2008 International Conference on Intelligent Computation Technology and Automation

Semantics driven Decision Making and its Dynamic Semantics Presentation

Zhang Bo
College of Electronics and Information Engineering Tongji University Shanghai, China

Xiang Yang
College of Electronics and Information Engineering Tongji University Shanghai, China

Zhang Bei
College of Electronics and Information Engineering Tongji University Shanghai, China

Zou Guobing
College of Electronics and Information Engineering Tongji University Shanghai, China

drinker007@sina.com

Abstract

A new semantics driven decision making pattern is proposed because of the lack of semantic comprehension and sharing in traditional decision support system. The structure of decision support system is a three layer architecture and composed by interaction level, ontology level and decision level. In this semantics based decision support system, the decision domain problem description ontology and the decision model ontology are defined to describe the static and dynamic semantics which appears in the course of decision making and support knowledge interaction and sharing. And then the decision could be making by two steps: semantics understanding of decision problem and collaborative problem solving by multi-models.

1 Introduction

There are some bottle-necks in current researches on Decision Support System (DSS), such as, the structured characteristic of problems, the comprehension and representation of the semantics of problems, the representation and reasoning of model construction and knowledge inference, and the intelligence of the interaction between human and computers. We have noticed that there is a common issue behind these bottlenecks which is how to represent and comprehend the semantics of decision problems and their solving models, and how to interact and share on the base of this common semantics among different hosts (human and agents). In the decision domain, both decision problems and decision models have characteristic semantics and any decision from decision makers is based on the comprehension of this. However, researches on the representation and comprehension of the semantics of decision problems and models have been lacking for a long time, which causes the failure of helping decision makers work effectively when facing with the dynamic semantic environment. We consider that decision is not only based on the static information representation, but also a kind of process which is driven by problems' semantics and operated by models. And the semantics of the behaviors of decision making also need to be represented explicitly. The process of decision making includes not only the semantic reasoning of static knowledge but also the reasoning of specific decision behavior in dynamic semantic environment.

Along with the maturation and wide spread of Internet, software has been evolving in several aspects, such as configuration, internal requirement, key techniques and application models. As a kind of computing software, DSS should be an open distributed system based on Internet which has some advantages on sharing knowledge and cooperating in decision making. The concepts of the semantic web and ontology proposed a good method in comprehending semantics automatically during decision making. Ontologies are used to represent semantics and semantics could be a guide or a benchmark in each step of decision making [1], so that DSS could adapt to the change of environment automatically and aim at the purpose of decision-makers accurately.

In recent years, researchers have been working on how DSS could make use of ontologies and the semantic web theory. In 2006, Lee Seok-Won[2] suggested retrieving knowledge from natural languages according to instances and experiences to establish domain ontologies for decision problems. Kebair[3] proposed that DSS could make use of multi-agents because those agents could communicate in semantics which represented by ontologies. Smirnov [4] proposed a kind of knowledge integration driven by ontologies in

978-0-7695-3357-5/08 $25.00 © 2008 IEEE
DOI 10.1109/ICICTA.2008.295

dynamic environment, so that information and knowledge related to decision making could be integrated into ontology specification which could provide a knowledge representing and sharing platform for establishing and solving models during decision making.

There are many methods of semantic representation. Reference [6] proposed a model which uses concept, content and context to construct semantics. Shi Zhong-zhi and his colleagues proposed a kind of dynamic description logic which extends the traditional description logic by behaviors description to describe the dynamic semantics in the semantic web [8, 9]. Reference [10] suggested a kind of representation of constructing semantics by using features space.

In this paper, dynamic logic and description logic combined together by using ontologies as the representation of decision making problems and models, so that the representations and related algorithms of static and dynamic semantics could be presented. The architecture and principles of semantic DSS are also proposed.

2 Semantics representation of decision making

2.1 Ontologies in decision making domain

The ontologies of DSS can be divided as: Decision Domain Problem Description Ontology (DDPDO) and Decision Model Ontology (DMO). DDPDO describes the concepts related to decision making problems and their relationships, and provides problem categories and properties which could be used to describe a certain kind of problems. DMO describes constructing model methods by constructing-model sub-ontology and algorithms for solving problems by solving-model sub-ontology.

Definition 1 Decision Domain Problem Description Ontology (DDPDO) can be defined as a set of 6 elements $DO = (C, R, OP, SP, GP, I)$. C means the concept of a problem witch is the exclusive standard of a certain kind of problems; R is the set of relations between this concept and others; OP represents the set of possible initial properties of the certain problem concept; SP is the set of properties which emerges during solving process; GP describes the set of possible target properties of this concept; and I is the set of instances of the concept.

Definition 2 Decision Model Ontology (DMO) is $MO = (C, R, P, RS, I)$, C represents the model concept; R is the relations set of the model; P means the set of properties of this model concept; RS is the solving algorithm of this model; and I is the instances set.

2.2 Semantic representation for static objects

DSS mainly deals with decision problems; decision models; databases during decision making process; method bases and knowledge bases. And these objects are all static ones. According to the thought we explained above, we divided these static objects into two categories: semantics for decision problems and semantics for decision models. Semantics representation method is defined as follows.

Definition 3 Decision problem semantics is defined as $PS = (Id, O, S, A, G, SubP, \mathrm{Re}\,lationP)$. Id is the exclusive identification of this problem; O is a non-empty set of initial states of the problem; the non-empty set S includes states at a certain time; A represents the possible purposes description of decision-makers; G is the set of possible targets of decision making problems; $SubP$ is the set of possible sub-problems; $RelationP$ means the existence of a kind of decision making problems which have some other relations with this problem except father-son relationship. Obviously, any problem semantics can be formalized by ontologies: $PS.Id \subseteq DO.C$, $PS.O \subseteq DO.OP$, $PS.S \subseteq DO.SP$, $PS.G \subseteq DO.GP$.

Definition 4 Decision model semantics is $M = (Id \mid MC, O, Input \mid Output, S, B)$. The ordered pair $Id \mid MC$ means the exclusive identification and the name of this model; the non-empty set O includes the ontologies describing this model; $Input \mid Output$ represents the possible input and output format of the model; S is a non-empty set of states at a certain time; the non-empty set B is the set of possible decision behaviors.

2.3 Dynamic behaviors semantics and surroundings semantics

Behaviors mean the states transformation operated by decision models. The executor of behaviors is decision models, and the target objects of behaviors are usually composed of problems, data, and models, etc. We proposed a new kind of behaviors description method which can illustrate not only the executor and his target but the precondition and effect of the execution.

Definition 5 A behavior description is a kind of representation as $B(x_1, \cdots x_n \mid y_1, ..., y_m) \equiv (P_B, E_B)$.

1. B is the symbol of the behavior;

2. $x_1, \cdots, x_n$ are individual variables, representing executors of the behavior, named executing

variables; $y_1, \cdots, y_m$ are also individual variables, representing the targets of the behavior, named object variables;

3. P_B is the set of premise formulas, defining necessary preconditions for executing variables before execution;

4. E_B is the set of result formulas, defining the results of execution, E_B is the set of exu/obj, the state order pairs, exu represents the set of results of executing variables after execution, obj represents the set of results of objects variables after execution.

DSS needs efficient semantic descriptions of surroundings for each decision making individuals. We proposed a semantic definition of dynamic surroundings as follows:

Definition 6: A surroundings description can be described as $E(X\,|\,(R,S)) \equiv (E_B, E_F)$. In the formula, the order pair $X\,|\,(R,S)$ points out the set of binary relations $R_1(X, y_1),..., R_n(X, y_n)$ of any certain object X, and the set of states of X at a certain time; E_B is the set of behaviors which can be executed by X under these surroundings; E_F is the set the condition formulas of the object X.

Surroundings semantics is a kind of illustrated context for the objects during decision making process. This kind of semantics make it clear that each object has a certain context witch could be understood explicitly by all the terminals of decision making process. The preconditions set for behaviors semantics can be regarded as a series of conditions for the execution of behaviors, while surroundings semantics is a confirmed illustration, a context of the behaviors.

2.4 Support algorithm for behaviors semantics by surroundings semantics

For a certain surroundings, it is possible that different objects executing different behaviors. In this situation, system should consider not only whether the preconditions of certain behavior are satisfied, but whether any collisions would emerge in the surroundings. So the judgment algorithm of surrounding supporting behavior can be divided into two parts: one is judgment of preconditions of the certain behavior; the other is surrounding collision judgment of the certain object.

Algorithm1: Surrounding supporting for behavior

1. $P \leftarrow b.P_b$

2. For behavior $b\ (x_1, \cdots x_n\,|\,y_1,..., y_m) \equiv (P_b, E_b)$, which would be executed, algorithm dose following operations:

A. For all x_n, y_m, algorithm finds out the $E(x_n\,|\,(R,S)) \equiv (E_B, E_F)$ and $E(y_m\,|\,(R,S)) \equiv (E_B, E_F)$;

B. For each $x_n = \{x_n\,|\,b \in E(x_n).E_B\}$, behavior b is related to object x_n. Algorithm would compute function $match(b.P_b, E(x_n).E_F)$;

C. If all the objects x_n can return the function $match(b.P_b, E(x_n).E_F)$ values as true, algorithm does $P \leftarrow P \bigcup E(x_n).E_F$, and goes to step 3. If the any of functions return value as false, the precondition set of behavior is not satisfied. The behavior would be hung up, and wait for next judgment, algorithm finishes.

3. For each object $x_n = \{x_n\,|\,b \notin E(x_n).E_B\}$, behavior b is not related to object x_n, algorithm does $P \leftarrow P \bigcup E(x_n).E_F$;

4. For $p_l = \{p_l\,|\,p_l \in b.P_b \wedge p_l \notin E(x_n).E_F\}$, algorithm finds out the corresponding object set L. For all $l \in L$, algorithm does function $match(b.P_l, E(l).E_F)$;

5. If all matching functions are successful, algorithm dose $P \leftarrow P \bigcup E(l).E_F$, and goes to step 6. Otherwise, precondition set is not satisfied, and the behavior would be hung up, and algorithm finishes.

6. Algorithm computes function $confilctset(P)$. If there is no collision, the behavior would be executed, and algorithm finishes.

3. Computing environment and principles of semantic decision making

In this section, we prototyped the architecture of DSS based on semantics which can be divided into 3 layers. Comparing with the traditional DSS computing environment, the one of semantic decision could have some features as follows: Enough capability of describing semantics. Parallel and distributed processing. Collaboration of multi-DSS.

3.1 Software architecture of semantic decision making

Different from traditional DSS, the semantic decision model must be established on a well understanding of any object's semantics during decision making process. During semantic decision making, a large amount of decision objects come from Internet and have different semantics. In this circumstance, each of the decision terminals needs the capability of automatically dealing with semantics. Semantic decision making is constructed by 3 layers:

semantics understanding, semantics interaction and decision realization.

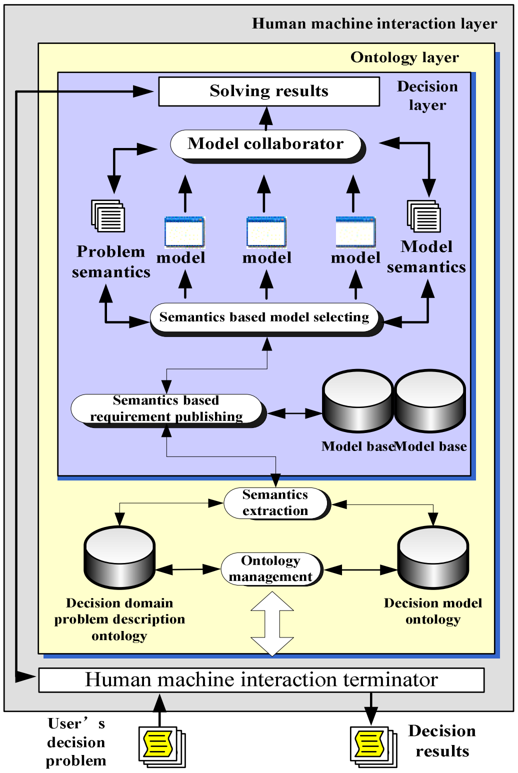

Fig.1 architecture of semantic decision making

Showed as figure 1, semantic decision making is divided as: □ Human-machine interaction layer. This layer provides an intelligent interaction interface between decision-makers and computers, or agents and agents. Information from different terminals in the web can be captured by agents, and operated half-manually in the interface so that semantics can be identified. However, other information input by executors is also operated here to realize semantics identification. □ Ontology layer. Ontology layer defines definitions of semantic decision and stores two kinds of ontologies: DDPDO and DMO. In this layer, semantics is extracted from the information provided by human-machine interaction layer according to ontologies and provided to decision layer. ③ Decision layer. This layer is the core layer and handles intelligent construction of decision model, selection of model and collaborative solving, by using formalized semantics.

3.2 Working model of semantic decision making

Semantic decision making is processed by different terminals witch are distributed in the web and communicate with each other by asynchronous messages. The system provides decision making for the executors through decision making clients. During decision making, a client first pre-processes the decision problem of users by dividing it to several sub-problems, and then distributes some of them which it can not solve itself to other clients around the web, so that others can provide cooperation to complete decision making. Therefore, the process of semantic decision making can be divided into 2 parts as follows:

1. Semantic understanding of decision problems

The representation of a decision problem must indicate the features of the problem. In this paper, we classified problems into two types: user-oriented problems and machine-oriented ones. For user-oriented problems, natural languages aided by symbols are used to describe a problem; as for machine-oriented ones, a problem is represented by key words to keep the semantics. Key words in natural languages description reflect the meaning of the sentences. According to DDPDO, key words can be extracted by the technology of natural languages understanding. Semantics can be well represented by corresponding eigenvectors, and key words can be the parameters in these semantic vectors. An agent first decides witch concept in DDPDO this problem belongs to according to the key words; and then gets the features of this problem on the base of the properties of the certain concept defined in the ontology. The agent can find the ontology where the key words are so as to find out related concepts. And these concepts can link DDPDO and DMO; witch can help to establish specific relationships between decision problem semantics and decision model semantics.

Interactions between executors and computers, or among agents, are based on semantics. Interactions between human and computers are divided into two categories: interactions between executors and computers, and the ones among agents. When executors input a decision problem, agents can consider the description of the problem in DDPDO and dynamically provide special interface for the certain problem. Decision process interactions means catching new information from executors and feeding back the real-time states of decision making during the process of understanding decision problem, constructing intelligent model and solving model, so that users can adjust decision at any moment.

2. Collaborative solving by multi-models

Usually, complex decision problems can not be solved by a single decision model. The cooperation of multi-models is a good thought. As showed in figure 2 we arrange cooperation in two aspects: on one hand, based on the semantics understanding of a complex problem, the problem is disassembled into ordered sub-problems, facts related to decision and data. And solving method is also decided; on the other hand, corresponding model is selected for each sub-problem, according to the definition of each model in DMO and cooperation mechanism is ready for multi-models working together. We choose multi-model cooperation based on roles to solve problems. Model's role is an element of model semantics. It describes the mission and function of the model in a certain decision problem. We can sense witch of the role-models are needed through problem semantics and then select proper models according to requirements and arrange semantic relations among these models, so as to establish a cooperative problem-solving architecture, and construct cooperative context for solving problems. During the solving process, models use collision prediction, collision negotiation and cooperation to solve the decision problem.

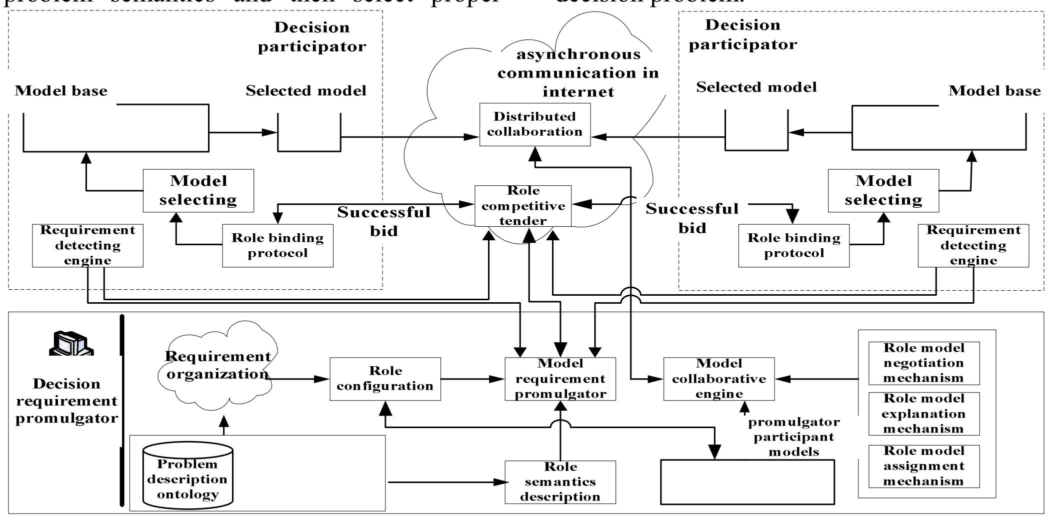

Fig.2 collaborative solving of multi-models

4. Conclusion

Concerning some issues in DSS researches, we imported technologies from ontology and the semantic web to propose a new DSS driven by semantics, and designed the DSS architecture with 3 layers. In this DSS, different hosts (human and agents) can share knowledge and interact with each other on the base of common semantics, and agents can understand the purpose of executors efficiently and effectively. Therefore, DSS is driven by semantics in word and deed.

Acknowledge

This work is supported by CNSF Grant #70371054, #70771077 and the National High-Tech Research and Development Plan of China (863) under Grant No. 2008AA04Z106.

References

[1]Gruber TR, "*A Translation Approach to Portable ontology Specifications* ", Knowledge Acquisition , 1993, 5:199-220.

[2]Lee, Seok-Won ,Muthurajan, Divya, "*Building decision support problem domain ontology from natural language requirements for software assurance*", International Journal of Software Engineering and Knowledge Engineering, 2006, p 851-884

[3]Kebair, F. Serin, F, "*Multiagent approach for the representation of information in a decision support system, Artificial Intelligence: Methodology, Systems, and Applications*". Lecture Notes in Artificial Intelligence Vol. 4183, 2006, 98-107.

[4]Smirnov, A.V, Pashkin, M.P., "*An Ontology-driven information integration for operational decision support*", 2005 7th International Conference on Information Fusion , 2005, 8 pp.

[5]De la Asuncion M, Castillo L, "*SIADEX: An interactive knowledge-based planner for decision support in forest fire fighting*" , AI Communication, 2005, 18 (4): 257-268

[6]Edwards, S.H, "*Toward a model of reusable software subsystems*". In: Philbrick, S. Stevens, M. Eds. Proceedings of the 15th Annual Workshop on Software Reuse. Larry Latour, 1992

[7]Xiang Yang, Shen Hong, "*Intelligent Decision Support System Based on Agent*", Computer Engineering, 2006, 32(16): 167-168.

[8]Shi Zhongzhi, Dong Mingkai, Jiang Yuncheng, Zhang Haijun, "*Logic base for semantic web*", Science in China. Ser. E., 2004, 34(10): 1123-1138

[9]Dong Mingkai, Zhang Haijun, Shi Zhongzhi, "*An Agent Model Based on Dynamic Description Logic*", Journal of Computer Research and Development, 2004, 41(5): 780-786

[10]Dong Guangzhi, Liu Junfei, Qi Xuan, "*A kind of reactive SPM and the expression of its dynamic semantics with XYZ*", Journal of Software, 2005, 16(11): 1876-1885