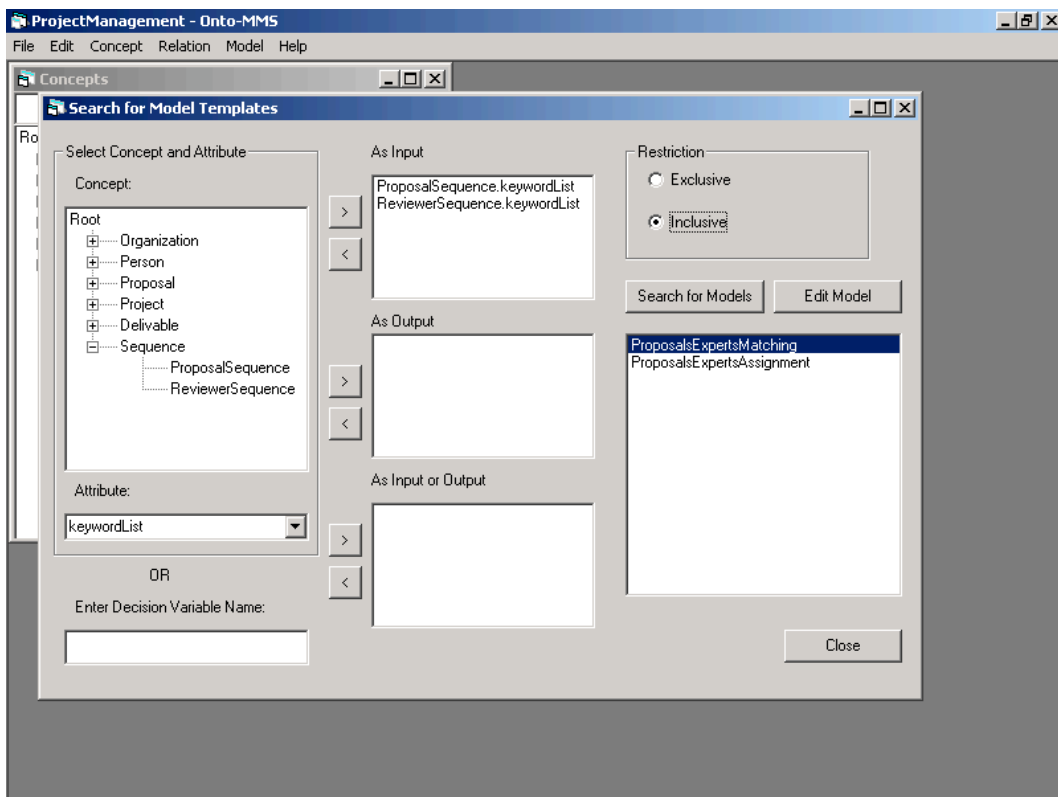


Fig.9. Searching for Model Templates

With this Web service-based model management architecture, decision models can be easily accessed from anywhere through the Internet. In Fig.10, a decision model “ProposalsReviewersAssignment” is applied, and the user is confirming the result in the ISIS.

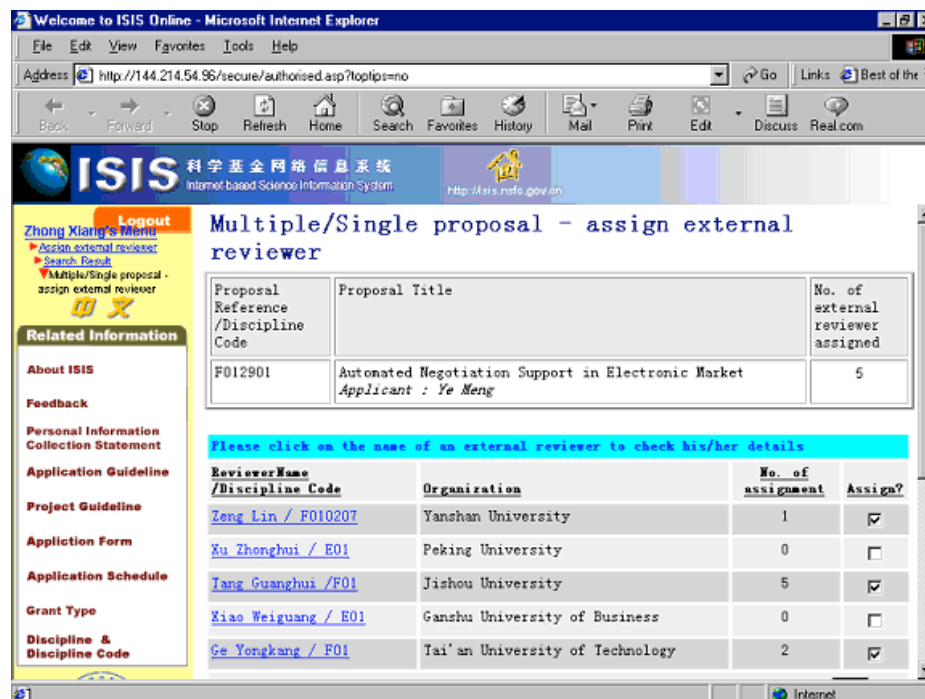


Fig 10. Access of a decision model through the Internet

To sum up this application, we can say, some problems aroused in the first section have been solved by the proposed framework. The resources in the ISIS (including models, domain knowledge, et al.) are shared as ontology, and can be understood and utilized correctly by many different categories of users with different knowledge background. DOML provides a standard and unified way to represent the resources so that they can be shared and interchanged easily in the Internet environment. Outside IRISs in other organizations can interact with decision models in the ISIS following the standards for Web services.

#### 4. Conclusion

This paper proposes a practicable model management framework based on ontology engineering and Web services technology. Compared with previous model management framework, the proposed framework has benefits coming from its components at different levels of abstraction:

- 1) With Decision Ontology, we make the domain knowledge and decision models sharable and reusable to users from different background, and interoperable to other software agents.
- 2) Using DOML, the modeling system retains the distinctive characteristics of XML: simplicity, extensibility, interoperability and openness; models can be understood easily by both humans and computers, and can be processed by any XML parser and easily interchanged over the Internet.
- 3) The Web service-based system architecture provides a standard means for users and other software systems to interact with decision models over the Internet; with models implemented as Web services, the proposed approach is also hopeful to achieve the independence of model representation and model solution, with model interface

standards to facilitate building a library of models and easily accessed solvers for retrieval, which is regarded as one of the desirable features of a new generation of modeling systems (Krishnan et al. 2000).

Further research directions include the following:

- 1) Formal specification of Decision Ontology can be made by Description Logic. It will give more "semantic" information about the ontology and provide a means for reasoning on the metadata. The work of Semantic Web (Horrocks 2002) community in Web Ontology area can be used for reference.
- 2) The exchange of DOML model specification with that of other management frameworks (such as SML, GAMS) will be useful for the evolution of the proposed framework. The aforementioned XSL will be hopefully an optional means.

## Reference

- Blanning, R.W. "A relational theory of model management," in: *Decision Support Systems: Theory and Applications.*, C.W. Holsapple and A.B. Whinston (eds.), Springer, 1987.
- Blanning, R.W. "Model management systems : An overview," *Decision Support Systems* (9:1), 1993/1 1993, pp 9-18.
- Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., and Weerawarana, S. "Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI," *IEEE Internet Computing* (6:2) 2002, pp 86-93.
- Dolk, D.R. "Integrated model management in the data warehouse era," *European Journal of Operational Research* (122:2), 2000/4/16 2000, pp 199-218.
- Elst, L.v., and Abecker, A. "Ontologies for information management: balancing formality, stability, and sharing scope," *Expert Systems With Applications* (22:4) 2002, pp 357-366.
- Fensel, D. *Ontologies: a silver bullet for knowledge management and electronic commerce* Springer, 2001.
- Geoffrion, A.M. "Structured Modeling: survey and future directions," *INFORMS Interactive Transactions of ORMS* (1:3) 1996.
- Gibbins, N., Harris, S., and Shadbolt, N. "Agent-based Semantic Web Services," *Web Semantics: Science, Services and Agents on the World Wide Web* (1:2), 2004/2 2004, pp 141-154.
- Harold, E.R. *XML bible*, (2 ed.) Hungry Minds, New York, 2001.
- Hong, S.N., Mannino, M.V., and Greenberg, B. "Measurement theoretic representation of large, diverse model base: the unified modeling language Lu," *Decision Support Systems* (10) 1993, pp 319-340.
- Horrocks, I. "DAML+OIL: a description logic for the semantic Web," *Bull. of the IEEE Computer Society Technical Committee on Data Engineering* (25:1) 2002, pp 4-9.
- Huh, S.-Y., and Kim, H.-M. "A real-time synchronization mechanism for collaborative model management," *Decision Support Systems* (37:3), 2004/6 2004, pp 315-330.
- IBM "Web Services: Taking e-business to the Next Level, White Paper," IBM, 2000.
- Kim, H. "An XML-based modeling language for the open interchange of decision models," *Decision Support Systems* (31:4), 2001/10 2001, pp 429-441.
- Klein, M., Fensel, D., Harmelen, F.v., and Horrocks, I. "The relation between ontologies and XML schemas," *Electronic Trans. on Artificial Intelligence*) 2001.
- Krishnan, R., and Chari, K. "Model management: survey, future research directions and a bibliography," *INFORMS Interactive Transactions of ORMS* (3:1) 2000.
- Lenard, M.L. "An object-oriented approach to model management," *Decision Support Systems* (9) 1993, pp 67-73.

- Liu, O., Tian, Q., and Ma, J. "A Fuzzy Description Logic Approach to Model Management in R&D Project Selection," The 8th Pacific Asia Conference on Information Systems (PACIS2004), 2004.
- Ma, J. "Type and inheritance theory for model management," *Decision Support Systems* (19:1), 1997/1 1997, pp 53-60.
- Marakas, G.M. *Decision Support System in the 21st Century*, (2 ed.) Prentice Hall, 2002.
- McIlraith, S.A., and Martin, D.L. "Bringing semantics to Web services," *IEEE Intelligent Systems* (18:1) 2003, pp 90-93.
- Muhanna, W.A., and Pick, R.A. "Meta-modeling concepts and tools for model management: A systems approach," *Management Science* (40:9) 1994, pp 1093-1123.
- Nambiar, U., Lacroix, Z., Bressan, S., Lee, M.L., and Li, Y. "Current approaches to XML management," *Internet Computing, IEEE* (6:4) 2002, pp 43 - 51.
- Noy, N.F., and McGuinness, D.L. *Ontology development 101: a guide to creating your first ontology* Knowledge Systems Laboratory Stanford, 2001.
- Schneeweiss, C. "Distributed decision-making -- a unified approach," *European Journal of Operational Research* 2003.
- Suh, C.-K., Suh, E.-H., and Lee, D.-M. "Artificial intelligence approaches in model management systems: A survey," *Computers & Industrial Engineering* (28:2), 1995/4 1995, pp 291-299.
- Tian, Q., Ma, J., Liang, J., Kwok, R.C.W., and Liu, O. "An organizational decision support system for effective R&D project selection," *Decision Support Systems* (In Press, Corrected Proof).
- Tian, Q., Ma, J., and Liu, O. "A Hybrid Knowledge and Model System for R&D Project Selection," *Expert Systems with Applications* (23:3) 2002a, pp 265-271.
- Tian, Q., Ma, J., and Liu, O. "An ontology-based approach to model management in organizational decision support systems," Proceedings of International Conference on Decision Making and Decision Support in the Internet Age (DSI-Age2002), 2002b.
- Tsai, Y.C. "Comparative analysis of model management and relational database management," *Omega* (29) 2001, pp 157-170.
- Walsh, A.E. (ed.) *UDDI, SOAP and WSDL : the Web services specification reference book*. Prentice Hall PTR, Upper Saddle River, NJ, 2002.
- Wang, M., Wang, H., Xu, D., Wan, K.K., and Vogel, D. "A web-service agent-based decision support system for securities exception management," *Expert Systems with Applications* (27:3), 2004/10 2004, pp 439-450.
- Wu, I.-L. "Model management system for IRT-based test construction decision support system," *Decision Support Systems* (27:4), 2000/1 2000, pp 443-458.