INTEGRATING ARTIFICIAL INTELLIGENCE TECHNIQUE AND LEXICON OF VERBS IN AUTOMATED DATABASE DESIGN DIAGNOSING

Shahrul Azman Mohd Noah
Faculty of Science and Information Technology
Universiti Kebangsaan Malaysia
43600 UKM Bangi, Selangor
Malaysia
Tel.: 603-8292482
Fax: 603-8255732
email: samn@ftsm.ukm.my

Michael D. Williams
European Business Management School
University of Wales Swansea
Singleton Park, Swansea SA2 8PP
UK
Tel.: 01443-482243
Fax: 01443-482715
email: m.d.williams@swansea.ac.uk

ABSTRACT

Conceptual database design is seen as the most important stage of a database design process, as the conceptual model produced at this stage is the first design model constructed with formal and detailed semantics. The stage, however, is also viewed as a difficult task for designers, and the potential for committing and correcting errors is significant. As a result, it is not surprising to see the emergence of a number of automated systems employing artificial intelligence (AI) techniques in providing assistance in the design of such a model. This paper presents one such tool, called the Intelligent Object Analyser (IOA). Although the tool exhibited the capacity of performing two aspects of design (design synthesis and design diagnosis), its major contribution is to assist users within the scope of design diagnosis from its capability to detect errors, and suggest corrections in design. Apart from incorporating the AI techniques, the tool also integrates a domain specific lexicon of verbs, which is meant to enhance the tool's diagnostic capabilities. The tool demonstrates the practical integration of AI techniques and domain specific lexicon to the aspects of database design diagnosing. The paper concludes with a discussion of areas for future development in this field.

Keywords: ***Conceptual Database Design, Knowledge-Based System, Design Diagnosis, Object Modelling Technique***

1.0 INTRODUCTION

Database design is a complicated and tedious process that begins with a very informal and poorly defined requirements. The concern is to represent some aspects of real-world situation by means of a database system [1]. Database design process usually proceeds into four distinguishable stages, namely: requirements specification and analysis, conceptual design, logical design and physical design. Conceptual database design, however, is seen as the most important stage of a database design process [2]. This is due to the fact that a conceptual model produced at this stage is the first design model constructed with formal and detailed semantics. Therefore, the quality of the conceptual model has a significant influence in the later stages of database design since it provides the initial reference for the complete implementation of a database system. The conceptual database design stage is also viewed as a difficult task for designers [3, 4], and the potential for committing and correcting errors is significant.

Due to its importance and yet error-prone, this paper therefore presents the development of an automated system to support such activities of database design. The system under consideration is the Intelligent Object Analyser (IOA). The unique feature of this tool is that it employs technique from the field of artificial intelligence (AI) and integrates with a library of lexicon containing a series of verb phrases.

Although contemporary database design systems have excellent facilities for editing and maintaining database development project, these systems are characterised as being passive and incapable of supporting the basic characteristics of database design process [5, 6, 7]. Database design is a knowledge intensive activity that begins with a vague requirement of the problem domain of what is to be done and results in a highly detailed formal object [8].

This paper, however, has revealed that techniques from the field of AI in the form of knowledge-based systems can be exploited to overcome the limitations of current database design systems particularly in the process of database design diagnosing.

2.0 THE INTELLIGENT OBJECT ANALYSER

IOA was developed to provide support during the conceptual stage of database design process. In constructing a conceptual design model, IOA employs the Object Modelling Technique (OMT) methodology advocated by Rumbaugh et al. [9]. Therefore, the outputs produced by the IOA exhibit the key object-oriented characteristics of object classes and instances, and the three most popular forms of data abstraction; association, aggregation and generalisation (including multiple inheritance).

As illustrated in Fig. 1, the structure of IOA comprises three main components: the user interface, the inference engine and the knowledge bases, plus a plugable component, known as a domain specific lexicon.

The *user interface* is a medium for communication between the user and the IOA. The IOA system employs an interactive window system interface which includes the use of pull down menus and a natural language interface. The system contains multiple menus for controlling a design session; viewing an evolving design model; saving and loading a design model.

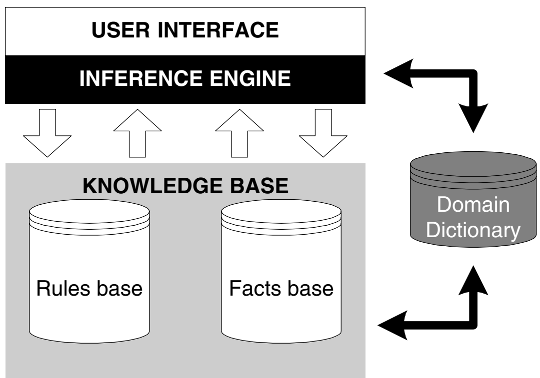

Fig. 1: The structure of the IOA system with plugable domain dictionary component

The IOA knowledge-base contains a mixture of rules and facts. Rules correspond to knowledge of how to perform the design task (the order in which design activities take place); detecting and resolving ambiguities, redundancies and inconsistencies within an evolving design; and handling the gradual augmentation of an evolving design as a design session progresses. Facts are used to represent two views of the application domain; an initial representation (the problem domain model) as provided by the user, and the object-oriented design generated from this initial representation.

The system is also being provided with a domain specific lexicon comprising of verb information in the format *is_verb (undergoes, pp, underwent)*, with both past and present forms of each verb being presented [10]. The use of such a lexicon will be discussed at a later section of this paper.

The inference engine of the IOA system acts as a controller that controls the interaction between the user and the system. It directs any part of the user input to the correct processor for processing and decides which rules to trigger.

The approach to design processing employed by similar type of tools or systems can be broadly categorised as design synthesis and design diagnosis [11]. Design synthesis is where the systems are capable of generating design output, whereas, design diagnosis is where the systems detect any inconsistencies and redundancies that may exist and suggest corrections in design. Although IOA was intended to support both aspects of design, being capable of synthesising an object-oriented conceptual schema and assisting users in the detection of the possible occurrences of inconsistencies, its major contribution is to assist users within the scope of design diagnosing from its capability to detect errors and suggest corrections in design. As will be discussed in the remaining part of this paper, this aspect is greatly achieved from the use of a series of rules and a lexicon containing domain specific verbs, previously discussed.

The processing employed by the IOA system is in accordance with the general knowledge-based database process described in [12], and are as follows:

- The first step involves creating an initial representation of the application domain (known as the problem domain model) and subsequent refinement of this model.
- The second step involves the creation of an analysis model, and subjecting this model with analysis and refinement rules to generate an object-oriented conceptual schema.

3.0 KNOWLEDGE-BASED DATABASE DESIGN DIAGNOSING

In general, there are four types of design inconsistencies (errors) that should be detected and resolved by automated database design tools [13]. The errors are as follows:

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

In the remaining of this section, each of these types of inconsistencies is discussed and how such inconsistencies can be intelligently detected and diagnosed are presented. The implementation of the diagnostic activities using the IOA tool previously described is illustrated.

3.1 Semantic Inconsistencies

The authors classify those concepts that do not participate in any relationships into this category. For example, Fig. 2 illustrates an isolated concept, "*Lecture*" which does not link with any other concepts.

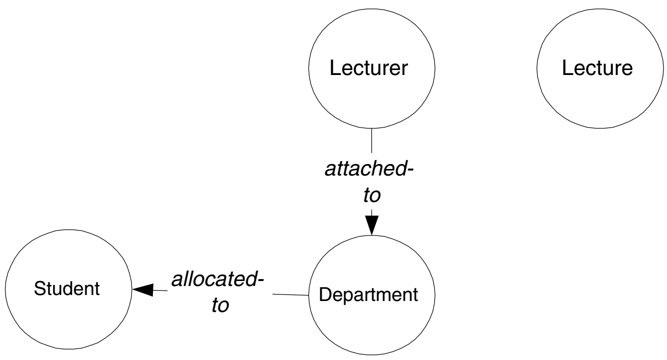

Fig. 2: An example of semantic inconsistencies (isolated concept)

In automated database design diagnosing, such a concept is detected by inspecting each node of the evolving design model and the corresponding links associated to it. Once the isolated concept is found, a series of interaction with the user will then take place in order to seek the user's decision on what to do next. As for example, the following dialogue is generated by IOA when the isolated concept of "*Lecture*" is being detected.

IOA > It was found that the concept lecture has no associated relationships. Do you wish to …
a) Provide the relationships associated with LECTURE now.
b) Remove the concept LECTURE from your specification.
c) Take no action for the time being
User > a

IOA > Please enter relationships associated with LECTURE – type END to finish
User > lecturer gives lecture
User > end

The other type of error considered in this category is the existence of transitivity. Transitivity holds in aggregation relationships [9, 14, 15], that is, if A is a component part of B and B is a component part of C, then A is also a component part of C. If all these three were to appear in the evolving design model, the relationship A is a component part of C is redundant and should be removed. Transitivity also holds in generalisation relationships [15] that is, if A is a subclass of B and B is a subclass of C, then A is also a subclass of C. If all of these three relationships were to appear in the evolving design model, the relationship A is a subclass of C is redundant and should be removed.

The detections of transitivity in KB database design diagnosing should not only consider a two-tier level of generalisation or aggregation hierarchies but in any n-tier level of hierarchies. Similar to the detection of concepts with missing links, IOA detects transitivity by inspecting each nodes and links that corresponds to the following rules.

Rule 1:

(i) Transitivity in a generalisation hierarchy
If A is a superclass of B and B is a superclass of C. Therefore, if there exist the generalisation relationship of A is a superclass of C. Such a relationship is redundant and should be removed from the evolving design model.

(ii) Transitivity in an aggregation hierarchy
If A is a component part of B and B is a component part of C. Therefore, if there exist the aggregation relationship of A is a component part of C. Such a relationship is redundant and should be removed from the evolving design model.

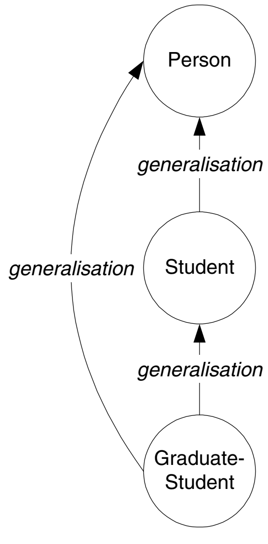

Fig. 3: An example of transitivity in a generalisation hierarchy

Fig. 3 illustrates an example of transitivity between the concepts "*Person*", "*Student*" and "*Graduate-Student*". Once such an occurrence of transitivity is being detected by IOA, a series of dialogue is generated to the user for confirmation of its removal as illustrated below.

IOA > It was found that GRADUATE-STUDENT is a STUDENT and GRADUATE-STUDENT is a PERSON, however, STUDENT is a PERSON. This exhibit redundancy. The structure GRDUATE-STUDEN is a PERSON should be removed. Do you agree? (Y/N)
User > Y

3.2 Inconsistent Concepts

Inconsistent concepts refer to concepts with no associated properties. For example the structure in Fig. 4 illustrates the inconsistent concept of "*Graduate-Student*". As a result the following dialogue is triggered by the IOA's diagnostic engine requesting confirmation from the user as whether to provide the property(ies) associated with the concept, or to remove the inconsistent concept from the evolving design model, or to take no immediate action.

IOA > It was found that the concept GRADUATE-STUDENT has no associated properties.
Do you wish to.....
a) Provide the properties associated with GRADUATE-STUDENT now.
b) Remove the concept GRADUATE-STUDENT from your specification.
c) Take no action for the time being.
User > a

IOA > Please enter properties associated with GRADUATE-STUDENT - type END to finish.
User > course-type
end

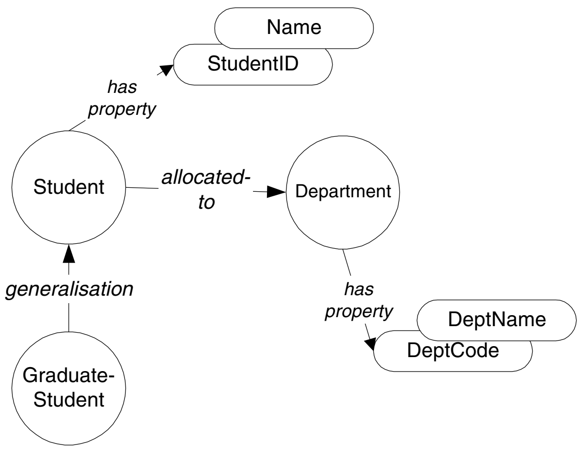

Fig. 4: An example of inconsistent concept

3.3 Redundant Inherited Properties and Relationships

Redundant inherited properties and relationships exist within generalisation hierarchies. For example if a concept "*Student*" has property "*studentId*", therefore the concept "*Graduate-Student*" which is the subclass of "*Student*" inherit the property "*studentId*" from the concept "*Student*". In this case, the property "*studentId*" should not be explicitly represented within the concept "*Graduate-Student*". Similarly any relationships that the concept "*Student*" participates, the concept "*Graduate-Student*" will also participate. From this basis, redundancy may occur in terms of inherited properties and inherited association and aggregation relationships of a generalisation hierarchy if a generic concept and its corresponding specific concept(s) explicitly contain the same attributes or participate with the same relationships.

The detection of the aforementioned elements of redundancies is achieved by the use of the following rules:

Rule 2:
If *x* is a property of *A* and *x* is also the property of *B*, but *A* is a subclass of *B*; then it could be that the property *x* in *A* is redundant and has to be removed from *A* since *A* can inherit the property *x* from *B*.

Rule 3:
If *C* is a component part of *A* and *C* is a component part of *B* and *A* is a subclass of *B* then it could be that the component *C* in *A* is redundant and has to be removed.

Rule 4:
If *A association C* and *B association C*, but *A* is a subclass of *B*, then it could be that *A association C* is redundant and has to be removed.

For example, the structures illustrated in Fig. 5, would eventually results in the following dialogues being triggered by the IOA's diagnostic engine.

IOA > The attribute NAME exists in the class PERSON and its subclass(es) (i.e. STUDENT). The attribute NAME should be removed from the subclass(es) because it can inherit the attribute NAME from class PERSON. Do you agree? (Y/N)
User > Y

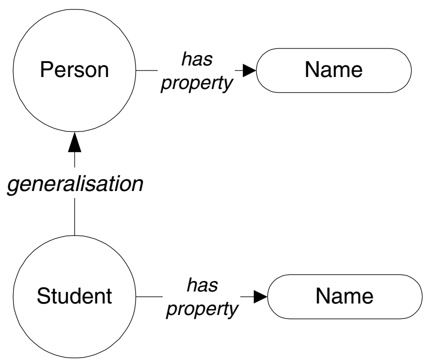

Fig. 5: An example of redundant inherited property

However, in cases where involving multilevel inheritance hierarchies or multiple subclasses as illustrated by the structures in Fig. 6, the following dialogue instead will be triggered by the IOA's diagnostic engine.

IOA > The attribute RESEARCH-TYPE appears in the concept GRADUATE-STUDENT and its superclass PERSON. This exhibit redundancy. Is it true to say that the attribute RESEARCH-TYPE is also an attribute of the other subclass(es) of the concept PERSON (i.e., STUDENT, PROFESSOR, and VISITING-PROFESSOR)? (Y/N)
User > N
IOA > In that case the attribute RESEARCH-TYPE will be removed from the concept PERSON and retained in the concept GRADUATE-STUDENT since the attribute RESEARCH-TYPE is not an attribute to the other subclasses(es) of the concept PERSON (i.e., STUDENT, PROFESSOR, and VISITING-PROFESSOR). Do you agree? (Y/N)
User > Y

Similar user/tool interactions are required during the detection and resolving of redundant inherited aggregation and association relationships within a generalisation hierarchy.

3.4 Redundant Elements

Redundant elements which are the result of synonyms such as synonymous concepts and relationships are conceivably the most complicated tasks performed in automated database design diagnosing. The most common way of detecting such design inconsistencies is perhaps by examining pairs of relationships between two distinct concepts of the form "*A verb-phrase B*" and "*C verb-phrase D*" [14] or sometimes referred to as object type and mismatch rules. For instance, comparing the structures of "*Lecturer Teaches Course*" and "*Academic Teaches Course*", would eventually suggest the synonymous between the concepts "*Lecturer*" and "*Academic*"; and comparing the structures of "*Student Enrols Course*" and "*Student Signs-up Course*" would result in the possible synonymous verb-phrases of "*Enrols*" and "*Signs-up*". However, there are cases where synonyms would not be the expected solutions. For example the structures of "*Student Registers Course*" and "*Graduate-Student Registers Course* " would normally viewed as the missing generalisation relationship between "*Student*" and "*Graduate-Student*".

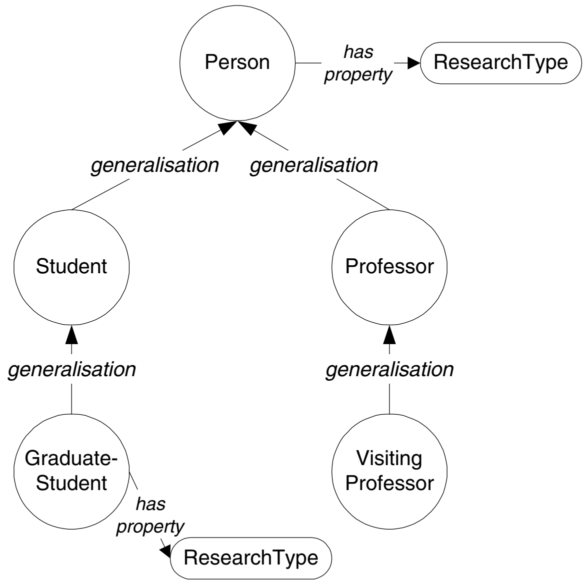

Fig. 6: An example of redundant inherited property in multilevel inheritance hierarchies

The rules used by the IOA's diagnostic engine to detect situations similar to the aforementioned examples are as follows:

Rule 5:
Let A v_1 B and X v_2 Y be the pair of associations structures, where A, B, X and Y are the concepts and v_1 and v_2 are the verb-phrases.

(i) If {A = X and $v_1 = v_2$} then B may be a synonym of Y (B = Y) or either one of the concepts is a subclass of the other concept.
(ii) If {A = Y and $v_1 = v_2$} then B may be a synonym of X (B = X) or either one of the concepts is a subclass of the other concept.
(iii) If {B = X and $v_1 = v_2$} then A may be a synonym of Y (A = Y) or either one of the concepts is a subclass of the other concept.
(iv) If {B = Y and $v_1 = v_2$} then A may be a synonym of X (A = X) or either one of the concepts is a subclass of the other concept.

Rule 6:
Let A v_1 B and X v_2 Y be the pair of associations structures, where A, B, X and Y are the concepts and v_1 and v_2 are the verb-phrases. Therefore, if (A = X and B = Y) or (A = Y and B = X) then it could be that v_1 is a synonym of v_2 ($v_1 = v_2$), or v_1 and v_2 are related by tenses.

Therefore, the existence of the structures "*Lecturer Teaches Course*" and "*Academic Teaches Course*" that match with Rule 5, as illustrated in Fig. 7, will be brought to the attention of the user who may give consideration to removing one of the redundant concepts as illustrated by the following dialogue:

IOA > An element of redundancy may occur in the following structures:
LECTURER TEACHES COURSE
ACADEMIC TEACHES COURSE

Which of the following is correct?
a) The concept LECTURER and ACADEMIC express the same thing.
b) One of the concepts is a specific type of the other concept or vice-versa.
c) One of the structures is wrong.
d) Leave it as it is.

User > a

IOA> Since LECTURER and ACADEMIC are synonymous, one of the concepts has to be removed. Which of the concepts would you like to remove?
a) LECTURER b) ACADEMIC

User> b

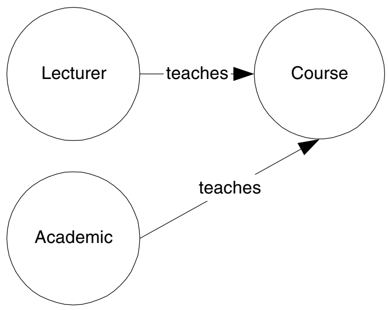

Fig. 7: Example of redundant elements corresponding to Rule 5

The structures of "*Student Registers Course*" and "*Graduate-Student Registers Course*", however, will result in a different user's solution. The following dialogue illustrates this situation.

IOA > An element of redundancy may occur in the following structures:
STUDENT REGISTERS COURSE
GRADUATE-STUDENT REGISTERS COURSE
Which of the following is correct?
a) The concept STUDENT and GRADUATE-STUDENT express the same thing.
b) One of the concepts is a specific type of the other concept or vice-versa.
c) One of the structures is wrong.
d) Leave as it is.

User > b

IOA> IOA will create a subclass/superclass relationship between the concepts STUDENT and GRADUATE-STUDENT. Which of the following is true?
a) GRADUATE-STUDENT is a specific type of STUDENT
b) STUDENT is a specific type of GRADUATE-STUDENT

User> a

In this case, IOA will update its evolving design model by incorporating the generalisation structure between the concepts "*Graduate-Student*" and "*Student*".

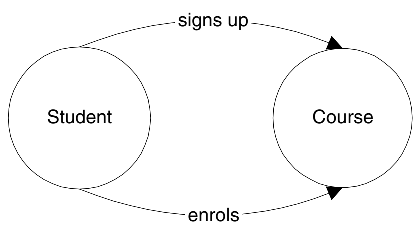

Fig. 8: Example of redundant elements corresponding to Rule 6

Rule 6, on the other hand, is used to detect the possible synonyms among verb-phrases relationships as previously illustrated by the structure of "*Student Enrols Course*" and "*Student Signs-up Course*" (Fig. 8). Similarly, the removal of such a redundant structure will be brought to the attention of the user for confirmation of its removal as illustrated by the following dialogue:

IOA > Do the structures STUDENT ENROLS COURSE and STUDENT SIGNS-UP COURSE represent the same information? Meaning that the words ENROLS and SIGNS-UP are synonymous or are related. (Y/N)
User > Y

IOA> Since the above statements represent the same information, one should be removed. Which structure would you like to remove?
a) STUDENT ENROLS COURSE b) STUDENT SIGNS-UP COURSE
User> b

As previously mentioned, the provision of lexicon of verbs is meant to enhance the diagnostic capabilities of the IOA system. Such a lexicon is mainly to ensure that the system is capable of identifying the similarities implied by the use of different tenses of the same verb [16]. For instance, if the structures of "*lecturer teaches course*" and "*course taught by academic*" are part of the problem domain model as illustrated in Fig. 9, the system would be able to identify the similarity implied by "*teaches*" and "*taught*", and to subsequently suggests that the concepts "*lecturer*" and "*academic*" are synonymous. Such structures are undiagnosed if a lexicon of verbs is not provided because the system is unable to detect the similarity implied by the use of the words "*teaches*" and "*taught*". The following provides an illustration of the dialogue between IOA and the user that ensues when IOA detects potential synonyms within the problem domain model:

IOA > From the following relationships:
LECTURER TEACHES COURSE
COURSE TAUGHT BY ACADEMIC
the system found that the verb/association TEACHES and TAUGHT may be related (e.g. they are related by time/tense).
If it is true that TEACHES and TAUGHT are related, what is the relationship between LECTURER and ACADEMIC?
a) LECTURER and ACADEMIC are synonymous.
b) LECTURER is a specific type of ACADEMIC.
c) ACADEMIC is a specific type of LECTURER.
d) The verbs TEACHES and TAUGHT are not related.
Please enter <a>, <b>, <c> or <d> and press <ENTER>.
User > a

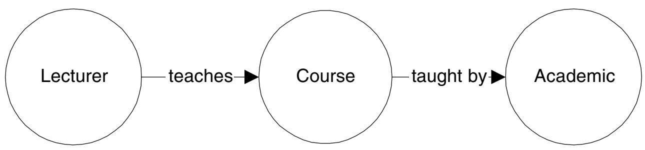

Fig. 9: Example of redundant elements capable of being detected with the use of domain lexicon

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 "*graduate student supervised by lecturer*" and "*lecturer supervises graduate student*" are detected by IOA, the requirement on the part of the user to confirm the similarities between "*supervised*" and "*supervises*" as illustrated previously is removed. Such a simplification is illustrated by the following dialogue:

IOA > After referring to the domain specific lexicon, the system found that the verbs/associations SUPERVISES and SUPERVISED are related. As a result the followings are redundant.
GRADUATE STUDENT SUPERVISED BY LECTURER
LECTURER SUPERVISES GRADUATE STUDENT
Please consider removing one of the above.
a) Remove the structure GRADUATE STUDENT SUPERVISED BY LECTURER.
b) Remove the structure LECTURER SUPERVISES GRADUATE STUDENT.
c) Leave as it is.
Please enter <a>, <b> or <c> and press <ENTER>.
User > a

4.0 DISCUSSIONS

Work to date has concentrated upon developing knowledge structures representing three different application domains (university, healthcare and library). Testing performed on a wide range of example applications related to the aforementioned domains has so far produced encouraging results [16, 17]. The testing has been focussed on assessing the significance of incorporating form of domain lexicon within the diagnostic processing. Therefore, only errors in the form of redundant elements have been considered. The postulation underpinning this decision is that the detection and removal of the other types of inconsistency (semantic inconsistencies, inconsistent concepts and redundant inherited properties and relationships) is not influenced by the existence of any forms of external knowledge such as the lexicon of verbs [13]. The detection of these types of errors is, however, a straightforward process which requires IOA to inspect each node of the problem domain model and the corresponding links associated with it. Detection of any inconsistency will then be brought to the attention of the user for confirmation of its removal.

During testing, IOA was exposed to a range of design problems within the scope of conceptual design. These design problems were extracted from the available literature, the advantage being that the accompanying solution could be used as a benchmark and compared with the IOA-suggested solution in order to confirm the appropriateness of the designs produced. In each case, the over-riding criterion was that the IOA-generated design should be comparable with the accompanying solution. Once this general criterion was satisfied, a number of test-cases were generated for each example, each case introducing a combination of different types and numbers of synthesised errors. The errors introduced included synonymous concepts, synonymous relationships and combinations of both. Appendix I illustrates the results of this test. As can be seen, exploiting the lexicon of verbs has managed to increase the diagnosis capability of the IOA system (measured in terms of the number of errors detected and resolved by the system) by an average of 37% for the university domain and 40% and 18% respectively for the healthcare and library domains.

The testing has also shown that the lexicon of verbs has managed to reduce the number of user/tool interactions required to diagnose and resolve errors by an average of 17% for the university domain and 22% and 15% respectively for the healthcare and library domains for each increase of error as compared to when no such lexicon is being used [16]. Fig. 10, 11 and 12 illustrate the results. Such results indicate that elements of intelligence have been increased with the use of domain lexicon whereby certain aspects of uncertainty can be ‘understood’ and resolved internally by the system without the requirement of users’ participations.

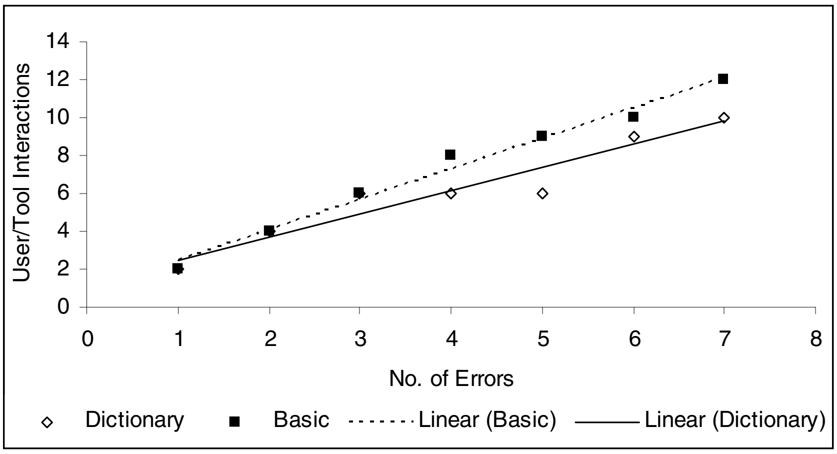

Fig. 10: User/tool interactions for the university domain

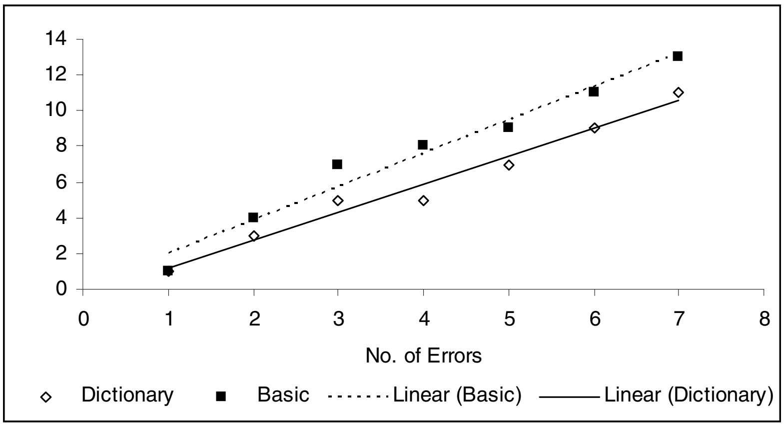

Fig. 11: User/tool interactions for the healthcare domain

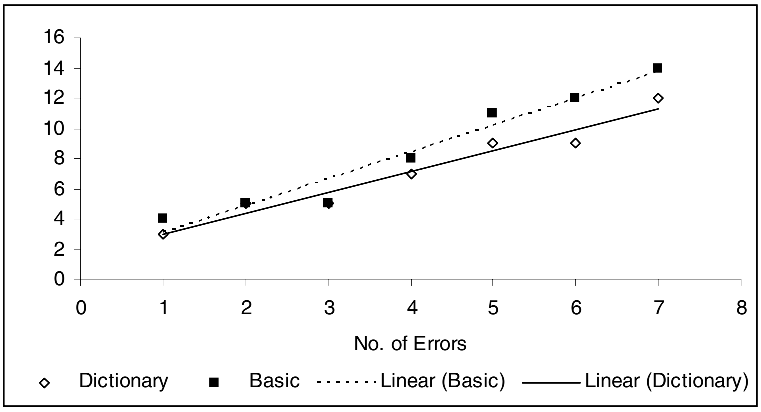

Fig. 12: User/tool interactions for the library domain

Despite the encouraging results obtained to date, the use of domain specific knowledge is not claimed to be a panacea [18]. Indeed it is recognised that there are certain aspects of the design process that are best resolved by user interactions such as multiplicity of relationships linking concepts, participations within a generalisation structure and differences of terms for the same concepts [18].

5.0 CONCLUSIONS AND FUTURE WORKS

This paper has presented the development of an automated database design system; the Intelligent Object Analyser (IOA). The system developed exhibits the practical application of artificial intelligence technique to the aspects of conceptual modelling in database design. During use, the IOA system has exhibited the capacity of performing two aspects of design (design synthesis and design diagnosis), being capable of synthesising a conceptual schema and assisting users in the detection of the possible occurrences of inconsistencies. However, the main contribution exhibited by the IOA system is within the aspect of database design diagnosing. Four main types of inconsistency have been discussed: semantic inconsistency; inconsistent concepts; redundant inherited properties and relationships; and redundant elements. The detection and diagnosis these types of inconsistency were practically demonstrated in this paper by the use of IOA system. In order to enhance the diagnostic capabilities of the IOA system, a lexicon consisting a series of verb-phrases has been integrated with the IOA knowledge-base. The inclusion of the lexicon of verbs proved to be one of the solutions towards enhancing the intelligence of the system in terms of detecting and resolving inconsistencies as exemplified by the results of a series of initial testing conducted.

There are a number of avenue exist for further research activities within this area, particularly with the research directions in database communities are more geared towards complex system such as the data warehousing. The complexity of data warehouse is due not only on the huge volume of integrated and aggregated information extracted from multiple, heterogeneous, autonomous and distributed information sources, but also from the multidimensionality of the data models [19]. Therefore, a more sophisticated and complex intelligent system is required particularly to cater the issues of integrated and multidimensionality of data models. This is not an easy task, as the subject matter is relatively lack of maturity and there is currently no universally accepted methodology to support such complexities in design [20].

This paper has also revealed that providing the system with simple elements of domain knowledge (in the form of domain specific verb-phrases) has the capability of enhancing the system's diagnostic capabilities. The lexicon of verbs, however, does not provide semantic information or meanings of a particular domain. Therefore other approaches in knowledge representation should be considered such as thesaurus-type structure, domain ontologies and naïve semantics [21, 22, 23].

Another desirable research direction in this area is the automatic augmentation of system-held domain knowledge. In this case the tool can learn from each of design activities, and previous design models can be reused again for other future design sessions. This seems to be a straightforward process. However, few feasibility studies should be taken into account and the following questions should be answered first [24]:

- How can design knowledge be modified to meet new insights into the problem or to remove undesired contents of the knowledge?
- How should implications of a change in the design knowledge be perceived?
- How to get users to participate in the augmentation of the design knowledge?

REFERENCES

[1] M. Lloyd-Williams and P. Beynon-Davies, "Expert System for Database Design: a Comparative Review". *Artificial Intelligence Review*, Vol. 6, 1992, pp. 263-283.

[2] G. Saake, "Conceptual Modelling of Databases Applications", in *Information System and Artificial Intelligence: Integration Aspects*, Berlin: Springer-Verlag, 1990, pp. 213-232.

[3] D. Batra and S. R. Antony, "Novice Errors in Conceptual Design". *European Journal of Information System*, Vol. 3, No. 1, 1994, pp. 57-69.

[4] D. Batra and S. H. Zanakis, "A Conceptual Database Design Approach Based on Rules and Heuristics". *European Journal of Information Systems*, Vol. 3, No. 3, 1994, pp. 228-239.

[5] V. C. Storey and R. C. Goldstein, "Knowledge-Based Approach to Database Design". Management Information Systems Quarterly, Vol. 17, No. 1, 1993, pp. 25-46.

[6] S. Ram and V. Ramesh, "Collaborative Conceptual Schema Design: A Process Model and Prototype System". *ACM Transactions on Information System*, Vol. 16, No. 4,1998, 347-371.

[7] W. Tatjana, I. Rozman and M. Druzovec, "Database Reuse as a Support for Conceptual Database Modelling", in: *Proceedings of the 4th Joint Conference on Knowledge-Based Software Engineering*, Amsterdam: IOS Press, 2000, pp. 129-132.

[8] M. Lowry and R. K. Duran. "Knowledge Based Software Engineering". *The Handbook of Artificial Intelligence: Volumen IV*. Reading, MA: Addison-Wesley, 1989, pp. 241-322.

[9] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy and W. Lorenson, "*Object-Oriented Modelling and Design*", Englewood-Cliffs: Prentice-Hall, 1993.

[10] A. Kawaguchi, N. Taoka, R. Mizoguchi, T. Yamaguchi and O. Kokusho, "An Intelligent Interview System for Conceptual Design of Databases", in: *ECAI'86: The 7th European Conference on Artificial Intelligence*, 1986, pp. 1-7.

[11] R. Oxman and J. S. Gero, "Using an Expert System for Design Diagnosis and Synthesis". *Expert Systems: The International Journal of Knowledge Engineering*, Vol. 4, No. 1, 1987, pp. 4-14.

[12] M. Lloyd-Williams and P. Beynon-Davies, "Expert System for Database Design: A Comparative Review". *Artificial Intelligence Review*, Vol. 6, 1992, pp. 263-283.

[13] S. A. Noah and M. Williams, "Exploring and Validating the Contributions of Real-World Knowledge to the Diagnostic Performance of Automated Database Design Tools", in *Proceedings of 15th IEEE International Conference on Automated Software Engineering*, IEEE: New York, 2000, pp.177-186.

[14] V. C. Storey, "Understanding Semantic Relationships". *Very Large Database Journal,* Vol. 2, No. 4, 1993, pp. 455-488.

[15] M. E. Winston, "A Taxonomy of Part-Whole Relations". *Cognitive Science,* Vol. 11, 1987, pp. 417-444.

[16] S. A. Noah and M. Lloyd-Williams, "Exploitation of Domain Knowledge by Knowledge-Based Database Design Tools: the Dictionary Approach", in *Proceedings of the 3rd Joint Conference on Knowledge-Based Software Engineering, Smolenice, Slovakia*, 1998, pp. 250-257.

[17] S. A. Noah and M. Williams, "Exploiting Real-World Knowledge to Enhance the Intelligence of Knowledge-Based Database Design Tool: A Validation", in Proceedings of *The 4th Joint Conference on Knowledge-Based Software Engineering*, Amsterdam: IOS Press, 2000, pp. 155-162.

[18] M.Lloyd-Williams, "Exploting Domain Knowledge During Automated Design of Object-Oriented Databases", in *Proceedings of the 16th International Conference on Conceptual Modelling*. Berlin: Springer Verlag, 1997, pp. 16-29.

[19] B. Hüsemann J. Lechtenbörger and G. Vossen, "Conceptual Data Warehouse Design", in *Proceedings of the International Workshop on Design and Management of Data Warehouse*, Stockholm, Sweden, 2000, pp. 1-11.

[20] E. Franconi & G. Ng, "The i•com Tool for Intelligent Conceptual Modelling", in *Proceedings of The 7th International Workshop on Knowledge Representation Meets Databases (KRDB'00)*, Berlin: Springer-Verlag, 2000, pp. 45-53.

[21] M. Lloyd-Williams, "Knowledge-Based CASE Tools: Improving Performance Using Domain Specific Knowledge". *Software Engineering Journal*, Vol. 9, No. 4, 1994, pp. 167-173.

[22] Y. Wand V. C. Storey and R. Weber, "An Ontological Analysis of the Relationship Construct in Conceptual Modelling". *ACM Transactions on Database Systems*, Vol. 24, No. 4, 1999, pp. 494-528.

[23] V.C. Storey, R.C. Goldstein and H. Ullrich, "Naïve Semantics to Support Automated Database Design". *IEEE Transactions on Knowledge and Data Engineering*, Vol. 14, No. 1, 2002, pp. 13-27.

[24] J. Vanwelkenhuysen, "Using DRE to Augment Generic Conceptual Design". *IEEE Expert*, Vol. 10, No. 1, 1995, pp. 50-56.

BIOGRAPHY

Shahrul Azman Mohd. Noah is a lecturer at the Faculty of Information Science and Technology, National University of Malaysia, and currently heads the Department of Information Science within the faculty. He received his MSc and PhD in Information Studies from the University of Sheffield, UK in 1994 and 1998 respectively. His research interests include conceptual modelling, data warehouse design, knowledge representation, and agent based information retrieval. He is also actively involved in the health informatics research and development projects. He is a member of the IEEE Computer Society.

Michael D. Williams is a senior lecturer in the European Business Management School at the University of Wales Swansea. He holds a BSc from the CNAA, an MEd from the University of Cambridge, and a PhD from the University of Sheffield. He is currently a member of the British Computer Society and is registered as a chartered engineer. Prior to entering academia Dr. Williams spent twelve years developing and implementing information systems in both public and private sectors in a variety of domains including finance, telecommunications, manufacturing, and local government. He is the author of numerous fully refereed and invited papers within the ICT domain. His current research interests are focused primarily upon evaluating the organisational and societal impact of ICT introduction and use, including the introduction of e-commerce initiatives

Appendix I (Results of the Testing)

Domain: University

Test No.	Total Generated Errors	Diagnosed Errors (Basic)	Diagnosed Errors (Dictionary)
1	2	0	0
2	4	2	2
3	5	2	2
4	6	3	3
5	7	4	4
6	9	4	6
7	10	4	10
8	11	4	11
9	3	0	0
10	4	1	1
11	5	2	2
12	7	2	4
13	8	2	6
14	9	2	7
15	1	1	1
16	2	2	2
17	4	2	4
18	6	2	6
19	7	2	7
20	2	0	2
21	4	0	4
22	5	0	5
23	3	0	0
24	4	0	0
TOTALS	128	41	89
Percentage Resolved		**32%**	**69%**

Domain: Healthcare

Test No.	Total Generated Errors	Diagnosed Errors (Basic)	Diagnosed Errors (Dictionary)
1	2	2	2
2	4	4	4
3	7	4	4
4	8	5	5
5	9	5	9
6	11	5	11
7	12	5	12
8	3	0	0
9	4	1	1
10	5	1	5
11	7	1	7
12	9	1	9
13	3	0	0
14	6	0	0
TOTALS	90	34	69
Percentage Resolved		**37%**	**77%**

Domain: Library

Test No.	Total Generated Errors	Diagnosed Errors (Basic)	Diagnosed Errors (Dictionary)
1	2	2	2
2	3	2	3
3	5	4	5
4	7	4	6
5	8	5	6
6	11	7	9
7	13	7	9
8	3	1	2
9	5	2	3
10	7	3	5
11	9	5	5
12	10	5	8
13	12	7	9
14	13	7	9
15	14	7	9
TOTALS	122	68	90
Percentage Resolved		**56%**	**74%**