Exploring and Validating the Contributions of Real-World Knowledge to the Diagnostic Performance of Automated Database Design Tools

Shahrul Azman Noah
Faculty of Information Science & Technology, National University of Malaysia,43600 UKM, Bangi, Selangor, Malaysia.
samn@sun1.ftsm.ukm.my

Michael Williams
Department of Computing & Mathematics, Manchester Metropolitan University,John Dalton Building, Chester Street,Manchester M1 5GD, UK.
m.williams@doc.mmu.ac.uk

Abstract

Automated database design tools employ knowledge-based systems technology in order to provide intelligent support to humans during the process of database analysis and design. However, the level to which these tools can simulate the diagnostic capabilities of human designers when performing a design task remains in question. Human designers employ what might be called "knowledge of the real world" in carrying their design activities; such knowledge is employed by only a few automated database design tools. Therefore, in recent years there have been a number of attempts to develop tools that are capable of exploiting such real-world knowledge. It has been claimed that the use of such knowledge has the potential to increase the diagnostic performance of automated database design tools. However, to date, little if any formal exploration and validation of this claim has taken place. This paper presents our activities in exploring and validating the implications for exploiting three approaches facilitating the use and exploitation of real-world knowledge in the diagnostic performance of database design tools. Results obtained have demonstrated that the improvement of certain aspects of diagnostic performance have been achieved. However, the extent to which these aspects have been attained and subsequently statistically validated varies.

1. Introduction

Automated database design tools that employ knowledge-based (KB) systems technology, have the capability of providing intelligent support to humans during the process of database analysis and design. The main function of these tools, however, has been to assist users in producing a consistent design model by detecting any inconsistencies and redundancies that may exists and suggesting corrections in design.

Although a number of automated database design tools have emerged from research activities, there is a dawning realization that, despite their practical applications of KB techniques to database design, such tools may be lagging well behind human designers particularly in its diagnostic capabilities. Human designers, even when working in an unfamiliar domain, are able to make use of their knowledge of the real world in order to interact with users, make helpful suggestions, and identify errors and inconsistencies. For instance, a human designer is capable of diagnosing inconsistencies occurring as a results of using different terms or words for the same concept such as "*client*" and "*customer*", or different forms of the same verb such as "*teach*" and "*taught*", regardless of the application domain. The majority of existing automated database design tools do not possess such real-world knowledge, and are therefore required to engage users in a series of trivial questioning dialogues in order to solve simple inconsistencies, and in certain circumstances, such inconsistencies remains undiagnosed. This situation has resulted in numerous calls for the representation of real-world knowledge within such tools, coupled with the ability to reason with and make use of this knowledge.

A number of approaches to representing and exploiting real-world knowledge have been proposed, including the *thesaurus* approach [1, 2] the *dictionary* approach [3] and the *knowledge reconciliation* approach [4, 5]. It has been claimed that the use and exploitation of these approaches are capable of improving the diagnostic performance of automated database design tools [6]. However to date, little if any formal validation of these claims has taken place. This paper presents our work on exploring and validating the implications of exploiting the aforementioned approaches to the aspects of diagnostic performance of automated database design tools, the intention being to initiate the gathering of evidence to support or otherwise the claim previously stated.

2. Brief overview of existing approaches to representing real-world knowledge

The following provides a brief overview of the methods of knowledge representation employed by the dictionary, thesaurus and knowledge reconciliation approaches. Those interested in detailed discussions of each approach, along with the claimed benefits associated with their use, are referred to the relevant source literature [1-5, 7].

The dictionary approach to representing and exploiting real-world knowledge by an automated database design tool is illustrated by the Intelligent Interview System (I^2S) of Kawaguchi et al. [3]. The I^2S approach organizes the encapsulated real-world knowledge into a series of domain specific dictionaries, each dictionary comprising of verb information with both past and present forms of each verb being presented. During a design session, I^2S attempts to make use of the dictionary in an attempt to "interview", extracting a series of simple queries that the eventual database will be expected to satisfy. These queries are analyzed by the system and used in the generation of a logical structure representing the target database.

The thesaurus approach is illustrated by the Object Design Assistant (ODA) system of Lloyd-Williams [2, 8]. Knowledge is represented within the thesaurus structure by making use of series of concepts and associated synonyms, linked together via abstraction mechanisms, categorized according to those recognized by the system (aggregation, association and generalization). During a design session, the ODA system attempts to make use of the thesaurus in order to obviate the need to ask what may be viewed as being trivial questions of the user.

The knowledge reconciliation approach is illustrated by the Common Sense Business Reasoner (CSBR) system of Storey et al. [4-6]. The approach is similar to the thesaurus in that it organizes domain knowledge into a collection of domain specific concepts and relationships between these concepts. Synonyms, however, are not represented by this approach. During a design session the system attempts to reason with this knowledge in order to reconcile it with the user-specified application domain, the intention being to provide the user with meaningful suggestions of concepts and relationships missing from the evolving database design, and to augment the contents of the domain knowledge after the completion of each design session.

3. Exploring the contribution of real-world knowledge

This section presents the approach used by an automated database design tool in employing the thesaurus, dictionary and knowledge reconciliation approaches to representing real world knowledge. The automated database design tool under consideration is the Intelligent Object Analyzer (IOA), which provides support for the conceptual design of databases. As IOA was used as a research vehicle in this study, it has the capability of operating in four different modes; either using real-world knowledge provided by the thesaurus, dictionary or knowledge reconciliation approaches (this paper refers them as the thesaurus mode, dictionary mode and knowledge reconciliation mode respectively) or without the use of real-world knowledge (basic mode).

When processing in the basic mode, IOA follows a two-step procedure.

- The first step involves creating an initial representation of the application domain (known as the problem domain model) and the subsequent refinement of this model.
- The second step involves the refinement of the problem domain model by detecting and resolving any inconsistencies that may exist, and the transformation of the model into object-oriented form.

It is not the purpose of this paper to discuss IOA in depth, however, a brief outline of the diagnostic method and process employed by the tool is required in order to illustrate how the real-world knowledge may be represented and exploited during design processing.

There are four types of inconsistencies (errors) that should be detected and resolved by the IOA. These are as follows.

1. *Semantic inconsistencies*. Inconsistencies occurring as a result of missing links (i.e. no associated relationships for a particular concepts) or transitivity that may exists within the generalization or aggregation hierarchies.
2. *Inconsistent concepts*. Inconsistencies occurring as a result of missing properties (i.e. no associated properties for a particular concept).
3. *Redundant inherited properties and relationships*. Redundancy occurring within a generalization hierarchy where a generic class (superclass) and its corresponding specific class(es) (subclass) contain the same properties or participate with the same relationships.
4. *Redundant elements*. Inconsistencies occurring as a result of synonyms such as synonymous concepts and relationships which usually lead in turn to redundancies.

The detection and removal of the first three types of inconsistencies is a straightforward process which requires the IOA to inspect each node of the problem domain model and the corresponding links associated with it. Detection of any inconsistency will then brought to the attention of the user for confirmation of its removal. Based upon our testing of the tool during the development process, these types of inconsistencies can be resolved consistently by the tool without the requirement of knowledge of the real world.

However, detection of the fourth type of inconsistency (redundant elements), can be regarded as a complex process. The approach taken by IOA in detecting and resolving this type of inconsistency is based upon the object type and mismatch rules involving the comparison of pairs of association structures of the form "*A association-1 B*" *and* "*X association-2 Y*" within the problem domain model. For instance, if the verb-phrase "*association-1*" is identical to the verb-phrase "*association-2*", and *A* is identical to *X* then the possibility exists that concepts *X* and *Y* are synonymous. The detection of this type of inconsistency, however, is only performed if *A* and *B* and *association-1* and *association-2* are *exactly* the same.

Figure 1 illustrates an example of such an inconsistency capable of being detected by IOA when processing in the basic mode.

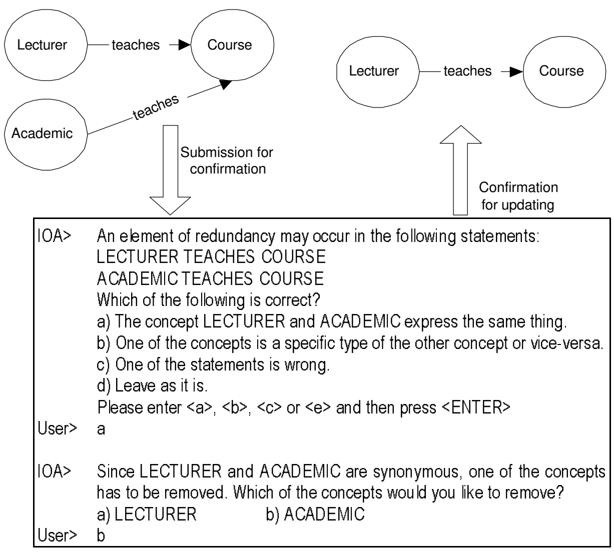

Figure 1. Detection and removal of synonymous concepts

As can be seen in Figure 1, the removal of the synonymous concepts between "*lecturer*" and "*academic*" is achieved from a series of interaction between the tool and the user.

The object type and mismatch rules are also used to detect and resolve redundant relationships. Such circumstances occur if the concepts *A* and *X* and *B* and *Y* are *exactly* the same. An example of such a situation is as illustrated in Figure 2.

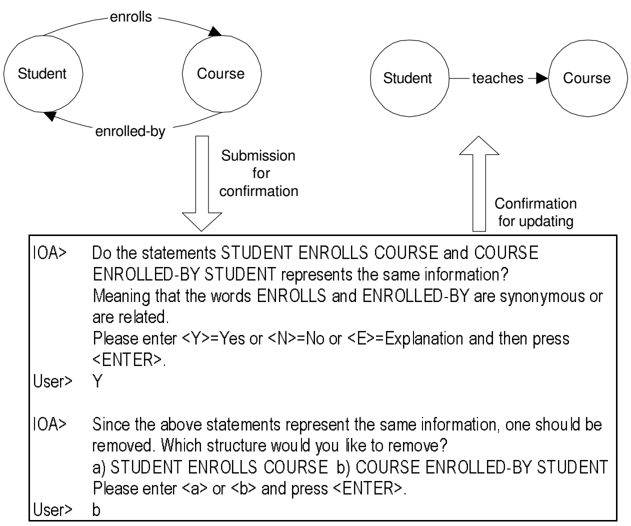

Figure 2. Detection and removal of redundant relationships

Similar to the previous case, the removal of the redundant relationship is achieved from a series of dialogue.

As can be seen, in the basic mode of processing, the IOA conducts a dialogue with the user in order to confirm its understanding of a domain or to obtain additional information. When making use of real-world knowledge provided by the thesaurus and dictionary approaches, the tool refers to this knowledge wherever possible, only resorting to questioning the user if the real-world knowledge cannot provide the required information. This procedure is also followed by when making use of real-world knowledge provided by the knowledge reconciliation approach. However, in addition to this procedure, IOA also attempts to conduct a process of *reconciliation of knowledge*, where the description of the application domain submitted by the user is compared and matched with the system-held real-world knowledge in order for IOA to construct an understanding of the contents of the application domain. The understanding gained by the tool from this activity, is used and exploited throughout the design process wherever possible.

The remainder of this section describes an exploration of the capabilities of the dictionary, thesaurus and knowledge reconciliation approaches to improving the performance of the IOA tool in detecting and resolving the fourth type of inconsistency previously described. This exploration is mainly concerned with the ability of each approach to enhance the tool's diagnostic capability and the possible improvement of the diagnostic processing.

3.1. The dictionary approach

Based upon the object type and mismatch rules previously discussed, if such structures as "*academic teaches course*" and "*course taught-by lecturer*" are presented to IOA, the tool cannot detect the possible synonymous concepts between "*academic*" and "*lecturer*", since it is unable to detect the similarity implied by "*teaches*" and "*taught-by*". The use of verb information provided by the dictionaries however allows the IOA to identify such similarities, and to subsequently request confirmation from the user of the possible synonymous concepts between "*academic*" and "*lecturer*". Therefore, the use of the dictionaries facilitates the detection of inconsistencies within the problem domain model that might otherwise remain undetected.

The capability of identifying potential similarities between various forms of verbs also results in the simplification of some of the interactions between IOA and the user during the diagnostic activities. For instance, if the relationships "*student enrolls course*" and "*course enrolled-by student*" are detected by IOA, the requirement on the part of the user to confirm the similarities between "*enrolls*" and "*enrolled-by*" as illustrated previously in Figure 2 is removed.

Based upon this exploration, it is expected that exploiting the dictionary approach could increase the number of errors capable of being detected by IOA; and reduce some of the trivial user/tool interactions (such as those occurring during an unnecessary questioning process) which therefore increases the IOA processing efficiency in terms of both the number of user/tool interactions and the processing time required.

3.2. The thesaurus approach

One of the unique features of the thesaurus approach when compared to the other approaches under examination in this study is the explicit representation of synonyms. Our investigation has revealed that exploiting such synonyms may simplify some of the diagnostic process. For instance, if the structures "*lecturer teaches course*" and "*academic teaches course*" are presented to IOA during processing in the basic mode, detection of possible synonyms - "*lecturer*" and "*academic*" - is achieved from the object-type and mismatch rules previously illustrated in Figure 1. However, when referring to the thesaurus-type structure (if the required information is available), the requirement for comparing pairs of association structures in order to detect the possible synonymous concepts is removed. Such an explicit representation of synonyms also facilitates the detection of synonymous concepts by IOA (undetected without the use of real-world knowledge), provided that such information is available within the thesaurus-type structure.

Another unique feature of the thesaurus-type structure is the use of abstraction mechanisms to link between concepts. This feature enables the IOA to group together structures that share similar semantic constructs such as "*graduate-student enrolls course*", "*postgraduate-student sign-up course*" and "*course enrolled-by postgraduate-student*". These structures exhibit redundancies and the requirement to compare each pair of the possible redundant structures (without the use of real-world knowledge) is again removed.

Similar to the dictionary approach, it is expected that exploiting the real-world knowledge provided by the thesaurus approach could increase the number of errors capable of being detected by IOA and at the same time minimize the processing required to detect such errors.

3.3. The knowledge reconciliation approach

As previously discussed, the knowledge reconciliation approach requires the real-world knowledge structure to be reconciled with the structure of the user's application domain, in order for the IOA to construct an understanding of the contents of the application domain. Results from this process are the classification of concepts either as *same* (concepts which have the same name or they are synonymous); *close* (where one concept is a specialization of the other concept or both are specialization of a common generic concept) or *different* (if concepts cannot be classified either as *same* or *close*); and the classification of the association either as *same* (associations are exactly the same or singular/plural of each other), *close* (active/passive form, related by time/tense) or *different* [7].

The initial "understanding" gained by IOA from the reconciliation process is exploited during the process of detecting and resolving inconsistencies. For instance if the structures "*program enrolled-by student*" and "*student takes course*" are presented to IOA and from the reconciliation process IOA "understand" (if available) that the terms "*enrolled-by*" and "*takes*" are synonymous, then IOA will suggests that the concepts "*program*" and "*course*" are potentially synonymous. Such an inconsistency will remain undiagnosed if such an understanding is not available.

The "understanding" also simplifies the process of detecting and resolving inconsistencies. For instance if the concepts of the problem domain model such as "*learner*" and "*student*" were classified as the same with the real world knowledge concept of "*student*", then the possibility of the concepts "*learner*" and "*student*" to be synonymous exist. As a result, the requirement to compare pairs of association structures based on the object type and mismatch rules is removed.

As described earlier, existing automated tools sometimes engage users with a series of questioning that may be viewed as being trivial. For instance, the structures "*lecturer supervises student*" and "*student advised-by lecturer*", would be detected by IOA as being potentially redundant, although in reality it would be recognized otherwise by humans. Therefore, a series of interactions similar to Figure 2 will take place. However, if an understanding that the terms "*advised-by*" and "*supervises*" are not related (different) based from the reconciliation process, the requirements for similar interactions as in Figure 2 is removed.

From this exploration, it is believed that the understanding gained from the knowledge reconciliation process could increase the number of errors capable of being detected and resolved by IOA. The approach may also capable of simplifying the diagnostic process (in terms of the number of user/tool interactions and CPU time required), but the reconciliation process associated with this approach may distort this assertion.

4. Empirical testing and validation

In order to validate the possibility of improving the diagnostic performance of automated database design tools previously explored, a series of testing and validation activity were conducted. The general approach to systems' testing and validation employed within this research work included: 1) the establishment of performance related criteria; 2) the development of suitable testing and validation methodology; and 3) the analysis of the results produced. It thus basically follows an approach recommended by Liebowitz [9] and O'Keefe and Preece [10].

4.1 The Establishment of Criteria

The establishment of criteria from which the diagnostic performance of an automated database design tool can be tested and subsequently validated was found to be central in the testing and validation activity. The performance-related criteria chosen in this study mainly concerns to the aspects of systems' processing efficiency, systems' appearance of intelligence and quality of systems' design output, which were mainly derived from the opinions of [2, 4, 6, 11, 12, 13].

The criteria therefore are as follows.

- *Processing time*. Processing time refers to a single design action (such as time required to resolve a single inconsistency). It is not influenced by human factors as it is measured from the point at which the tool commences an action until that action is complete. Processing time is, however, influenced by the complexity of the design input, the complexity of the system-held domain knowledge and reasoning associated with it; and the specification of the processor of the PC in use.
- *User/tool interactions*. User/tool interactions refers to the number of interactions required between the tool and the user, in order for the tool to confirm its understanding of some aspects of the application domain or to acquire additional information should it be required.
- *Detected and resolved errors*. This criterion refers to the number of errors that have been successfully detected and resolved by the tool.
- *Errors within designs produced*. This criterion refers to the number of errors or inconsistencies that may exist from the design output produced by the tool.

4.2. The Testing and Validation Method

The case-testing method was declared by O'Keefe and Preece [10] as the most dominant and prevalent method of validation. This method usually involves an execution of a set of benchmarks (test-cases) by a design tool where the results produced are compared with those of human experts or computational models (such as simulation models or regression models). This method has exhibited satisfactory results, and will continue to be the most prevalent approach in years to come [14]. The case-testing method was, therefore, seen suitable to be implemented in this experimentation work.

The main problem in case-testing approach, however, is to find a good and reliable set of test-cases that guaranteed to expose all aspects of errors in data modeling. As a result, intentionally synthesized test-cases which were mainly derived from a set of initial design problem by systematically altering them in sensible combinations are usually acceptable [10, 15]. This approach (similar to the sensitivity analysis) was initially proposed by O'Keefe & O'Leary [16] and implemented in Miller [17] and Gonzalez et al. [14].

As illustrated in Figure 3, the approach taken in this study, firstly requires the production of a number of synthesized errors. The errors introduced included synonymous concept(s), synonymous or related relationship(s) and a combination of both. Secondly, each of these synthesized errors and combination of them were then systematically embedded into the corresponding design problem to generate the set of test-cases.

The set of generated test-cases were then executed in all the four modes of processing. Thus, for each executed test-case, four sets of results were obtained and compared. The design problems used to generate the set of test-cases were primarily extracted from existing literature, the advantage being that the accompanying solutions could be used as a benchmark and compared with the IOA-suggested solution in order to confirm the appropriateness or otherwise of the designs produced.

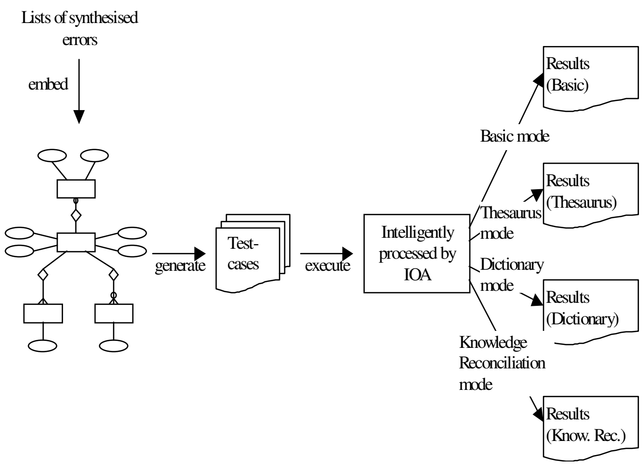

Figure 3. Approach to case-testing

4.3. Method for Analysis of Results

Since all the selected criteria under observation in this study are quantifiable, the quantitative method of analysis was employed. In this study the results observed from the execution of test-cases in the dictionary, thesaurus and knowledge reconciliation modes were subsequently compared with the results observed from the execution of test-cases in basic mode. A statistical *hypothesis test* was then conducted to assess the significant differences between these observed results at the 5% of significance level.

Although there are several recommended statistical methods available to test such hypotheses, the paired t-test method is highly appropriate in such circumstance as those prevailing in this study [10, 16]. The paired t-test method is a form of *repeated measures design*, where the same variable (observed criterion) is measured on several occasions (processing modes) for each subject (test-case).

5. Analysis of Results

The results presented here emanate from a series of tests performed on university domain problem found in the general literature [18, 19, 20, 21]. A total of 84 test-cases were generated from these initial problems. The real-world knowledge (thesaurus, dictionary and knowledge reconciliation) used was constructed as the result of a series of interviews with researchers, the results of the interviews being integrated in order to form a single representative of the domain. This was a deliberate attempt to minimize any bias that might be introduced by taking the content of the test material into account. It is not claimed that these representations portray the definitive knowledge of the university domain, but it does provide an illustration of the way in which such knowledge may be stored and exploited by an automated database design tool.

5.2. Processing Time

In this criterion, the intention was to observe as whether the use and exploitation of the thesaurus, dictionary and knowledge reconciliation approaches capable of reducing the processing time required to resolve each detected errors. The linear regression analysis results of the processing time as illustrated in Figure 4 shows that only the thesaurus approach reduces the processing time required by of approximately 26.7% as compared when no such knowledge was in use. Exploiting the dictionary and knowledge reconciliation approaches, however, resulted in an increase of the processing time by of approximately 48.7% and 63.7% respectively.

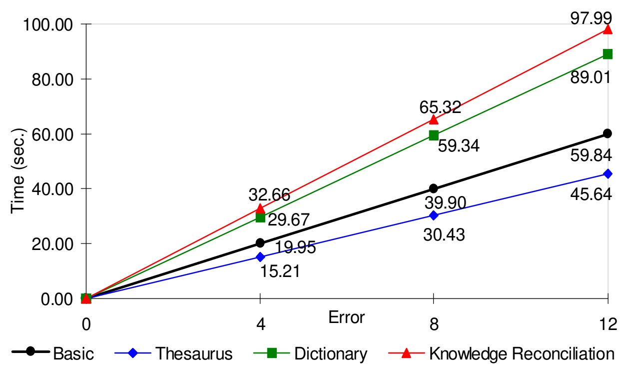

Figure 4. Processing time required for each increase in inconsistency

The previous initial findings were further supported by the paired t-test results as illustrated in Table 1.

Table 1. Paired t-test results - Processing time

Representation Approaches	t-Value	P
Thesaurus	-3.44	0.000
Dictionary	2.30	0.025
Knowledge Reconciliation	3.38	0.000

The results indicate that only the thesaurus approach significantly reduced the processing time required to detect and resolve each error, as evidenced from the significant value of P (P<0.05) and the negative t-Values. Although, the results also show that there are significant differences for the dictionary and knowledge reconciliation approaches as compared to the basic approach, the positive t-Values indicate that exploiting both approaches results in the increasing of the CPU time required.

5.3. User/Tool Interactions

Similar to the previous observed criterion (processing time), the intention was to assess as whether exploiting the thesaurus, dictionary and knowledge reconciliation approaches capable of reducing the number of user/tool interactions required to resolve each detected errors.

Figure 5 illustrates the linear regression analysis of the number of user/tool interactions required by the four approaches in detecting and resolving errors. This figure illustrates that when using the thesaurus and dictionary approaches, the number of user/tool interactions required to detect and resolve each error were reduced by of approximately 23.7% and 20.0% respectively compared as when no such knowledge was in use. The knowledge reconciliation on the other hand resulted in an increase of approximately 4.5% of the number of user/tool interactions.

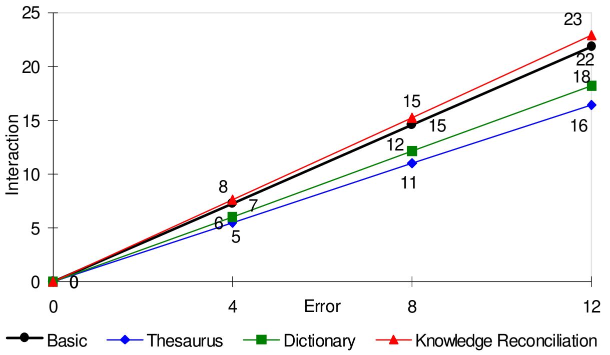

Figure 5. User/tool interactions required for each increase in inconsistency

These preliminary findings were again strongly supported by the paired t-test results as illustrated in Table 2.

Table 2. Paired t-test results - number of user/tool interactions

Representation Approaches	t-Value	P
Thesaurus	-7.99	0.000
Dictionary	-3.42	0.001
Knowledge Reconciliation	4.04	0.000

The values of P and the negative t-Values associated with the thesaurus and dictionary approaches indicate that exploiting both approaches has resulted in the significant increased in the number of user/tool interactions required to detect and resolve errors as compared when no such knowledge was in use. The knowledge reconciliation approach, however, incapable of increasing this aspect of diagnostic performance.

5.4. Number of Errors Detected and Resolved

Encouraging results for this performance-criterion have been achieved, as evidenced by the values of P and t-Value illustrated in Table 3. The table shows that all the three approaches under examination in this study significantly increased the capability of the IOA tool in detecting and resolving errors as compared with the basic approach.

Table 3. Paired t-test results - number of errors detected and resolved

Representation Approaches	t-Value	P
Thesaurus	9.74	0.000
Dictionary	4.55	0.000
Knowledge Reconciliation	4.68	0.000

5.5. Errors within Designs Produced

The criterion number of errors of design produced has a direct impact with the previous criterion of number of errors detected and resolved per test. Accordingly, as illustrated in Table 4, exploiting all the three approaches has resulted in the significant reduction of the number of errors within design produced per test as compared with the basic approach.

Table 4. Paired t-test results - number of errors within designs produced

Representation Approaches	t-Value	P
Thesaurus	-9.74	0.000
Dictionary	-4.55	0.000
Knowledge Reconciliation	-4.68	0.000

6. Discussion

Table 5 presents a summary of results for the thesaurus, dictionary and knowledge reconciliation approaches from the previously discussed validation test.

Table 5. Summary of results

Observed Criteria \ Representative Approaches	Thesaurus	Dictionary	Knowledge Reconciliation
Reduces the no. of user/tool interactions per error.	Yes	Yes	No
Reduces the CPU time required per error.	Yes	No	No
Increases the no. of errors resolved per test.	Yes	Yes	Yes
Reduces the no. of errors per design produced.	Yes	Yes	Yes

The table suggests that certain elements of diagnostic performance have been achieved by the use and exploitation of real-world knowledge represented by the approaches under examination. A discussion associated with the capability of each approach, therefore, follows.

The thesaurus approach has the capability of achieving all the performance-related criteria under observation in this study. These achievements may due to the nature of the thesaurus-type structure represented by this approach. For instance, synonyms which were explicitly represented by this approach were proven capable of simplifying the process of detecting and resolving errors in the form of synonymous concepts and redundant relationships, that subsequently resulted in the reduction in the number of user/tool interactions and CPU time required. Such explicit representations of synonyms have also resulted in an increase in the number of errors detected and resolved per test. However, as relationships within the thesaurus-type structure do not contain any verb-phrase related information, it is believed that the increased of capacity of the IOA tool in detecting errors when exploiting the thesaurus approach relies heavily on the synonyms associated to each concept.

The real-world knowledge structure of the dictionary approach on the other hand, contains information about the past and present forms of each verb being presented. Therefore, exploiting such an information has enabled the IOA to remove certain aspects of confirmation and augmentation process with the user, which subsequently results in the reduction of the number of user/tool interactions. However, the process of consulting the dictionary of verbs in order to detect the possible various forms of the same verb presented within a design model is believed to cause the consumption of the CPU time. The capability to recognize the different forms of the same verb has also facilitated IOA in detecting and resolving inconsistencies from the object-type and mismatch rules. As a result an increased in the number of inconsistencies capable of being detected and resolved has been achieved when exploiting this approach.

As for the knowledge reconciliation approach, the "understanding" gained as the result of the "reconciliation of knowledge" process, has certainly enhance the IOA capability in detecting and resolving inconsistencies. Although, such an "understanding" should also capable of minimizing the number of user/tool interactions required to detect and resolve inconsistencies, the interactions associated with the knowledge reconciliation activity seems to jeopardize this claim.

Although encouraging results have been obtained from the testing and validation work, it is recognized that consideration must be given to a number of practical issues. The effectiveness of the tool depends greatly on the accuracy and completeness of the system-held real-world knowledge, and the results obtained from the tests may be influenced to certain extent by the variety and coverage of the generated test-cases.

7. Conclusion & Future Works

This paper has presented the findings of an assessment of the dictionary, thesaurus and knowledge reconciliation approaches to representing and exploiting real-world knowledge by an automated database design tool (IOA). The aim of the investigation was to ascertain whether the use of real world knowledge has the capability of increasing the diagnostic performance of automated database design tools. The performance related criteria under observation in this study were the number of user/tool interactions; the processing time; the number of errors detected and resolved; and the number of errors within designs produced.

Among these observed criteria, results from this study have demonstrated that the exploitation of real world knowledge has strong potential for enhancing the capability of automated database design tools in detecting and resolving inconsistencies or errors which subsequently results in the improvement of the quality of the designs produced (minimizing the number of errors within the designs produced). A mixed of results however, were obtained for the other criteria under observation in this study. Similar experiments were conducted to the clinical/hospital and library domains, and consistent findings as presented in this paper were obtained.

Considering the fact that a consistent conceptual model has a strong influence to the overall successful implementation of a database system project, it might be of importance for any automated database design tools to incorporate elements of real world knowledge into their processing activities. This is because this study has revealed that system-held real world knowledge, even when represented in a simple form (such as list of verbs), is potentially able to improve the consistencies of design output produced.

Although, there are a number of methodologies proposed for the testing and validation of expert systems [15, 16], there are only a handful of papers that describe actual experiences related to the testing and performance evaluation of operational systems. This paper has presented an approach and methodology for performing such a task. The methodology employed involved: the generation of a series of test-cases, the processing of the test-cases, and the use of statistical analysis to validate the observed results.

Future work includes investigating the possibility of combining the main characteristics represented by the tested approaches to representing real-world knowledge in this study, in order to establish a more reliable and effective real world knowledge model.

References

[1] M. Lloyd-Williams, “Knowledge-based CASE tools: improving performance using domain specific knowledge”, *Software Engineering Journal*, 9(4), 1994, pp. 167-173.

[2] Lloyd-Williams M., “Exploiting domain knowledge during the automated design of object-oriented databases”, in: Embley, D W and Goldstein, R C, eds., *Proceedings of the 16th International Conference on Conceptual Modeling*, Spinger-Verlag, Berlin, 1997, pp. 16-29.

[3] Kawaguchi, A., Taoka, N., Mizoguchi, R., Yamaguchi, T. and Kokusho, O., “An intelligent interview system for conceptual design of database”, in: *ECAI '86: The 7th European Conference on Artificial Intelligence*, Conference Services Ltd., London, 1986, pp. 1-7.

[4] Storey, V.C., Goldstein, R.C., Chiang, R.H.L. and Dey, D., “A common-sense reasoning facility based on the entity-relationship model”, in: Elmasri, R A, Kouramajian, V and Thalheim, B, eds., *Proceedings of the 12th International Conference on the Entity Relationship Approach*, Springer-Verlag, Berlin, 1993, pp. 218-229.

[5] Storey, V C, Chiang, R H L, Dey, D, Goldstein, R C, Sundararajan, A and Sundaresan, S, “Knowledge reconciliation for common sense reasoning”, in: De, P and Woo, C, eds., *Proceeding of the 4th Annual Workshop on Information Technologies and Systems*, 1994, pp. 87-96.

[6] V. C. Storey, “Real world knowledge for databases”, *Journal of Database Administration*, 3(1), 1992, pp. 1-19

[7] V.C. Storey, R.H.L. Chiang, D. Dey, R.C. Goldstein, A. Sundararajan and S. Sundaresan, “Database design with common sense reasoning and learning”, *ACM Transactions on Database System*, 22(4), 1997, pp. 471-512.

[8] M. Lloyd-Williams, “Expert system support for object-oriented database design”, *International Journal of Applied Expert Systems,* 1(3), 1993, pp. 197-212.

[9] J. Liebowitz, “Useful approach for evaluating expert systems”, *Expert Systems*, 3(2), 1986, pp. 86-96.

[10] R.M. O'Keefe and A.D. Preece, “The development, validation and implementation of knowledge-based systems”, *European Journal of Operational Research,* 92(3), 1996, pp. 458-473.

[11] C.Y. Suen, P.D. Grogono, R. Shinghal, and F. Coallier, “Verifying, validating, and measuring the performance of expert systems”, *Expert Systems With Applications,* 1, 1990, pp. 93-102

[12] Moody, D.L., “Metrics for evaluating the quality of entity relationship model”, in: *Proceedings of the 17th International Conference on Conceptual Modelling, Singapore,* Springer-Verlag, 1998, pp. 211-225.

[13] S. Kesh, “Evaluating the quality of entity relationship models”, *Information and Software Technology*, 37(12), 1995, pp. 681-689

[14] A.J. Gonzalez, U.G. Gupta and R.B. Chianese, “Performance evaluation of a large diagnostic expert system using a heuristic test case generator”, *Engineering Application of Artificial Intelligence,* 9(3), 1996, pp. 275-284.

[15] R.M. O’Keefe, O. Balci, and E.P. Smith, “Validating expert system performance”, *IEEE Expert*, Winter, 1987, pp. 81-89.

[16] R.M. O'Keefe and D.E. O'Leary, “Expert system verification and validation: a survey and tutorial”, *Artificial Intelligence Review,* 7(1), 1993, pp. 3-42

[17] L.A. Miller, “Dynamic testing of knowledge-bases using the heuristic testing approach”, *Expert Systems with Applications,* 1(3), 1990, pp. 249-269.

[18] Rob, P. and Rob, C.C., *Database Systems: Design, Implementation and Management*. Belmont, Wadsworth Publishing, California, 1993.

[19] Batini, C., Ceri, S. and Navathe, S., *Conceptual Database Design: An Entity Relationship Approach*, Benjamin-Cummings, California, 1992.

[20] Bowers, D.S., *From Data to Database* Chapman Hall, London, 1993.

[21] Elmasri, R. and Navathe, S.B., *Fundamentals of Database Systems*. Benjamin Cummings California, 1989.