FML-based Type-2 Fuzzy Ontology for Computer Go Knowledge Representation

Chang-Shing Lee*, Mei-Hui Wang*, Zhi-Rong Yan*, Yu-Jen Chen*, Hassen Doghmen**, and Olivier Teytaud**
*Dept. of Computer Science and Information Engineering, National University of Tainan, Taiwan
**TAO, Lri, Univ. Paris-Sud, Inria Saclay-IDF, Orsay, France
*leecs@mail.nutn.edu.tw **olivier.teytaud@gmail.com

***Abstract*—In this paper, an Fuzzy Markup Language (FML)-based type-2 fuzzy ontology is proposed to represent the computer Go knowledge, including an FML transformation mechanism, a type-2 fuzzy set construction, and a type-2 fuzzy set inference mechanism. The FML transformation mechanism transforms the generated smart game format (SGF) files into an FML-based document to describe the computer Go ontology. The type-2 set construction is responsible for building the type-2 fuzzy sets. Based on the built FML-based document and type-2 fuzzy sets, the type-2 fuzzy set inference mechanism infers the possibility of the game's winning rate. It is hoped that the proposed idea are feasible for inferring the winning rate of the games in the future.**

***Keywords*—Ontology, Type-2 Fuzzy Inference, Fuzzy Markup Lanuage, Computer Go**

I. Introduction

Knowledge is defined as any information that is relevant, actionable, and is based on a person's experience [1]. All knowledge workers share certain characteristic activities. However, when data is annotated, it's done against a framework or ontology [2]. Ontology has become a very powerful way of sharing the knowledge as well as representing the information and its semantics [3]. It is also a conceptualization of a real world domain into a human understandable, machine-readable format consisting of entities, attributes, relationships, and axioms [4]. Moreover, ontology mediation allows us to combine knowledge from the ontologies [2]. For example, Lee *et al.* proposed a fuzzy ontology to apply for news summarization [5] and they also proposed an ontology-based intelligent decision support agent for capability maturity model integration (CMMI) project monitoring and control [6]. Reformat and Ly [3] proposed an ontology-based approach to provide a rich environment for expressing different types of information including perceptions.

Fuzzy Logic Systems (FLSs) have been credited with providing an adequate methodology for designing robust systems that are able to deliver a satisfactory performance when contending with the uncertainty, noise, and imprecision attributed to real world environments and applications. Moreover, FLSs provide a framework for representing the information in a human readable form [7]. As a result, FLSs have been used in wide range of applications including fuzzy ontologies. Several researchers have explored the use of fuzzy ontologies. Quan et al. [8] presented the automatic fuzzy ontology generation for semantic help desk support. Knappe et al. [9] proposed a fuzzy ontology-based query enrichment. Calegari et al. [10] introduced the fuzzy ontologies and scale-free networks analysis. Hudelot et al. [11] proposed the fuzzy spatial relation ontology for image interpretation. Quan et al. [12] proposed the automatic fuzzy ontology generation for semantic web. Type-2 FLSs could be used to handle the uncertainties in the group decision making process as they can model the uncertainties between expert preferences using type-2 fuzzy sets. A type-2 fuzzy set is characterized by a fuzzy Membership Function (MF), i.e. the membership value (or membership grade) for each element of this set is a fuzzy set in [0,1], unlike a type-1 fuzzy set where the membership grade is a crisp number in [0,1] [13]. The MFs of type-2 fuzzy sets are three dimensional and include a Footprint of Uncertainty (FOU). Hence, type-2 fuzzy sets provide additional degrees of freedom that can make it possible to model the inter user (group) uncertainties which involve the varying opinions and preferences of experts. The type-2 fuzzy sets can model the requirements of a person specification that's reflective of all the experts' opinions [14].

Fuzzy Markup Language (FML) is a fuzzy-based markup language that can manage fuzzy concepts, fuzzy rules, and a fuzzy inference engine [15, 16]. Additionally, FML is composed of three layers—eXtensible Markup Language (XML), a document-type definition, and extensible stylesheet language transformations. Differently from other similar approaches such as Fuzzy Control Language (FCL) or MATLAB Fuzzy Inference System (FIS) developed by the Matworks, FML exhibits additional benefits in the FLC programming that are related to its XML derivation. Indeed, whereas FCL or FIS code is totally based on a textual representation, FML programs are coded through a collection of correlated semantic tags capable of modeling different controller components by exploiting abstraction benefits offered by XML tools. From the implementation point of view, these benefits allow fuzzy designers (1) to simply code their ideas on heterogeneous hardware without having knowledge about programming details related to the different platforms and (2) to program a fuzzy controller without referring to general purpose computer languages. In our scenario, FML allows us to program diet agents in a very short time and simultaneously to test the inferred results on different hardware without additional programming works [17].

The game of Go is one of the last board games where the strongest humans are still able to easily win against computers in 19×19 games. Researchers, however, have discovered new performing algorithms and computers are catching up really fast. The game results also have shown that if the computer can learn more knowledge and strategy from professional Go players through the constructed ontology, then the computer Go will approach the level of professional go player very fast [18]. In this paper, an FML (Fuzzy Markup Language)-based type-2 fuzzy ontology for computer Go representation is proposed. The remainder of this paper is organized as follows. Section II briefly describes the FML. Type-2 fuzzy ontology is introduced in Section III. The computer Go knowledge representation is introduced in Section IV. Conclusions are finally given in Section V.

II. FUZZY MARKUP LANGUAGE

A. Fuzzy Markup Language Overview

Fuzzy logic controllers (FLCs) have been applied in consumer goods. A fuzzy control allows the designer to specify the control in terms of sentences rather than equation by replacing a conventional controller, a proportional-integral-derivative (*PID*) controller, with linguistic IF-THEN rules. The use of linguistic variables represents a significant paradigm shift in system analysis: using the linguistic approach, the focus of the attention in the dependencies representations is shifted from differential equations to fuzzy IF-THEN rules in the form *if X is A then Y is B*. The propositions *X is A* and *Y is B* are called fuzzy clauses; *A* and *B* are linguistic variables and *X* and *Y* are their linguistic values (for instance, *if Pressure is high then Volume is low*). The main components of fuzzy controller are (organized in a hierarchical way): (1) fuzzy knowledge base; (2) fuzzy rule base; (3) inference engine; (4) fuzzification subsystem; and (5) defuzzification subsystem [17]. The root of fuzzy controller taxonomy, the *Controller node*, is represented through the FML tag <FUZZYCONTROL>. <FUZZYCONTROL> uses three tags: *type*, *defuzzyfyMethod*, and *ip*. The *type* attribute permits to specify the kind of fuzzy controller, Mamdani or Takagi-Sugeno-Kang (TSK); *defuzzyfyMethod* attribute defines the defuzzification method used in modeled controller; *ip* is used to define the location of controller in the computer network.

B. Fuzzy Markup Language Knowledge Base

The fuzzy knowledge base is defined by means of the tag <KNOWLEDGEBASE> which maintains the set of fuzzy concepts used to model the fuzzy rule base. The <KNOWLEDGEBASE> uses the attribute *ip* that determines the location in the network of whole fuzzy knowledge base of our system. In order to define the fuzzy concept related controlled system, <KNOWLEDGEBASE> tag uses a set of nested tags: (1) <FUZZYVARIABLE>; (2) <FUZZYTERM>; (3) a set of tags defining a shape of fuzzy sets [17]. <FUZZYVARIABLE> defines the fuzzy concept, for example, "temperature"; <FUZZYTERM> defines a linguistic term describing the fuzzy concept. The attributes of <FUZZYVARIABLE> tags are: *name*, *scale*, *domainLeft*, *domainRight*, *type*, and *ip*. <FUZZYTERM> uses one attribute, *name* used to define the linguistic value associate with fuzzy concept. Fuzzy shape tags, used to complete the definition of fuzzy concept, are: <TRIANGULARSHAPE>, <LINEARSHAPE>, <TRAPEZOIDSHAPE>, <SSHAPE>, <ZSHAPE>, and <PISHAPE>. Every shaping tag uses a set of attributes which defines the real outline of corresponding fuzzy set. The number of these attributes depends on the chosen fuzzy set shape [17].

C. Fuzzy Markup Language Rule Base

The root of fuzzy rulebase component is modeled by the <RULEBASE> tag which defines a fuzzy rule set. The <RULEBASE> tag uses two attributes: *inferenceEngine* and *ip*. The former is used to define inference operator type: *MinMaxMinMamdani* or *LarsonProduct*. The latter defines the network location of the set of rules used in fuzzy controller. In order to define the single rule, the <RULE> tag is used. The tags used by <RULE> are: *id*, *connector*, *weight*, and *ip*. The definition of antecedent and consequent rule part is obtained by using <ANTECEDENT> and <CONSEQUENT> tags. <CLAUSE > tag is used to model the fuzzy clauses in antecedent and consequent part. In order to treat the fuzzy operator "*not*" in clauses, the tag <CLAUSE > uses the boolean attribute "*not*". To complete the definition of fuzzy clause, the <VARIABLE>, <TERM>, and <TSKPARAM> have to be used. In particular, the pair "<VARIABLE>, <TERM>" is used to define fuzzy clauses in antecedent and consequent part of Mamdani controllers rules as well as in antecedent part of TSK controllers rules. While, the pair "<VARIABLE>, <TSKPARAM>" is used to model the consequent part of TSK controllers rules [17].

III. TYPE-2 FUZZY ONTOLOGY

A. Definition of Type-2 Fuzzy Set

This paper presents a *type-2 fuzzy ontology* (*T2FO*) based on type-2 fuzzy sets (*T2FSs*) to describe the fuzzy concepts and fuzzy relations. The proposed *T2FO* stores *T2FSs* and is an extended version of the *fuzzy ontology* [5] proposed by our previous research works. As described above, *T2FSs* provide us with more design degrees of freedom than type-1 fuzzy sets; so using *T2FSs* has the potential to outperform the system using type-1 fuzzy sets, especially when dealing with an environment with high inter-user uncertainty levels such as computer Go handling where we have several experts and each expert is having a different opinion. A trapezoidal *T2FS* $\tilde{T}$ is a *T2FS* which is shown in Fig. 1 with the upper bound of *FOU*($\tilde{T}$), called T_U, and the lower bound of *FOU*($\tilde{T}$), called T_L. $\tilde{T}$ is represented by Eq. (1).

$$\tilde{T}=\{[l_L,m_{l_L},m_{r_L},r_L],[l_U,m_{l_U},m_{r_U},r_U]\} \quad (1)$$

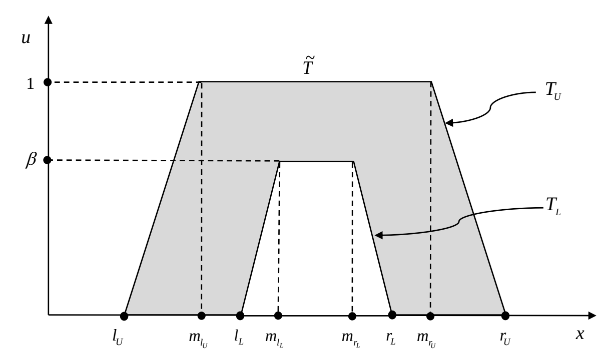

Figure 1. A trapezoidal *T2FS* $\tilde{T}$.

B. Structure of Type-2 Fuzzy Ontology

A *Type-2 Fuzzy Ontology* (*T2FO*) is a knowledge representation model for describing the domain knowledge with uncertainty. It is an extension of the domain ontology [5] and contains six layers, including a *domain layer*, a *category layer*, a *fuzzy concept layer*, *a fuzzy variable layer*, *a type-1 fuzzy set layer*, and a *type-2 fuzzy set* (*T2FS*) *layer*. The concepts and relations of the *T2FO* are constructed by fuzzy variables, fuzzy sets, and *T2FSs*. That is, a fuzzy variable including some fuzzy sets is used to represent a fuzzy concept. The *type-1 fuzzy set layer* and the *T2FS layer* are constructed by some fuzzy sets and *T2FSs*. The *T2FS layer* is an extension of the *type-1 fuzzy set layer*. The concepts in the *type-1 fuzzy set layer* are type-1 fuzzy sets (*T1FSs)* and the concepts in the *T2FS layer* are *T2FSs* aggregated from the *T1FSs* in the *type-1 fuzzy set layer*.

The structure of the six-layer *T2FO* is shown in Fig. 2. There are *i* fuzzy concepts in the *fuzzy concept layer*. Each fuzzy concept has some fuzzy variables in the *fuzzy variable layer*. There are m_i fuzzy variables, FV_{i1}, ..., FV_{im_i}, which are defined for the fuzzy concept *i*. The fuzzy sets, $FS_{in_i\_1}$, …, $FS_{in_i\_q_{ni}}$, are defined in the fuzzy variable FV_{im_i}, where q_{n_i} denotes the number of fuzzy sets for fuzzy variable FV_{im_i}. In the *T2FS layer*, there are some *T2FSs* generated by combining some *T1FSs* in the *type-1 fuzzy set layer*. For example, the "$T2FS_{11}$" is generated by combining two *T1FSs* FS_{11_1} and $FS_{1n_1\_1}$.

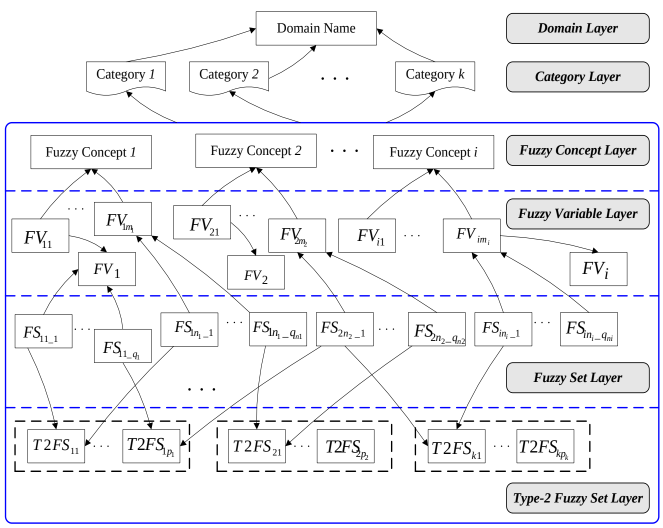

Figure 2. Structure of the type-2 fuzzy ontology.

C. Type-2 Fuzzy Onotlogy for Computer Go

Based on Fig. 2, the type-2 fuzzy ontology for computer Go is shown in Fig. 3. The domain name is *Type-2 Fuzzy Ontology for Computer Go*. In the fuzzy concept layer, there are several categories, such as *Go Player*, Computer Go Game, and *KGS.* In our opinion, the Go player's facial expression and the length of the thinking time that one player takes for per move could reflect the Go player's feelings when he/she is playing with the computer Go program. Additionally, the length of the game time and the number of games that the Go player competes are also the facts that may affect the game's result. Hence, in the *fuzzy concept layer*, there are fuzzy concepts *People*, *Computer Go Game*, and *KGS Platform*. Additionally, *Move Evaluation* (*ME*), *Affective Evaluation* (*AE*), *Thinking Time* (*TT*), *Game Time* (*GT*), and *Game No* (*GN*) are the fuzzy variables. The *type-1 fuzzy set layer* has several type-1 fuzzy sets such as *Low*, *Medium*, …, and *VeryHigh*. *Winning Rate* (*WR*) is defined in the *type-2 fuzzy set layer*. That is, if *ME* is *VeryLow*, *AE* is *Low*, *TT* is *Low*, *GT* is *Short*, and *GN* is *Short*, then *WR* is *VeryLow*.

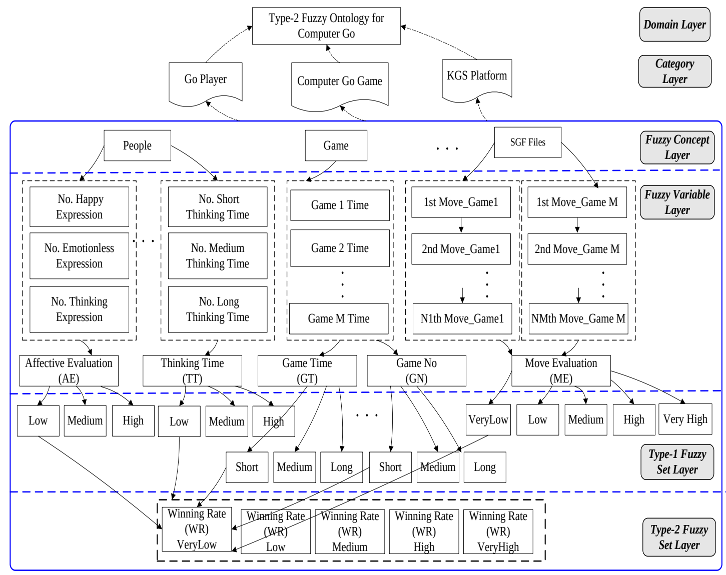

Figure 3. Type-2 Fuzzy Ontology for Computer Go.

IV. COMPUTER GO KNOWLEDGE REPRESENTATION

A. Structure of the Go Knowledge Representation

Figure 4 shows the structure of the Go knowledge representation, including an FML transformation mechanism, a type-2 fuzzy set construction, and a type-2 fuzzy set inference mechanism.

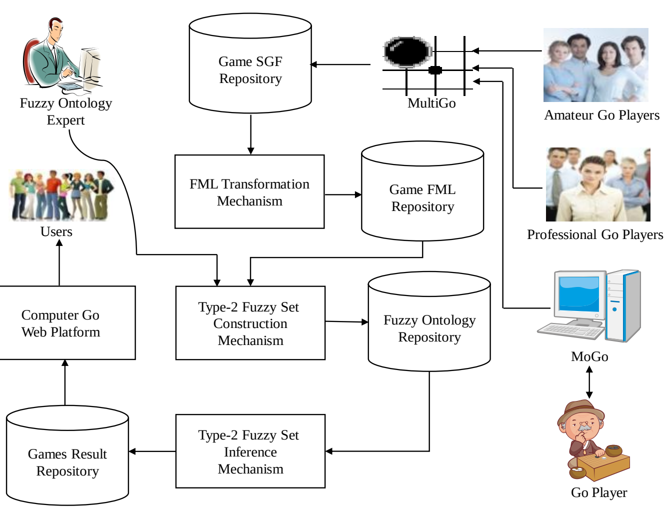

Figure 4. Structure of Go Knowledge Representation.

Figure 4 operates as follows. (1) Several amateur-level and professional-level Go players are invited to play with the computer Go program (MoGo) on the KGS platform. (2) The smart game format (SGF) files are generated and displayed via the MultiGo software. (3) The invited Go players give their valuable comments on the played games to improve the bad moves via the MultiGo software. The SGF files with comments are stored into the game SGF repository. (4) The FML transformation mechanism transforms SGF files into an FML-based document to describe the computer Go ontology, and store them into the game FML repository. (4) Based on the game FML repository, the fuzzy ontology expert constructs the type-2 fuzzy sets via the type-2 fuzzy set construction mechanism, and stores them into the fuzzy ontology repository. (5) The type-2 fuzzy set inference mechanism infers the winning rate of the game based on the fuzzy ontology repository, and stores the results into the game result repository. (6) Finally, the user can understand the game result via the computer Go web platform.

B. FML-based Knowedge Base and Rule Base

Based on the FML, an FML editor is used to construct the important knowledge base and rule base of the type-2 fuzzy set inference mechanism. The knowledge base describes the fuzzy concepts, including fuzzy variables, fuzzy terms, and membership functions of fuzzy sets. On the other hand, the rule base describes the fuzzy rule set, including the antecedent and consequence rule part. Table I lists part of FML view of the proposed approach, which is divided into knowledge base and the rule base. It indicates that there are one output fuzzy variable Winning Rate (*WR*), 405 fuzzy rules, and five input fuzzy variables, including *Move Evaluation* (*ME*), *Affective Evaluation* (*AE*), *Thinking Time* (*TT*), *Game Time* (*GT*), and *Game No* (*GN*). Each fuzzy variable has several fuzzy terms. For example, fuzzy variable *TT* has three fuzzy terms, namely *Short*, *Medium*, and *Long*.

TABLE I. PART OF FML VIEW OF THE PROPOSED APPROACH.

```
<?xml version="1.0"?>
<FUZZYCONTROL defuzzifymethod="CENTROID" ip="localhost" type="MAMDANI" fuzzysettype="Type2">
  <KNOWLEDGEBASE ip="">
    <FUZZYVARIABLE domainleft="0" domainright="100" ip="localhost" name="MoveEvaluation" scale="%" fstype="INPUT">
      <FUZZYTERM name="VeryGood" hedge="Normal">
        <TRAPZOIDALSHAPE>
          <param1 UpperBound="75" LowerBound="80" />
          <param2 UpperBound="85" LowerBound="85" />
          <param3 UpperBound="100" LowerBound="100" />
          <param4 UpperBound="100" LowerBound="100" />
        </TRAPZOIDALSHAPE>
      </FUZZYTERM>
    …
    </FUZZYVARIABLE>
    <FUZZYVARIABLE domainleft="0" domainright="100" ip="localhost" name="AffectiveEvaluation" scale="%" fstype="INPUT">
      <FUZZYTERM name="Happy" hedge="Normal">
        <TRAPZOIDALSHAPE>
          <param1 UpperBound="60" LowerBound="70" />
          <param2 UpperBound="95" LowerBound="95" />
          <param3 UpperBound="100" LowerBound="100" />
          <param4 UpperBound="100" LowerBound="100" />
        </TRAPZOIDALSHAPE>
      </FUZZYTERM>
    …
    </FUZZYVARIABLE>
    <FUZZYVARIABLE domainleft="0" domainright="100" ip="localhost" name="ThinkingTime" scale="%" fstype="INPUT">
      <FUZZYTERM name="Long" hedge="Normal">
        <TRAPZOIDALSHAPE>
          <param1 UpperBound="45" LowerBound="80" />
          <param2 UpperBound="95" LowerBound="95" />
          <param3 UpperBound="100" LowerBound="100" />
          <param4 UpperBound="100" LowerBound="100" />
        </TRAPZOIDALSHAPE>
      </FUZZYTERM>
    …
    </FUZZYVARIABLE>
    <FUZZYVARIABLE domainleft="0" domainright="120" ip="localhost" name="GameTime" scale="min" fstype="INPUT">
      <FUZZYTERM name="Long" hedge="Normal">
        <TRAPZOIDALSHAPE>
          <param1 UpperBound="60" LowerBound="70" />
          <param2 UpperBound="80" LowerBound="80" />
          <param3 UpperBound="120" LowerBound="120" />
          <param4 UpperBound="120" LowerBound="120" />
        </TRAPZOIDALSHAPE>
      </FUZZYTERM>
    …
    <FUZZYVARIABLE domainleft="0" domainright="8" ip="localhost" name="GameNo" scale="" fstype="INPUT">
      <FUZZYTERM name="Long" hedge="Normal">
        <TRAPZOIDALSHAPE>
          <param1 UpperBound="5" LowerBound="5" />
          <param2 UpperBound="6" LowerBound="6" />
          <param3 UpperBound="8" LowerBound="8" />
          <param4 UpperBound="8" LowerBound="8" />
        </TRAPZOIDALSHAPE>
      </FUZZYTERM>
    …
    </FUZZYVARIABLE>
    <FUZZYVARIABLE domainleft="0" domainright="100" ip="localhost" name="WinningRate" scale="%" fstype="OUTPUT">
      <FUZZYTERM name="VeryHigh" hedge="Normal">
        <TRAPZOIDALSHAPE>
          <param1 UpperBound="75" LowerBound="80" />
          <param2 UpperBound="85" LowerBound="85" />
          <param3 UpperBound="100" LowerBound="100" />
          <param4 UpperBound="100" LowerBound="100" />
        </TRAPZOIDALSHAPE>
      </FUZZYTERM>
    …
    </FUZZYVARIABLE>
  </KNOWLEDGEBASE>
  <RULEBASE inferenceengine="MINMAXMAMDANI" ip="localhost">
    <RULE connector="AND" ip="localhost" weight="1" id="R1">
      <ANTECEDENT>
        <CLAUSEA not="FALSE">
          <VARIABLE>MoveEvaluation</VARIABLE>
          <TERM>VeryGood</TERM>
        </CLAUSEA>
        <CLAUSEA not="FALSE">
          <VARIABLE>AffectiveEvaluation</VARIABLE>
          <TERM>Happy</TERM>
        </CLAUSEA>
        <CLAUSEA not="FALSE">
          <VARIABLE>ThinkingTime</VARIABLE>
          <TERM>Short</TERM>
        </CLAUSEA>
        <CLAUSEA not="FALSE">
          <VARIABLE>GameTime</VARIABLE>
          <TERM>Short</TERM>
        </CLAUSEA>
        <CLAUSEA not="FALSE">
          <VARIABLE>GameNo</VARIABLE>
          <TERM>Short</TERM>
        </CLAUSEA>
      </ANTECEDENT>
      <CONSEQUENT>
        <CLAUSEC not="FALSE">
          <VARIABLE>WinningRate</VARIABLE>
          <TERM>VeryHigh</TERM>
        </CLAUSEC>
      </CONSEQUENT>
    </RULE>
…
  </RULEBASE>
</FUZZYCONTROL>
```

C. *Type-2 Fuzzy Set Inference Mechanism*

Herein, the fuzzy ontology repository stores the established knowledge base and rule base that are provided by domain experts. Additionally, the Karnik-Mendel (KM) algorithms [19] are adopted to compute the centroids of all of the embedded T1FSs. Table II lists the parameters of the type-2 membership functions that are determined by the domain experts. Figures 5(a), 5(b), 5(c), 5(d), 5(e), and 5(f) shows the membership functions of the fuzzy variables *ME*, *AE*, *TT*, *GT*, *GN*, and *WR*, respectively.

TABLE II. PARAMETERS OF THE TYPE-2 FUZZY SET.

Fuzzy Variable	Linguistic Term	$T_L = [l_L, m_{l_L}, m_{r_L}, r_L]$
		$T_U = [l_U, m_{l_U}, m_{r_U}, r_U]$
Move Evaluation (ME)	*VeryGood*	[75,85,100,100]
		[80,85,100,100]
	Good	[50,70,75,85]
		[60,70,75,80]
	Medium	[25,45,55,75]
		[30,45,55,70]
	Bad	[10,20,30,50]
		[15,20,30,40]
	VeryBad	[0,0,15,25]
		[0,0,15,20]
Affective Evaluation (AE)	*Happy*	[60,95,100,100]
		[70,95,100,100]
	Emotionless	[25,35,80,90]
		[30,35,80,85]
	Thinking	[0,0,30,50]
		[0,0,30,40]
Thinking Time (TT)	*Long*	[45,95,100,100]
		[80,95,100,100]
	Medium	[25,35,45,75]
		[30,35,45,65]
	Short	[0,0,20,30]
		[0,0,20,25]
Game Time (GT)	*Long*	[60,80,120,120]
		[70,80,120,120]
	Medium	[40,50,60,80]
		[50,60,60,70]
	Short	[0,0,25,50]
		[0,0,25,40]
Game No (GN)	*Long*	[5,6,8,8]
		[5,6,8,8]
	Medium	[2,4,4,6]
		[3,4,4,5]
	Short	[0,0,1,4]
		[0,0,1,3]
Winning Rate (WR)	*VeryHigh*	[75,85,100,100]
		[80,85,100,100]
	High	[50,65,75,85]
		[55,65,75,80]
	Medium	[25,45,55,75]
		[30,45,55,70]
	Low	[12,20,30,50]
		[15,20,30,40]
	VeryLow	[0,0,10,25]
		[0,0,10,22]

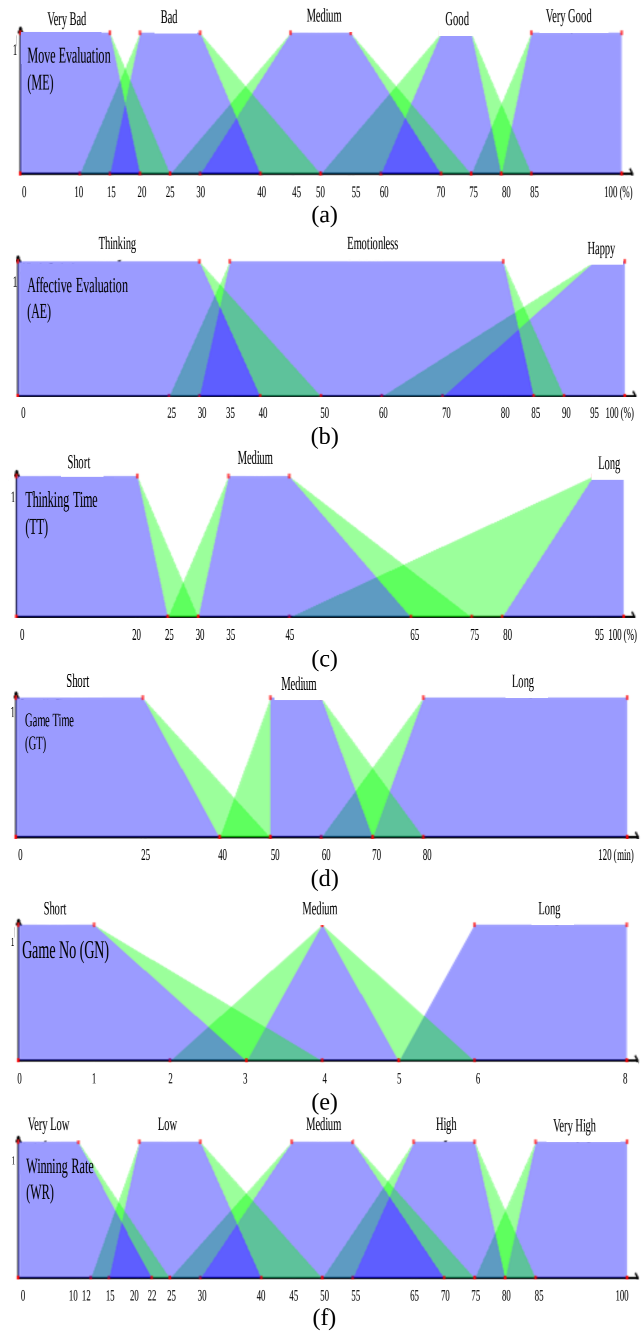

Figure 5. Membership functions of the fuzzy variables (a) *ME*, (b) *AE*, (c) *TT*, (d) *GT*, (e) *GN*, and (f) *WR*.

V. CONCLUSION

The paper proposes an FML-based type-2 fuzzy ontology for computer Go knowledge representation. The invited Go players give their comments on the bad moves via the Multi-Go software. Then, the FML transformation mechanism changes SGF files into the FML document. The fuzzy ontology experts construct the fuzzy ontology through the type-2 fuzzy set construction mechanism. Based on the fuzzy ontology repository, the type-2 fuzzy set inference mechanism infers the winning rate of the game. In this future, the experimental results will be shown to illustrate its effectiveness to demonstrate that the winning rate of the game will be increased based on the proposed approach. It is hoped that the proposed idea are feasible for inferring the winning rate of the games in the future.

ACKNOWLEDGMENT

This work is supported by the National Science Council of Taiwan under the grant NCS99-2923-E-024-003-MY3, NSC97-2221-E-024-011-MY2, NSC 98-2631-S-024-001, and NSC98-2221-E-024-009-MY3.

REFERENCES

[1] B. Ghosh and J. E. Scott, “Comparing knowledge management in health-care and technical support organizations,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 9, no. 2, pp. 162-168, 2005.

[2] P. Warren, “Knowledge management and the semantic web: from scenario to technology,” *IEEE Intelligent Systems*, vol. 21, no. 1, pp. 53-59, 2006.

[3] M. Reformat and C. Ly, “Ontological approach to development of computing with words based systems,” *International Journal of Approximate Reasoning*, vol. 50, no. 1, pp. 72-91, 2009.

[4] Q. T. Tho, S. C. Hui, A. C. M. Fong, and T. H. Cao, “Automatic fuzzy ontology generation for semantic web,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 6, pp. 842-856, 2006.

[5] C. S. Lee, Z. W. Jian, and L. K. Huang, “A fuzzy ontology and its application to news summarization,” *IEEE Transactions on Systems, Man and Cybernetics Part B*, vol. 35, no. 5, pp. 859-880, 2005.

[6] C. S. Lee, M. H. Wang, and J. J. Chen, “Ontology-based intelligent decision support agent for CMMI project monitoring and control,” *International Journal of Approximate Reasoning*, vol. 48, no. 1, pp. 62-76, 2008.

[7] H. Hagras, “Type-2 FLCs: A new Generation of Fuzzy Controllers,” *IEEE Computational Intelligence Magazine*, vol. 2, no. 1, pp. 30-43, February 2007.

[8] T. T. Quan, S. C. Hui, and A. C. M. Fong, “Automatic fuzzy ontology generation for semantic help-desk support,” *IEEE Transactions on Industrial Informatics*, vol. 2, no. 3, pp. 155-164, 2006.

[9] R. Knappe, H. Bulskov, and T. Andreasen, “Perspectives on ontology-based querying,” *International Journal of Intelligent Systems*, vol. 22, no. 7, pp. 739-761, 2007.

[10] S. Calegari and F. Farina, “Fuzzy ontologies and scale-free networks analysis,” *International Journal of Computer Science & Applications*, vol. 4, no. 2, pp. 125-144. 2007.

[11] C. Hudelot, J. Atif, and I. Bloch, “Fuzzy spatial relation ontology for image interpretation,” *Fuzzy Sets and Systems*, vol. 159, no. 15, pp. 1929-1951, August 2008.

[12] T. T. Quan, S. C. Hui, A. C. M. Fong, and T. H. Cao, “Automatic fuzzy ontology generation for semantic web,” *IEEE Transaction on Knowledge and Data Engineering*, vol. 18, no. 6, pp. 842-856, June 2006.

[13] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New directions*, Upper Saddle River, NJ: Prentice-Hall, 2001.

[14] C. S. Lee, M. H. Wang, and H. Hagras, "A type-2 fuzzy ontology and its application to personal diabetic-diet recommendation," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 2, pp. 374-395, 2010.

[15] G. Acampora G and V. Loia, “Using FML and fuzzy technology in adaptive ambient intelligence environments,” *International Journal of Computational Intelligence Research*, vol. 1, no. 2, pp. 171-182, 2005.

[16] G. Acampora G and V. Loia, “Fuzzy control interoperability and scalability for adaptive domotic framework,” *IEEE Transactions on Industrial Informatics*, vol. 1, no. 2, pp. 97-111, 2005.

[17] M. H. Wang, C. S. Lee, K. L. Hsieh, C. Y. Hsu, G. Acampora, and C. C. Chang, "Ontology-based multi-agents for intelligent healthcare applications," *Journal of Ambient Intelligence and Humanized Computing*, 2010. (DOI: 10.1007/s12652-010-0011-5)

[18] C. S. Lee, M. H. Wang, T. P. Hong, G. Chaslot, J. B. Hoock, A. Rimmel, O. Teytaud, and Y. H. Kuo, “A novel ontology for computer Go knowledge management,” in *Proceeding of the 2009 IEEE International Conference on Fuzzy System* (*FUZZ-IEEE 2009*), Jeju Island, Korea, Aug. 19-24, 2009, pp. 1056-1061.

[19] J. M. Mendel, “Type-2 fuzzy sets and systems: an overview,” *IEEE Computational Intelligence Magazine*, vol. 2, no. 1, pp. 20-29, 2007.