The Engineering of expert systems testing process

ABEER EL-KORANY
Computer Science Department
Faculty of computer & information, Cairo University
5 Dr Ahmed Zoweil St., Orman, Giza 12613
Egypt

Abstract: - Every expert system must continuously evolve, or it will become obsolete. This paper explores a successive testing process for evolutionary expert systems. The proposed test process is based upon applying software engineering model for overall life cycle of test process. The proposed testing process has two major complementary components: testing and maintenance. Each plays an important role in ensuring the quality of the developed expert systems. The characteristics of each phase are described in terms of testing capability and goals. The present work reports an accumulated experience gained through many years of an iterative process of developing, evaluating, deploying, and updating several expert system using CommonKADS knowledge engineering methodology at the agricultural domain at the Central Laboratory for Agricultural Expert systems (CLAES), Egypt.

Key-Words: - Expert systems, commonKADS, testing process, software engineering, corrective maintenance

1 Introduction

There is abundant evidence of the need for an integrated approach toward testing of expert systems (ES) [1]. New trends in research are focused on technologies to accompany the system development by an integrated testing process. Because of the important role of testing in expert systems development lifecycle and the limitations of existing process assessment models, we have focused our research efforts in the direction of providing a complete, evolutionary test process model. One fundamental characteristic of all expert systems is the clear and clean separation between the knowledge that the system is using and the program that utilize it for problem solving. Therefore, we propose a complete testing model that is capable of testing the expert systems product as well as ensure the validity of its knowledge.

The proposed testing model is based on agreed software engineering principles and practices. It allows the development of testing process in structured stepwise phases that follows natural process evolution. The aim of applying these subsequent phases is to improve the entire system quality. A set of levels that defines a testing process is provided. Each level represents a stage in the evolution to a mature testing process and characterizes by its activities, tasks and goals.

One principal reason in limitation of expert systems acceptance is the difficulty in maintaining such systems due to the nature of expert knowledge [3]. Maintenance aims at keeping the KB of the same quality level even after supply it to end-users. Therefore, we propose maintenance as a key partner to the expert systems testing in order to obtain a complete testing environment.

The trend of commonKADS development methodology [20] used in building KBs out from pre-defined components triggers our proposed approach for testing and maintenance. It makes it possible to derive the structure of the KB from sub components allowing each KB component to be tested before it is used. The proposed methodology is generally applicable to hybrid knowledge representation schemes that are common in many operational expert systems developed by the Central Laboratory for Agricultural Expert systemss (CLAES) such as concepts, rules, tables and mathematical function

2 Background

In this section, a brief background about testing and maintenance of expert systems is provided.

2.1 Expert systems testing

Current tools and techniques used for testing expert systems include test case generation, face validation, Turing test, field tests, subsystem validation, and sensitivity analysis. Since there is no one single test technique that captures all errors, developers must try a combination of different methods [1]. Of these techniques, test case generation continues to be the most popular technique. Approaches and tools to automating the testing process include Expert systems Validation Associate (EVA) [4], dependency charts, decision tables, graphs or Petri nets, and exploration of dynamic and temporal relationship between rules [16]. Recently, a complete methodology for the validation of rule-based expert systems is presented by [17]. This methodology has two central themes. The first one creates a minimal set of test inputs that adequately cover the domain represented in the knowledge base (KB). Second, applying Turing Test to evaluate the system's responses to test inputs and compares them to the responses of human experts.

2.1 Expert systems maintenance

Approaches of knowledge base (KB) maintenance are principally of two types [7]. First, approaches that take the knowledge base as presented and then try to control the maintenance process [15]. Second, approaches that engineer a model of the knowledge base so that it is in a form that is inherently easy to maintain [5,14].

In their research, Katsuno, H. and Mendelzon, A.O. distinguished between two fundamental modifications of the knowledge base update and revision. They provided a model-theoretic characterization of updates in terms of ordering among interpretations. A set of modification operators are defined which suggest that to update as contraction is to revision. [5] Established his maintenance approach based on the conceptual model of a knowledge base that specifies what should be in an implementation of that KB. Maintenance links are constructed by joining two items in the conceptual model if modification of one of these items could require that the other item should be checked for correctness and the validity of the conceptual model is to be preserved. The number of maintenance links can be very large. Thus, the efficiency of this maintenance procedure depended on a method for reducing the density of the maintenance links in the conceptual model.

3. Testing process Initiatives

To ensure the quality of any expert systems, a test model that serves as a guide for ESs developer is proposed. This section highlights the main initiatives upon which the test model is developed.

3.1 Structure of Testing Model

The proposed testing model decomposed of two major complementary components: testing and maintenance as shown in figure1. Each plays an important role in quality assessment of ES. The characteristics of each phase are described in terms of testing capability and goals. Testing aims to ensure that the expert systems provide the correct results in the correct form when it is called upon to solve any problem.

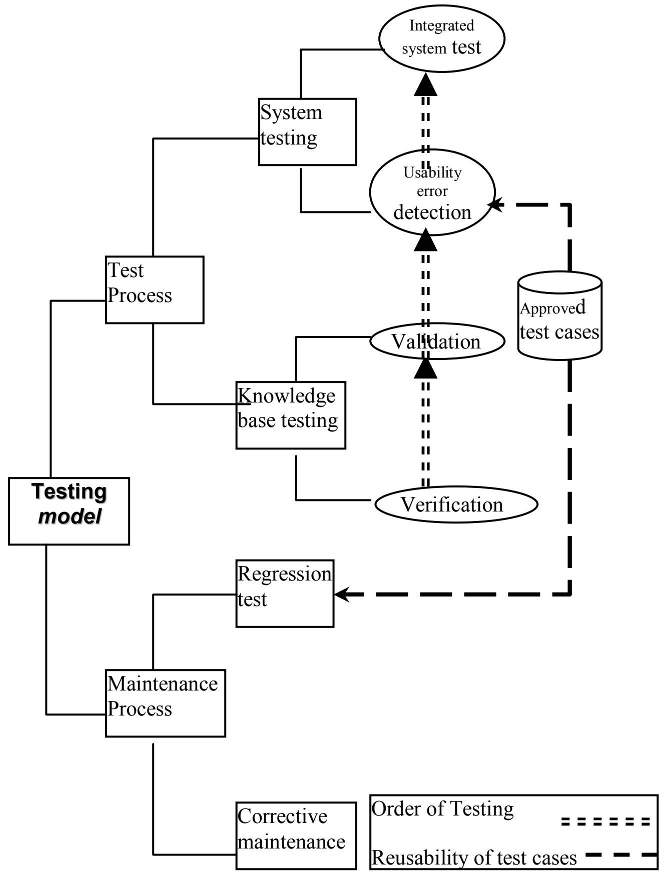

Fig1. Structure of the test model

To achieve this goal, testing process is divided into two main phases:

- Knowledge base testing and,
- System testing.

Maintenance plays an important role in keeping expert systems with the same quality level even after the development process. The maintenance phase is divided into two main parts:

- Corrective maintenance and,
- Regression testing.

Applying corrective maintenance is essential when structural or functional error is revealed in the knowledge during its testing (or either as a feed back from system users). The dynamic testing strategy of regression is used to affect after the modification of any KB components.

3.2 Testing Model Features

The overall expert systems testing model is composed into successive phases. Ordering of these phases implies the following distinguished features:

Hierarchal testing

The order of applying the testing phases indicated by fig1, conveys the following advantages:

⇒ Since verification is applied before validation takes place, syntax and semantic errors are eliminated as early as possible and avoid expensive correction later.

⇒ KB testing is applied before system testing. This provides an early help in correcting and improving internal KB components prior to the completion of ES product.

Incremental testing

The proposed test activities are provided in bottom-up fashion to different KB components which allows reusing of test cases in successive phases like follow:

⇒ The applied test cases generation techniques used in the validation activity depend mainly on reusing test cases of domain knowledge in testing inference knowledge and the test cases of inference knowledge in testing tasks.

⇒ Test cases used during the system test are based on agreed test cases result from the validation process.

⇒ Selective regression test cases are a subset of the test case pool used to store approved ES test cases.

4 Operative ES Testing Process

To implement a well organized ES test process, it is essential to distinct between test activities according to the goals of these activities. Testing cycle is performed through two phases: KB testing and, system testing.

4.1 PhaseI: Knowledge Base Testing

The aim of KB testing is to find out whether the developed system is free from internal errors such as consistency and completeness as well as ensure that the knowledge meets the required specification. KB testing is composed of two primary testing activities*: static testing* (is commonly known as **verification**) and *dynamic testing* (**validation**).

4.1.1 Verification

The verification process of the KB ensures that the system is free of induced errors by the developer during the design and implementation stages. Verification does not involve the execution of KB, it examines internal correctness and consistency of the KB. An automatic verification tool is used to detect consistency, completeness and other error of the KB [10]. This automatic tool has been enhanced to cover different KB representation paradigms such as table and mathematical function. The automatic verification process is composed of two main parts:

- Domain layer verification

Each KB component is checked for consistency and completeness. Consistency concerns detecting the following errors: redundancy, subsumption, circularity, and conflict. While completeness involves ensuring that all the knowledge is referenced, and there is no attempt to access non-existing knowledge. It concerns the following errors: unrefrenced concept, attribute, and attribute values, dead-end IF part (unfirable rule, table, and function), and unreachable conclusion (unused consequence) .

- KADS-layer matching

This verification activity aims at elimination of inconsistency errors appears due to interactions between different KB layers. This happens when any of the three layers that construct the KB refers to undefined parts of another layer[11].

4.1.2 Validation

Validation concerned with the execution of the KB on a set of well-defined test cases to evaluate the functional, structural or computational aspects of the system. Then compare between the system behavior against the specification of intended behavior expressed as test cases. The validation process composed of the following related steps.

- *Test case generation*
- *Test case experimentation & assessment*

4.1.2.1Test case generation

The aim of this step is to generate and optimize a set of test input combinations (test data) that will simulate the inputs to be applied to the expert systems in actual operation. Test cases are automatically generated in form of input concept-attribute pairs and their suggested values. Different techniques have been implemented to automatically generate test cases for each knowledge base components [9]. Two main competing objectives were considered during design and implementation of these techniques:

1) *coverage*

Provide complete coverage such that each KB components is fired at least once.

2) *efficiency*
Minimize the number of test cases to make the testing process practical. Constraints and heuristics were used to prune the undesirable cases from the original set of test cases.

4.1.2.1 Test case experimentation & assessment

Since intelligent systems emulate human expertise, it is clear that human opinion needs to be considered when evaluating the correctness of the system's response. The domain experts who were involved during the knowledge acquisitions phase analyze the outcome of each generated set of test cases to approve its validity. Approved test cases are stored in a test cases pool (composite data base) to be used in further testing phases.

4.2Phase II: System testing

The aim of this phase is to ensure that the ES product is running properly. Different tests that serve to examine the performance of the system in different situations are applied. This phase is decomposed into two important activities:

- Usability error detection
- Integrated system test

4.2.1 Usability error detection

The usability of any software product is a key factor when assessing the quality of interactive software [13]. Therefore, this step aims to assess the expert systems user friendliness as well as identify system robustness. User friendliness is sustained by navigation through the expert systems interface and find out the following:

1- Is it obvious to the user which actions are available to him

2- Is the ES consistent from page to page, including font sizes and colors, adjust of displays, meaningful titles, meaningful buttons in name and actions, use the same language

3- If a user forgets to fill in a required field, is there a friendly error message and a change of the color of the field label to some other conspicuous color

System robustness considers the correct output of the system w.r.t. incorrect input values and possibly biased knowledge in the system. Thus, robustness testing investigates the correct/expected behavior of the knowledge system even if some input values were falsely entered (e.g. due to typos of the user)[2]. Based on Torture tests[8], a series of empirical tests that gradually worsening the quality of the used knowledge is applied to the ES. By reusing the approved validation test cases, we modify its values so that they achieve the purpose of this test. These values (shown in table1) are

Concept-attribute type	Data value	Type of test
Numeric	Field boundary	Extreme data test
	Invalid data (e.g -ve values)	Invalid data test
	Invalid type (nominal values)	
Nominal	Invalid data (choose the most exceptional value e.g. unknown).	Extreme data test
	Invalid type (numeric values)	Invalid data test
	For multi-value fields, choose most of the possible values	Extreme data test
Date	Date boundary (very near and very far dates)	Extreme data test

Table1: Suggested testing values for system robustness

selected to measure the reliability of the expert system when being used in real environment. Basic data flows, error treatment and invalid data verification method are the basic for selection [18]. Robustness test case generation algorithm is shown in figure2.

```
TTCs:= List that contains test cases of task;
Ont:= KB ontology; RTCs:empty list;
Num := Number of test cases of TTCs
Begin
  For i:=0 to I:= Num do
  Begin
    Len= Number of records of TTCs[i]
    For j:=1 to j:= Len do
      Begin
      T:= concept-attribute type of TTCs [i,j]];
      V:= generate value for TTCs [i,j] according to table1;
      Store (T,V) in RTCs [I,j]
      j:=j+1;
      end
  i:=i+1;
  end
end
```

Fig2: Algorithm used to generate usability test cases

4.2.1 Integrated system test

During this test, the following attributes are evaluated

1. System installation
Verify that the installation procedures for full, partial, or upgrades works as documented. Also test the uninstall process.
2. Start Up & Quit
Run the expert systems and find out if it works when exit and rerun it again for several times.
3. Load Testing
Expose the system to a typical user experiences over a short period of time. The goal is to model real world interaction
4. Web-testing

Testing web-based expert systems should focus on the site's intended behaviour by assessing the following issues: navigation, page content, and load test.

5. The ES Maintenance phase

The maintenance of KB becomes an important issue to successful and efficient expert systems [19]. The maintenance problem is to determine which parts of the KB should be checked for correctness in response any change. The suggested maintenance phase decomposed into: corrective maintenance and regression testing. Corrective maintenance is used in order to correct erroneous system behaviour like addition, deletion or adaptation of incorrect knowledge. Regression testing attempts to validate modified system and ensure that no new errors are introduced into previously tested code.

5.1 Corrective maintenance

Modifying expert systems often requires several individual changes to various components, and all those changes must be carefully coordinated. Accordingly, corrective ES maintenance task is decomposed into:

- KB dependency analysis
- KB modification

These steps are described in the following subsections.

5.1.1 Knowledge base dependency analysis

The first step of the maintenance process is to identify the area in the KB that would be affected by any maintenance request. A set of indices is created that explicitly justify the interdependencies between different KB components. These indices are:

Domain index

The domain index controls the usage of any concept/property/value by different domain knowledge components such as rules, tables, and mathematical functions. This index comprises the location of every concept/property pairs defined in the domain ontology.

Inference index

The inference index controls the usage of any domain knowledge in the inference layer.

Task index

Actually there are two task indexes. The first one controls the usage of inference knowledge in the task layer. The second index controls the usage of different concept/property in the transfer-tasks.

5.1.2 KB modification

Maintenance of the KB including following operation: add, delete, and modify of different domain KB components such as (concepts, rules, tables and mathematical functions). Each of these operations not only affects the required component but also affect other related components. Therefore, this step implies understanding of the effect of different maintenance request on the whole KB, then apply the required changes

5.2 Regression testing

The dynamic testing strategy of regression is used to affect after the modification of any KB components. In the use of traditional regression, all previous test cases are reapplied. This leads to an expensive overhead. We introduce a fully automated selective regression testing process that only repeats test cases of modified KB component. Our regression testing implies the following:

1. Develop and maintain a composite test case library.
2. Apply selective regression testing.

5.2.1 Develop and maintain composite test case library

As test cases are generated and approved by the domain experts during the validation step, they are stored in a composite test case library. The architecture used in building the test cases library is similar to that of building the KB (i.e. each KB component should have its corresponding test case library).

5.2.2Apply selective regression test

Based on the reduction techniques (RED) for regression test [12], we propose a selective regression test case technique that aims at reducing the number of selected test. Our approach focuses on KB modification and its impacts. Affected KB components result from any KB modification is determined. Consequently, the test cases of the modified KB components and other affected KB components are only reused. Following are the steps used in the regression test process:

1. Identify parts of KB that will be changed according to the maintenance request.
2. Select $T_1 \cap T$ to execute on .
3. Test KB with T_1
4. Identify tests in T_1 that fail for KB, identify the faults that caused the failures, and restart the process.

The proposed regression testing protocol reduce the cost of regression testing by reusing a subset of approved test cases with minimal additional work.

6. Conclusion

This research presented the required model for testing expert systems at the central lab of agriculture expert systems (CLAES). The proposed testing model lends itself to an early assessment and improvement of the end-user product. The model consists of two main complementary processes: testing and maintenance. Testing decomposes two successive processes: KB testing and system testing. During the KB testing, the expert system is checked for its consistency, completeness and correctness. System testing aims to ensuring the reliability of the expert systems before distribution. The maintenance process aims to keep the KB with the same quality level even after applying the testing processes. Corrective maintenance is used in order to correct erroneous system behavior. Selective regression test aims to reduce the cost of retesting by selecting test cases from the initial suite of the approved test cases. The testing methodology is enriched by a supporting tool set for each of the recommended testing criteria. Evaluating expert systems performance is an essential to provide complete testing environment Therefore, we currently develop a methodology for evaluating expert systems and build a tool for automating the evaluation process.

References:

[1]Bellman, K. L., Landauer,C. (1995). Designing testable, heterogeneous software environments. J. Sys. Softw. 29, 3,199-217.

[2]Baumeister, J.; Bregenzer, J.; Puppe, F. “A Methodological View on Robustness Testing of Rule-Based Knowledge Systems Proceedings of the 13th LIT (Leipziger Informatik Tage), LNI 19, 131-138, 2005

[3]Coenen F., Bench-capon T. (1993). Maintenance of knowledge based systems, Academic Press,London.

[4]Chang C., Combs, J. B., Stachowitz, R. A. (1990) A report on the expert systemss validation associate (EVA). Exp. Sys. With Appl.1, 3, 217-230.

[5]Darwiche, A. (1999). “Compiling Knowledge into Decomposable Negation Normal Form” in proceedings International Joint Conference on Artificial Intelligence, IJCAI’99, Stockholm, Sweden, August 1999, pp 284-289.

[6]Debenham, J.K. (2003) "A Rigorous Approach to Knowledge Base Maintenance", in proceedings The Sixteenth International Conference on Industrial & Engineering Application of Artificial Intelligence & Expert systemss, IEA/AIE 2003 Loughborough, UK, pp219-228.

[7]Debenham, J.K. and T. Menzies (2000) "Expert systems Maintenance" in Encyclopedia of Computer Science and Technology, A. Kent and J.G. Williams (Eds), Vol. 42, pp35-53, (Marcel Dekker, Inc).

[8]Groot et al., P, Frank van Harmelen, and Annette ten Teije. Torture Tests: A Quantitative Analysis for the Robustness of Knowledge-Based Systems. In *Knowledge Acquisition Modeling and Management*, LNAI 1319, pages 403–418, Berlin, 2000. Springer Verlag.

[9]El-Korany, A., Rafea, A., Baraka, H., Eid, S. (2000). A Structured Testing Methodology for Knowledge-Based Systems, *In proceeding of DEXA international conference*, London.

[10]El-Korany A., Shaalan K., Baraka H., & Rafea A.(1998). An Approach for Automating the verification of KADS-based Expert systemss, *New Review of Applied Expert systemss*, Vol. 4, pp. 107-124, Taylor Graham, UK.

[11]El_Korany, A. (1996). *Knowledge base Verification*, MS.C thesis.

[12]Harrold, M, J.(1999). Testing evolving software, The Journal of Systems and Software 47, pp173.

[13][ISO99] International Standard (1999). ISO 13407. Human-centred design processes for interactive systems.

[14]Jantke, K.P. and Herrmann, J. (1999). “Lattices of Knowledge in Intelligent Systems Validation” in proceedings 12th International FLAIRS Conference, Florida, May, pp499-505.

[15]Katsuno, H. and Mendelzon, A.O. (1991). “On the Difference between Updating a knowledge Base and Revising It”, in proceedings Second International Conference on Principles of Knowledge Representation and Reasoning, KR’91.

[16]Kirani, S.H., Zualkernan, I.A., Tsai, W.T. (1994). Evaluation of expert systems testing methods. Commun. ACM 37, 11,pp 71-81.

[17]Knauf,R., Avelino J. Gonzalez, Thomas Abel (2002). A framework for validation of rule-based systems. IEEE Transactions on Systems, Man, and Cybernetics, Part B 32(3): 281-295.

[18]Myers, Glenford J..(2004) . The Art of Software Testing. John Wiley & Sons, 2nd .

[19]Qian,Y., Zheng,M., Xiuxi Li, Li Lin, 2005, ”Implementation of knowledge maintenance modules in an expert systems for fault diagnosis of chemical process operation”, Expert systems with Applications.

[20]Schreiber,G., H. Akkermans, A. Anjewierden, R. De Hoog, N. Shadbolt, W. Van De Velde & B. Wielinga, (2000). Knowledge Engineering and Management: The CommonKADS Methodology. MIT Press. ISBN 0-262-19300-0