OAV-VVT Expert, an Active System for Verification and Validation of Knowledge Base Using OAV Knowledge Representation

Chitra Dadkhah    Ahmad Abdollahzadeh Barforosh
Computer Engineering Department, Amirkhabir University of Technology
Tehran, Iran

Abstract

In this paper, we present an active expert system (OAV-VVT expert) to verify and validate the knowledge base (KB). Our work is based on a new approach to representation and reasoning techniques, E_{x}-OAV KB, based on Object-Attribute-Value (OAV) knowledge representation. All common issues of verification and validation are considered in OAV-VVT expert. OAV-VVT expert can be either used during the building of knowledge base or refinement of existing KB. OAV-VVT expert is an expert system with reasoning and explanation mechanisms. It is application and knowledge representation independent. OAV-VVT expert improves the performance of V&V phase in building the knowledge base by checking unreference attribute values and illegal attribute values, during transforming the existing KB to E_{x}-OAV KB. All necessary algorithms to transform different knowledge representation techniques have been presented in this paper to justify that the E_{x}-OAV KB is feasible.

Keyword: verification and validation, knowledge base, knowledge representation, OAV triple, expert systems, AI systems

1. Introduction

Verification and Validation (V&V) are the most crucial phases in building knowledge base systems. Researchers and practitioners in the field of AI applications generally agreed that knowledge base of any intelligent systems must be adequately verified and validated. To build an AI system, the quality of KB is crucial. The need for an integrated approach towards V&V of knowledge base is quite obvious and in this paper we present an active OAV-VVT expert to verify and validate the knowledge during the building of knowledge base or refinement of existing KB. In order to build such a V&V expert tool, it is necessary to have a common definition of V&V. Table 1 shows different approaches to the definition of V&V process in knowledge base system. We define “validation” as the correctness and existence of system output according to the test cases that have been defined by expert and “verification” as the checking of the consistency, completeness and logical & semantic contradiction of KB. All of these definitions are presented by logical formal language. Consequently with this definition, OAV-VVT expert will cover all the common issues in V&V of the knowledge. Majority of the previous works in this field are concentrated on the verification of KB and less attention has been paid to the validation. The evaluation of system’s output based on expert opinion is very complex and researchers have limitedly dealt with the issues of completeness, consistency or correctness in certain KB systems as expert system [5, 8, 21, 23, 30, 32]. We categorize the researches work in the field of V&V of the KB as below:

1. Representation of a new method for description of knowledge [27, 29, 34].
Table 1. The definitions of V&V by the researchers and practitioners

<table>
<thead>
<tr>
<th>References</th>
<th>Validation</th>
<th>Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>. The verification of the overall performance of the expert system.</td>
<td></td>
</tr>
<tr>
<td>MENSHOEL &amp; DELAB [3]</td>
<td>Addresses software system’s usefulness with respect to some real-world task, regardless of its specification.</td>
<td>Concerns a software system’s conformance to its specification.</td>
</tr>
<tr>
<td>KANDEL &amp; SMITH [4]</td>
<td>A test of whether the Expert System matches the design ideas, i.e., whether it matches the technical requirements and expectations.</td>
<td>Examine the technical aspects of an Expert System in order to determine whether the Expert System is built correctly.</td>
</tr>
<tr>
<td>WU &amp; al [5]</td>
<td>The correctness of inference without considering the Input/Output.</td>
<td>The correctness of systems outputs according to specific Inputs.</td>
</tr>
<tr>
<td>ROUSSET &amp; al [6]</td>
<td>The correctness of system’s output depends on Test cases and Initial valid Fact Base.</td>
<td>Examining the logical and semantic contradiction of Rule Base.</td>
</tr>
<tr>
<td>TSAI &amp;al [7]</td>
<td>The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirement.</td>
<td>1) The process of evaluating a system or component to determine whether the products of the given phase satisfy the conditions imposed at the start of that phase. 2) Formal proof of program correctness.</td>
</tr>
<tr>
<td>KNAUF &amp; al [8]</td>
<td>Asks whether or not a system is considered to be the required one, something that somehow lies in the eyes of the beholder.</td>
<td>Provides a firm basis for the question of whether or not a system meets its specifications.</td>
</tr>
</tbody>
</table>

2. Representation of a new algorithm to process [24], [28].
3. Development or extension of an existing method of process [26, 28].
4. Design a new structure for construction of the KB to simplify the V&V techniques [22, 27, 29].
5. Definition of new issues for V&V of the KB and its related solution, apart from completeness, consistency or correctness [18, 31].

All the previous research works carried out in this field are included in [38].

In OAV-VVT expert, we introduce $E_x$-OAV KB based on new approach to representation and reasoning techniques to create new KB from the existing KB. $E_x$-OAV KB is based on Object-Attribute-Value(OAV) Knowledge representation. We will present and demonstrate how the major techniques of knowledge representation (i.e. semantic network, frames, production rules) could be transformed to $E_x$-OAV KB. In our work the syntax of knowledge is changed but the semantic of knowledge has been remained unchanged. $E_x$-OAV KB will be verified and validated by OAV-VVT expert and this process is iterative and all the invalid knowledge will be recorded and then modified. The process of modification of invalid knowledge in OAV-VVT expert will be done by using the existing V&V methods [18], [35, 36, 40, 41] or knowledge engineer. This process will be completed when not new invalid knowledge is discovered in $E_x$-OAV KB and the process will start as soon...
as the new knowledge submits to the E_x-OAV KB. The knowledge flow in OAV-VVT expert is shown in Figure 1. OAV-VVT expert by using the explanation component, which is an essential part in any expert system, can explain “how” an invalid Knowledge has been discovered and “why” it is invalid. All common issues of verification and validation (V&V) such as consistency, completeness, logical & semantic contradiction and correctness of knowledge base are considered in OAV-VVT expert.

We present the design and implementation of the OAV-VVT expert. OAV-VVT expert is a tool to V&V of knowledge during the submitting a new knowledge or refinement of the existing knowledge base.

The structure of this paper is as follows:

• Section 2 describes the structure of E_x-OAV KB and present procedures to transform the major knowledge representation techniques to E_x-OAV KB.
• Section 3 defines the issues of V&V in a formal specification language.
• Section 4 presents the architecture and reasoning technique of OAV-VVT expert.
• Section 5 reports the implementation phase and describes detail information about OAV-VVT expert.
• Section 6 presents the evaluation of OAV-VVT expert.
• Section 7 will present the conclusion of this work.

2. The Structure of E_x-OAV KB

Object – Attribute – Value (OAV) provides a particularly convenient way in which to represent certain facts and heuristic rules within the KB. Each OAV triple is present with specific entity or object and a set of attributes with their values associated to every object. [2]

All knowledge in Ex - OAV KB is Rule Base System using OAV triple. The structure of E_x -OAV KB is shown in Figure 2.

In the next section we present the process of transforming the major knowledge representation techniques to E_x-OAV KB.
Example 2. The semantic network with the HAS relationship

Example 3. The rule-based system

2.1.2 Production rules to E_X -OAV KB

Rule-based technique of knowledge representation is known as production rule in literature [2]. Such rules are typically of the IF-THEN format. The Example 3 shows an example of the Rule-based system.

The premise and conclusion of rules contain attributes and values (AV) associated to an object either explicit or implicit as below:

Rule 1: If student.GREscore >= 1350
Then admission.status = yes

Rule 2: If GPA >= 3.5
Then accept = into honor society

We can simply define an OAV triple for each premise and conclusion and when there is an implicit object we use the system KB for finding the associated object to AV pairs. The OAV triple associated to the Example 3 are shown in Figure 7.

Figure 5 shows the final result of transforming the KB of example 1 to OAV triple.

2.1.3 Frame to E_X -OAV KB

The Frames are a robust way of presenting knowledge. A frame presents all the available fact as an object plus slots and value for all the information related to the objects. A frame has a name, slots with labels describing the concept’s major attributes or features and possible values for each attributes [2,37]. The Example 4 illustrates a frame-based knowledge representation (FB KR) for the earthquake.

We can simply define an OAV triple for each premise and conclusion and when there is an implicit object we use the system KB for finding the associated object to AV pair. The Figure 6 presents ” rbs-to-oav “ procedure for transforming the rule into an OAV triple, which will be active in event of receiving new rule. The function “Find-

obj” in this procedure searches in the KB systems for finding the object related to AV pairs. The OAV triple associated to the Example 3 are shown in Figure 7.

Figure 6. The procedure for transforming the rule to OAV triple

Figure 7. The OAV triple associated to Example 3

Figure 8 shows the procedure for transforming FB KR to E_X-OAV KB. Figure 9 illustrates the OAV triple associated to the Example 4.

A frame may have attached procedures that capture procedural information about the concept. A facet can be used to define constraints on the property value or to execute procedure for obtaining the property value or what to do if the value changes.
Example 4. A frame-based knowledge representation for Earthquake

<table>
<thead>
<tr>
<th>Earthquake:Time</th>
<th>8.45</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earthquake:Day</td>
<td>Today</td>
</tr>
<tr>
<td>Earthquake:Place</td>
<td>Lower-slabovia</td>
</tr>
<tr>
<td>Earthquake:Damage</td>
<td>500/000/000</td>
</tr>
<tr>
<td>Earthquake:Fatalities</td>
<td>25</td>
</tr>
<tr>
<td>Earthquake:Fault</td>
<td>Dadie-Hawkins</td>
</tr>
<tr>
<td>Earthquake:Magnitude</td>
<td>8.5</td>
</tr>
</tbody>
</table>

Example 5. The frame-based for the canary object

The “@” sign means that a method must be executed to assign or change the value of the associated object attribute. By calling the “rbs-to-oav” function, we transform the method to OAV triple and it will be recorded in OAV-VVT KB. Figure 13 shows the OAV triple related to the Example 5. OAV-VVT expert infers the new knowledge from OAV-VVT KB and assign or change the value of property during the reasoning. For example, if we have a fact like “canary.wing = 2” then OAV-VVT expert assigns the “True” value to “fly” attribute of “canary” object by reasoning strategy.

Function facet-to-oav (Objectname, {[property, {facet}]}) return (OAV-Knowledge)

Input: Objectname, The name of object
Output: {payload}, The list of OAV triple associated to Frame

For each (p, v) in {[property, {facet}]}

- property ← p
- For each (s, v) in {facet}
  - attribute ← s
  - operator ← “=”
  - case (s)
    - “Type” : Value ← v
    - “if-needed” or “if-changed” :
      - Begin
        - attribute ← property
        - value ← @ v
      - end
      - otherwise: Value ← v
      - OAV-Knowledge ← <Object> .<Attribute> <operator> <value>
      - end
      - Add OAV-knowledge to {OAV-knowledge}

end
Figure 11. The procedure for transforming the facets to OAV triple

<table>
<thead>
<tr>
<th>Function</th>
<th>met-to-oav (object,property,metname, {ifpart},{thenpart}) return ( {if-OAV-Knowledge}, {then-OAV-knowledge})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input:</td>
<td>Metname, The name of method Object, the name of object Property, the name of property {ifpart}, The list of attribute value pair of premises {thenpart}, The list of attribute value pair of conclusion</td>
</tr>
<tr>
<td>Output:</td>
<td>The abbreviations used in the OAV-VVT expert</td>
</tr>
<tr>
<td>For each</td>
<td>(A,O,V) in {ifpart} do rbs-to-oav (object,property,O,V,OAV-knowledge) Add OAV-knowledge to {if-OAV-knowledge}</td>
</tr>
<tr>
<td>End</td>
<td>For each (A1,O1,V1) in {thenpart} do rbs-to-oav (object,property,O1,V1,OAV-knowledge) Add OAV-knowledge to {then-OAV-knowledge}</td>
</tr>
<tr>
<td>End</td>
<td>Add ( {if-OAV-knowledge}, {then-OAV-knowledge} ) TO E_o OAV KB</td>
</tr>
<tr>
<td>Return</td>
<td>( {if-OAV-knowledge}, {then-OAV-knowledge} )</td>
</tr>
</tbody>
</table>

Figure 12. The procedure for transforming the method to OAV triple

Canary.min-cardinality = 2
Canary.type = ?boolean
Canary.fly = @canary-fly

Figure 13. The OAV triple associated to the Example 5

3. The Issues of V&V

The V&V activities which has been addressed in OAV-VVT expert are the following steps: [11, 12, 13, 14, 15, 16, 17].

- Completeness checking.
- Consistency checking.
- Logical & semantic contradiction checking.
- Validation checking.

In order to be able to define a detail description in OAV-VVT expert, we introduce the abbreviations based on the work of Bouali and Ignizio [18, 33] as shown in Figure 14. It shows the abbreviations in 3 sections. The first, second and third sections are based on the work of Bouali, Ignizio and ours respectively.

**Def1:** c (R) is an unachievable intermediate conclusions iff

\[ \{ c(R) \neq \{G\}, \forall S | S \in RB, S \neq R, c(R) \notin \{P(S)\} \} \]

**Def2:** c (R) is an unachievable intermediate goal iff

\[ \{ c(R) = \{G\}, \forall S, p(R) | S \in RB, S \neq R, p(R) \notin \{P(R)\}, (P(R) \notin \{C(S)\}) \}

**Def3:** p (R) is an unachievable premise iff

\[ \forall S, | S \in RB, S \neq R, p(R) \in \{C(S)\}, p(R) \notin \{ask(voc)\} \]

**Def4:** V is an unreferenced attribute value associated to attribute “A” of object “O” if

\[ \{ V \notin \{VAL(O/A)\}, V \notin \{OB-LST\} \}

**Def5:** V is an illegal attribute value associated to attribute “A” of object “O” if

\[ \exists V | V \in \{VAL(O/A)\}, V \notin \{OB-LST\} \]

Our definitions of the five types of incompleteness issues are as follows:

**Def6:** Rule r is redundant rule if

\[ \exists s | s \in RB, (P(r) = P(s)), c(r) = c(s) \]

**Def7:** it is a logical contradiction between rule r,s if

\[ \{P(r) = P(s), (O.A = V1) \subseteq c(r), (O.A = V2) \subseteq c(s), V1 \neq V2 \}

**Def8:** Rule r is subsumed by rule s if

\[ c(r) = c(s), \{P(s) \subseteq P(r)\} \]

**Def9:** A set of rules, \( r_1, r_2, r_3, \ldots, r_n \) are circular rules if

\[ c(r_i) \in \{G\}, c(r_{i+1}) \in p(r_i), c(r_{i+2}) \in p(r_{i+1}), \ldots, c(r_{i+n}) \in p(r_{i+n-1}), p(r_{i+1}) \subseteq c(r_{i+n}) \]

**Def10:** Rule r contains unnecessary premise clause if

\[ \exists s | s \in RB, c(r) = c(s), \exists p1, p2 | p1 \in \{P(r)\}, p1 = \{O1.A1 = V1\}, p2 \in \{P(s)\}, p2 = \{O1.A1 = V2\}, V1 \neq V2, p1 \neq p2, \forall pi | p_i \neq p1, p_i \in \{P(r)\}, p_i \in \{P(s)\} \]
The “∇” and “*” symbols introduce the logical & semantic contradiction in Def11 and Def12.

**Def11:** $E_o$-OAV KB contains a logical contradiction iff $\forall \in (LCxFB)$.  

**Def12:** $E_o$-OAV KB contains a semantic contradiction iff $* \in (SCxFB)$.  

We define three types of Validation that may be detected by the OAV-VVT expert as shown below:

**Def13:** $E_X$-OAV KB has a Contradiction output if  
\[ \forall \text{tstcase} \in \{ \text{TSTCASE} \}, \text{tstcase} = (FB1, output), \text{RBxFB}1 = \neq \text{output} \]

**Def14:** $E_X$-OAV KB has any output if  
\[ \forall \text{tstcase} \in \{ \text{TSTCASE} \}, \text{tstcase} = (FB1, output), \text{RBxFB}1 = \{ \} \]

**Def15:** $E_X$-OAV KB has an incorrect output if  
\[ \forall \text{tstcase} \in \{ \text{TSTCASE} \}, \text{tstcase} = (FB1, output), \text{RBxFB}1 \neq \text{output} \]

### 4. The Architecture of OAV-VVT Expert

The OAV-VVT expert contains knowledge base, reasoning and explanation components. The architecture of knowledge flow of OAV-VVT expert is illustrated in Figure 15. In this section, first the steps involved to apply V&V algorithm are presented and then the KB and definition of logical and semantic contradiction are presented. Based on KB and these definitions, the description of reasoning and explanation are followed.

The V&V algorithm of OAV-VVT expert contains some steps as shown below:

1. Define the test cases (facts/goal) by the expert and the set of LC & SC. Transform the existing KB to the $E_X$-OAV KB.
2. Completeness checking of $E_X$-OAV KB.
3. If the OAV-VVT expert finds any invalid knowledge from step 2, knowledge engineer records the invalid knowledge and modifies the $E_X$-OAV KB accordingly and then go to step 2, otherwise go to step 4.
4. Consistency checking of $E_X$-OAV KB.
5. If the OAV-VVT expert finds any invalid knowledge from step 4, Knowledge engineer records the invalid knowledge and modifies the $E_X$-OAV KB and then go to step 2, otherwise go to the step 6.
6. Logical & semantic contradiction checking of $E_X$-OAV KB.
7. If the OAV-VVT expert finds any invalid knowledge from step 6, Knowledge engineer records the invalid knowledge and modifies the $E_X$-OAV KB accordingly and then go to step 2, otherwise go to step 8.
8. Validation checking of $E_X$-OAV KB.
9. If the OAV-VVT expert finds any invalid knowledge from step 8, Knowledge engineer modifies the $E_X$-OAV KB and then goes to step 2, otherwise go to step 10.
10. Stop.

The KB of OAV-VVT contains following rules:

$$LC = \{ LC1 : IF \langle \text{?O.?A} = ?V \rangle \& \langle \text{?O.?A} \neq ?V \rangle \text{ THEN } \neq \}$$

$$LC2 : IF \langle \text{?O.?A} < ?V \rangle \& \langle \text{?O.?A} = ?V \rangle \text{ THEN } \neq \}$$

$$LC3 : IF \langle \text{?O.?A} > ?V \rangle \& \langle \text{?O.?A} = ?V \rangle \text{ THEN } \neq \}$$

$$LC4 : IF \langle \text{?O.?A} < ?V \rangle \& \langle \text{?O.?A} > ?V \rangle \text{ THEN } \neq \}$$

$$LC5 : IF \langle \text{?O.?A} = ?V1 \rangle \& \langle \text{?O.?A} = ?V2 \rangle \& \langle \text{V1} \neq \text{V2} \rangle \text{ THEN } \neq \}$$

$$LC6 : IF \langle \text{?O.?A} < ?V1 \rangle \& \langle \text{?O.?A} = ?V2 \rangle \& \langle \text{V1} \neq \text{V2} \rangle \text{ THEN } \neq \}$$

$$LC7 : IF \langle \text{?O.?A} > ?V1 \rangle \& \langle \text{?O.?A} = ?V2 \rangle \& \langle \text{V1} \neq \text{V2} \rangle \text{ THEN } \neq \}$$

$$LC8 : IF \langle \text{?O.?A} < ?V1 \rangle \& \langle \text{?O.?A} > ?V2 \rangle \& \langle \text{V1} < \text{V2} \rangle \text{ THEN } \neq \}$$

It contains the rules for methods associated to facet as explained in the section 2.1.3. We also record the semantic contradiction rule (Rule RC) in KB of OAV-VVT expert.

**Rule SC:** If man ^ Pregnant  
Then *

Figure 15. The architecture of OAV-VVT expert

The reasoning of OAV-VVT expert uses the backward strategy in order to discover the invalid knowledge in $E_o$-OAV KB based on our V&V definition as defined in the section 3. We applied the backward chaining for all goals, which are detailed in application domain, to find all associated facts. All the facts that resulted from goals will be recorded in a list. If these facts do not match the testcases based on Def13, Def14, Def15, then OAV-VVT expert discovers an invalid knowledge. Previous works shows that it is possible to have more than one list for a goal [18, 19] and we will checked all of lists. If the facts of these lists trigger the rules of KB then the system discovers any logical & semantic contradiction based on LC, SC rules.

Whenever the OAV-VVT expert discovers any invalid knowledge, it is able to explain how the system discovers this invalid knowledge and show the trace of discovering it. As the process of V&V in OAV-VVT expert is iterative and dynamic, we consider a threshold (some iteration of V&V
process) for copying the $E_x$-OAV KB to $E_x$-OAV KB dual. The application could use the $E_x$-OAV KB dual any time and the system continues the V&V of $E_x$-OAV KB. Whenever the algorithm of V&V in OAV-VVT expert stops, we will have an $E_x$-OAV valid KB.

5. Implementation

The prerequisites, which have been described in OAV-VVT expert, are as follow:

- All the rules of canonical KB ($E_x$-OAV KB) are backward chaining rules.
- The rule set is deterministic (there is no confidence factor).
- Only conjunctive clauses are employed in the premise and a single clause appears in the conclusion.

We designed a user-friendly tool to transform the semantic network or frame, to the $E_x$-OAV KB (Figure 16). We implemented this tool with Visual Basic 2000. We could draw and edit the semantic network or frame and construct the KB with this tool. This tool uses the standard algorithms that have been described in section 2 for transforming the knowledges to OAV triple. The transformed knowledges are shown in three forms: OAV triple, OAV hierarchy and OAV table.

Figure 16 shows the OAV triple associated to the KB of Figure 16. The OAV hierarchy and the OAV table of the constructed KB with this tool are shown in Figure 18 and figure 19.

Finally, the tool will be constructed the knowledge base which its knowledges are represented in OAV triple as shown below:

- $\text{bird(fulmar)} :- \text{voice(loud)}$, $\text{family(swan)}$.
- $\text{family(swan)} :- \text{order(waterfowl)}$, $\text{neck(long)}$, $\text{color(white)}$, $\text{can(ponderous)}$.
- $\text{order(waterfowl)} :- \text{feet(webbed)}$, $\text{bill(flat)}$. 

Figure 17. The OAV triple associated to the Figure 16

Figure 18. The OAV hierarchy associated to the Figure 16

Figure 19. The OAV table associated to the Figure 16
This KB is ready to be used by any expert system shell for reasoning or the verification and validation tools. OAV_VVT expert, could verify and validate the knowledge of this knowledge base as shown below:

You can choose one of the categories of V&V issues.

******V&VTool***************
Choose one of category:
1. Completeness
2. Consistency
3. Logical Contradiction
4. View Rule
5. Edit Rule
6. Exit

*********V&VTool*************
Choose one of category:
1. Illegal Value
2. Unreferenced Attribute Value
3. Unachievavle Premises
4. Unachievavle Intermediate Conclusion
5. Unachievavle Goal
6. Return

item(1-6)?1

Choose one of category:
1. Completeness, consistency, correctness, semantic & logical contradiction, validation
2. It is able to check unreference and illegal attribute value at the transforming the existing KB to E_x-OAV KB to increase the performance.
3. The building of knowledge base or refinement of existing knowledge base
4. Dynamic
5. The V&V process will be stopped when not new invalid knowledge is discovered in E_x-OAV KB. The system copies the E_x-OAV KB to the dual KB for using the applications after some iteration.
6. It can explain “why” and “how” the invalid knowledge has been discovered.

5. Evaluation

The evaluation of V&V tool is based on the evaluation parameters. We are defined some evaluation parameters as shown bellow:

- Kind of KB: the knowledge representation techniques that the tool could verify and validate.
- Issues of V&V: The issues of V&V that the tool could check.
- Speed: The performance of V&V process to discover an invalid knowledge
- V&V process time: When the tool could verify and validate the knowledge?
- V&V process mode: The mode of V&V process can be dynamic or static.
- Stop time: If the V&C process mode is dynamic, when the V&V process will stop?
- Explanation technique: Is the tool able to explain “how” and “why” the invalid knowledge has been discovered?

Table 2 shows the value of each evaluation parameters of OAV-VVT expert tool in a qualitative format.

<table>
<thead>
<tr>
<th>Evaluation Parameters</th>
<th>OAV-VVT Expert</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kind of KB</td>
<td>Semantic network, Rule-base, Frame</td>
</tr>
</tbody>
</table>

Table 2. The evaluation parameters for OAV-VVT expert tool

![EVALUATION RESULTS](image)

Figure 20. The evaluation chart of V&V tools.

7. Conclusions

This paper addresses the verification and validation of knowledge in KB systems. In previous works less attention has been paid to present a general procedure to cover all
knowledge representation techniques, the major objective of our work was to design an active expert system to V&V all KB based on different knowledge representation techniques in AI systems. The important aspects of our work are:

- Defining a canonical knowledge base (E_x-OAV KB) and necessary algorithms to convert all knowledge representation techniques to E_x-OAV KB. Having all knowledge in one knowledge representation technique make the system independent to knowledge representation technique. It is also application and domain independent.
- OAV-VVT expert in addition of covering all common issues of V&V considers the semantic & logical contradiction too.
- OAV-VVT expert consists reasoning component in order to make the tool more powerful.
- OAV-VVT expert can be either used during the building of knowledge base or refinement of existing knowledge base.
- OAV-VVT expert consists an explanation component in order to describe and answer “how” and “why”, in presence of invalid knowledge in KB.
- OAV-VVT expert is able to check the illegal attribute value as well as unreferenced attribute value during the transforming of the existing KB to E_x-OAV KB to increase the performance of V&V.

In this work, OAV-VVT expert is presented as a V&V tool with new approach, E_x-OAV KB, which is concentrate on OAV knowledge representation technique. OAV-VVT expert is a good test bed for future work and use the ontology for finding the semantic contradiction. The issue of ambiguity in V&V should consider in future work.

OAV-VVT expert can be either used during the building of knowledge base or refinement of existing knowledge base.

- OAV-VVT expert consists an explanation component in order to describe and answer “how” and “why”, in presence of invalid knowledge in KB.
- OAV-VVT expert is able to check the illegal attribute value as well as unreferenced attribute value during the transforming of the existing KB to E_x-OAV KB to increase the performance of V&V.

In this work, OAV-VVT expert is presented as a V&V tool with new approach, E_x-OAV KB, which is concentrate on OAV knowledge representation technique.

OAV-VVT expert is a good test bed for future work and use the ontology for finding the semantic contradiction. The issue of ambiguity in V&V should consider in future work by classifying the invalid knowledge in different categories.

References


Chitra Dadkhah received the B.S degree in Software Engineering from Shahid Beheshti University, Tehran, Iran in 1990. she received the Management Information System Courses of M.S degree from Azad University, Tehran, Iran in 1992. She also received the M.S degree in Computer Engineering (Artificial Intelligence) from Paris XI (Orsay) University , Paris, France in 1994. At present, she is a Ph.D student in the Computer Engineering Department at Amirkabir University of Technology in Tehran, Iran. Her research interests include Expert System, Knowledge Representation Techniques, Artificial Intelligent Techniques, Object-Attribute-Value Triple, Verification & Validation of the Knowledge.

Email: Dadkhah@ce.aut.ac.ir

Ahmad Abdollahzadeh Barforosh received the B.S degree in accounting from Tehran University in Tehran, Iran in 1975. He received the M.S degrees in Computer science from West Coast University, Los Angeles, california, U.S.A in 1980. He also received the Ph.D in computer science from university of Bristol,England,1990.

Dr.Abdollahzadeh serves as visiting professor in department of computer science of Maryland university at college park, USA and Orsay university in Paris, France from 2000-2002. Currently, he is a associate professor in the IT & Computer Engineering Departament at AmirKabir University of Technology in Tehran, Iran. His research interests include, Information Retrieval, Artificial Intelligence Techniques, Expert System, Natural Language Proceesing, Decision Support System, Knowledge Representation, Software Engineering.

Email: ahmad@ce.aut.ac.ir
