THE RETRIEVAL OF STRUCTURED DESIGN KNOWLEDGE

Hongwei Wang[1, 2], Aylmer L. Johnson[1] and Rob H. Bracewell[1]
(1) University of Cambridge, UK (2) University of Portsmouth, UK

ABSTRACT
Design rationale is an effective way of capturing knowledge, since it records the issues addressed, the options considered, and the arguments used when specific decisions are made during the design process. Design rationale is generally captured by identifying elements and their dependencies, i.e. in a structured way. Current retrieval methods focus mainly on either the classification of rationale or on keyword-based searches of records. Keyword-based retrieval is reasonably effective as the information in design rationale records is mainly described using text. However, most of the current keyword-based retrieval methods discard the implicit structures of these records, resulting either in poor precision of retrieval or in isolated pieces of information that are difficult to understand. This ongoing research aims to go beyond keyword-based retrieval by developing methods and tools to facilitate the provision of useful design knowledge in new design projects. Our first step is to understand the structured information derived from the relationship between lumps of text held in different nodes in the design rationale captured via a software tool currently used in industry, and study how this information can be utilised to improve retrieval performance. Specifically, methods for utilising various structured information are developed and implemented on a prototype keyword-based retrieval system developed in our earlier work. The implementation and evaluation of these methods shows that the structured information can be utilised in a number of ways, such as filtering the results and providing more complete information. This allows the retrieval system to present results that are easy to understand, and which closely match designers' queries. Like design rationale, other methods for representing design knowledge also in essence involve structured information and thus the methods proposed can be generalised to be adapted and applied for the retrieval of other kinds of design knowledge.

Keywords: Engineering design, knowledge management, design rationale, information retrieval

1 INTRODUCTION
An engineering design process essentially consists of a large number of activities related to solving problems and making decisions. Most of these activities are dependent upon previous knowledge and experiences, and require the effective and efficient processing of information. A designer's knowledge of what issues should be addressed, how specific solutions are generated, as well as why a particular solution should (or might not) work, is a valuable intellectual asset of an enterprise, and can be recorded as design rationale. Design rationale can be defined from a variety of views, e.g. "an explanation of why an artefact, or some part of an artefact, is designed the way it is", or "what includes all the background knowledge such as deliberating, reasoning, trade-off, and decision-making in the design process of an artefact - information that can be valuable, even critical, to various people who deal with the artefact" [1]. Research about how to capture, store, and re-use design rationale has been undertaken in a wide range of domains, e.g. social science, software engineering [2], engineering design [1], etc.

Design rationale can offer designers useful information about how previous designs evolved and in what context such evolution happened. This information is derived from the expertise of designers, and therefore can be viewed as a source of design knowledge. An important advantage of representing design knowledge using design rationale is that it help capture lots of engineering know-why. It is also deemed to be a necessary part of the information useful for identifying the explicit linkage between the design record and the emergent outcomes as seen in service [3]. Arora et al. listed several benefits of design rationale: firstly, it will help in recording the design decisions and the reasoning behind the design, which can be later analyzed to control the overall design and manage the complexity of the design process; secondly, it will also have an impact on maintenance, which has been observed to be very much dependent on the design; thirdly, capturing design rationale and providing a suitable representation scheme will also help in reverse engineering of the design and creating libraries of design-process artefacts for design re-use and design traceability; fourthly, the rationale can be used to reason about the changes made by the designer, based on the generated knowledge base [4].

A large scope of research on design rationale has been undertaken and published elsewhere, e.g., the devising of rationale models [5, 6], and the development of systems for capturing design rationale [7, 8]. A comprehensive review of the research of design rationale for engineering design was done by Regli et al. [1]. Compared with capturing design rationale, its subsequent retrieval has gained less attention. This is partially due to the lack of a unified model to represent design rationale. However, the retrieval of design rationale is as important as its capture because re-use is the ultimate goal of capturing and storing knowledge assets. Meanwhile, a number of design rationale capturing tools, e.g. the Design Rationale editor (DRed) developed by researchers in the Engineering Design Centre (EDC) of Cambridge University [9], are beginning to be tested and accepted in industry. Therefore, it is necessary and timely to study how design rationale records can be readily retrieved.

Some research work has already been undertaken to understand the information needs of designers to make classification for design knowledge, as well as to provide insights for the development of tools supporting design information retrieval [10]. Currently, the keyword-based retrieval is becoming increasingly mature as powerful methods developed by the information retrieval community become more readily available. Nevertheless, these methods still have some drawbacks. Firstly, previous research indicates that designers cannot always express their information needs explicitly [10]. Secondly, the pieces of design rationale found are isolated, and do not always match designs' queries well. One of the reasons for this is that the implicit structure in design rationale is not well utilised. Therefore, the motivation of this research is to see what structures information exist in design rationale records and how can they be utilised in retrieval. This paper reports our ongoing work on developing computer aids for engineering designers to effectively and efficiently retrieve design rationale captured in previous design projects.

2 RELATED WORK

This research is related to studies on knowledge management, design rationale, and design information retrieval. A design process essentially involves lots of tasks related to generating solutions and making decisions. The increasingly complex nature of modern artefacts entails a huge amount of knowledge to support these tasks. Knowledge Management (KM) in engineering design has been studied for a long while, with two specific focuses, namely the personalisation aspect and the codification aspect [11]. Knowledge has been recognised as an important part of product development systems [12]. A specific application of KM in engineering is Knowledge-Based Engineering (KBE) which embeds domain knowledge in systems providing computer aids for designers. For instance, Susca et al. presented a KBE application which was used to perform an automatic calculation and evaluation of the mass properties of racing cars [13]. Marx et al. developed a knowledge-based system, which was integrated with numerical analysis codes, to evaluate aircraft structural concept, material and process selections [14]. Skarka studied the automatic generation of design models based on the Methodology and tools Oriented to Knowledge-based engineering Applications (MOKA) framework [15].

Along with elaborating a more comprehensive representation of design knowledge, e.g. the function, form, and behaviour of artefacts, there is also a need to capture design rationale [12]. Design Rationale System (DRS) is also widely studied and developed to facilitate the tasks of capturing, storing and retrieving design knowledge. Design rationale or design intent encompasses a wide range of context surrounding the decisions made during a design process. The capture of design rationale is, therefore, related to how a design process is described. For instance, Brazier et al. proposed a Generic Task Model for Design (GTMD) which consisted of three subtasks, such that design rationale could be captured from these subtasks [16]. Ganeshan et al. used a design decision-making framework to record design histories. Design intent could then be captured from the two characteristics of the framework, namely the objectives and manipulation of those objectives [17]. As design rationale can be used to illustrate any decision-making tasks, it can be utilised for the capture of design knowledge in different design stages. A classification of design rationale is useful in the sense that it can provide insights for the development of DRS. Shipman and McCall suggested seeing design rationale from three points of view, namely the argumentation perspective, the documentation perspective, and the communication perspective [18]. These perspectives have complementary advantages in facilitating either the capture or the retrieval of design rationale. They recommended an integration of different perspectives and constructed the PHIDIAS system. The representations methods for design rationale are developed in light of the perspective they emphasise; interested readers are referred to the review paper of Regli et al. [1].

Information Retrieval (IR) is an interdisciplinary research field which concerns computer science, mathematical science, library science, linguistics, cognitive psychology, and physics. Lots of research work has been done to study the retrieval of design information. For instance, Charlton systematically studied the retrieval of various types of mechanical design information, from textual information to the structured representation of a design [19]. Data mining techniques and Natural Language Processing (NLP) were employed in his study. Similar to product knowledge and process knowledge, design rationale has also been viewed as an important source of design information. Giess et al. proposed to identify and link representation elements of product, process and rationale to support design learning [20]. Huet et al. identified design rationale as a key output of design review meetings [21]. In the review paper of Regli et al., they summarized three design rationale retrieval methods, namely a navigation approach, a query-based approach, and a hybrid approach [1]. A navigation approach involves permitting designers to explore design rationale by traversing from one node to another via existing links. A query-based approach provides retrieval strategies according to designers' queries. Approaches using automatic triggers can detect or monitor certain conditions according to the design context. Hybrid approaches provide retrieval strategies based on the combination of the above two approaches.

Kim et al. proposed two methods for the retrieval of design rationale captured using the DRed tool. The first approach is query-based, using NLP techniques to evaluate the similarity of rationale records [22]. The second approach is proposed by analysing the task models of design re-use, recommending supporting the re-use of design rationale within a design process [23]. Two prototype systems were developed to demonstrate these methods. The first approach essentially involves establishing semantic annotation for DRed files while the second approach tries to statistically anticipate the next design task likely to be performed by designers. These two approaches are shown to be successful in retrieving DRed files with different focuses. However, there still exist drawbacks in these methods. The query-based approach employs comprehensive NLP and machine learning techniques to establish semantic annotation for documents, but this only works when users can submit an explicit query. The second approach seeks to understand the re-using behaviour of designers, and provides automatic suggestions for them, but the method may omit some useful information and does not consider the dependencies between DRed elements.

In our earlier work, a keyword-based retrieval tool was developed for the design rationale captured using DRed [24]. A few methods were developed to improve the keyword-based retrieval, namely suggesting potential keywords to designers, quantifying the relevance of retrieved information, and recommending relevant information based on the dependencies. Such a retrieval system can find a set of nodes in response to users' queries and each of these nodes represents a basic element (an issue, answer, or argument) of a piece of rationale. Although the tool can rank the results based on the degree of matching the queries, it fails to tell the types and statuses of the node and more important the information in a single node is incomplete to make users understand its meaning. This can be further improved by taking into account the implicit structure of design rationale records.

3 METHODOLOGY

The ultimate goal of this research is to develop an efficient retrieval method for designers who are using DRed in their daily design tasks. In this paper, the utilisation of implicit structure of design rationale in particular will be discussed. The methods proposed will be evaluated by developing a prototype tool, with an ultimate aim of integrating them into the DRed tool. The methodology will be introduced in this section. Firstly, searching strategies for DRed files will be discussed. Then the structured information in DRed files will be analysed. Lastly, methods for utilising this information will be introduced.

3.1 Searching strategies for DRed files

The development of DRed was first driven by a prime requirement, as identified in a collaborative research project investigating the capturing, sharing and re-use of design knowledge in the aerospace industry, to provide a tool capable of capturing design rationale. A study of the existing aero engine design reports showed that a large portion of their contents would map naturally onto the Issue-Based Information System (IBIS) structure, so the rationale was modelled as the issues to be solved, the potential solutions to those issues, and the arguments for or against these solutions. The prime advantage of DRed is its simplicity, which allows it not to obstruct a design and was intended to be used by designers as the design proceeds, not just retrospectively. It mainly consists of four steps for capturing design rationale when a design project proceeds, namely (1) diagnosing the problem; (2) designing a solution; (3) completing a standard checklist template; (4) and communicating the final design and its rationale [25]. The creation of a design record is useful for product development in the future especially when this record could be evaluated using in-service information. Design rationale is identified as a necessary part of the design record, together with product information and process information [3]. Recent extensions to DRed include making bidirectional links between DRed graphs and external files, i.e. spreadsheets, MS Word documents, and CAD files, to enable designers to obtain necessary information from an integrated design space [26].

The rationale captured using DRed are represented using a directed graph of dependencies and stored as plain text files with a .dre postfix in the computer's filing system. Each of these files is termed a DRed file which can either be saved on a local computer or published via a Web server. A project involving a large rationale space can be decomposed into a number of smaller pieces in general, with each piece constructed as a separate DRed file. Nodes (termed DRed nodes) of the graph are the basic elements of a piece of design rationale and an earlier keyword-based retrieval system finds the nodes of interest in response to users' queries. Apart from finding and ranking DRed nodes using the state-of-the-art information retrieval techniques, the retrieval system can also recommend potential keywords for users, and suggest other DRed nodes correlating to the one currently viewed by them. For more details about DRed works and the keyword-based retrieval method, interested readers are referred to our earlier work published elsewhere [24]. Though such a method can find information with fine granularity (i.e. each node is viewed as a separate result), it still has some drawbacks. Firstly, the DRed nodes are found and ranked based on the degree to which their contents match the query, regardless of their types and statues. This can affect the retrieval performance when a few useful DRed nodes are returned together with a large number of other nodes with similar degree of matching. Secondly, the information contained by a single node is not adequate, which results in the difficulty of understanding what this node is about. The information about DRed nodes' types, statuses, positions in a graph, connections with other nodes, is actually very useful for the re-use of design rationale and thus should be taken into account by a retrieval system.

3.2 Understanding the implicit structure in DRed graphs

To make use of the structured information of DRed graphs, it is necessary at the first place to identify this information, i.e. what can distinct a DRed node from others apart from its textual contents. Design rationale records captured using DRed are organised as graphs with dependencies and are displayed in the Graphical User Interface (GUI) of DRed to be viewed by engineers. Engineers can understand how previous solutions were developed by reading contents in each node, together with its type and status. Therefore, such a distinction can generally be made by comparing a node's type, status, position in a graph, and connections with those of others. Based on this information, a retrieval system can infer the purpose of a DRed node created, and use the inferences to determine the degree to which a DRed node matches the queries submitted by users. The inferences made on different nodes with different types and statuses are listed in Table 1. The 'file' and 'text' nodes are not included in the table as they are used for either illustration or explanation and those nodes used by a newer version of DRed for Functional Analysis Diagram (FAD) are also excluded as design rationale information is targeted in this research.

Table 1: Inferences made on different DRed nodes with different structured information

Type of node	Inferences on different status of a node
Issue	**'Resolved'**: the issue is resolved by an effective solution and should be very useful to engineers. **'Open'**: the issue node still requires further work. **'Rejected'**, the issue should not be taken into account and should be useful to engineers. **'Insoluble'**, no effective solutions can be found and it is useful for reminding users working on similar problems.
Answer	**'Accepted'**: the answer is accepted as an effective solution and should be very useful to engineers. **'Likely'**: the answer can likely be used to resolve the issue but is still not accepted. **'Open'**: the answer node still requires further work. **'Unlikely'**, the answer can hardly be used to resolve the issue, but is still not rejected. **'Rejected'**, the answer is rejected and is useful for reminding users working on similar problems.
Pro-argument	**'Holds'**: the pro-argument is proved to be valid and is very useful for convincing users. **'Dominant'**: the pro-argument plays a major role in confirming the validity of a statement or solution. **'Fails'**, the pro-argument fails to support a statement or solution, and can be used to remind users with similar ideas.
Con-argument	**'Holds'**: the con-argument is proved to be valid and is very useful for convincing users. **'Dominant'**: the con-argument plays a major role in confirming the invalidity of a statement or solution. **'Fails'**, the con-argument fails to prove the invalidity of a statement or solution, and can be used to remind users with similar ideas.

Apart from the types and statues of DRed nodes, other information, e.g. . a node's position in a graph and connections with other nodes, can also be utilised to infer its importance and complexity. For example, if a keyword (e.g. 'combustion') appears in the top node of a DRed graph, it is very likely that the whole rationale record is trying to resolve a problem about combustion. On the other hand, if it appears at a lower level of a DRed graph, combustion may be just part of the problem or a further issue raised by an answer proposed. Likely, if many connections are created for a node, this probably means that this node is about a complex topic requiring many arguments or raising many further issues. It is noteworthy that the connections created for a node do not necessarily indicate its complexity as it is also possible some design rationale is captured in more details whereas some is not. Nevertheless, when two nodes are found to be equally important to a query (with all other factors taken into account), this information can still be helpful as a further evidence of importance.

If the structured information of DRed graphs can help with knowledge retrieval, then two questions need to be answered: firstly, under what circumstances will the information be required; and secondly, how will the information then be used. An initial problem is the amount of priority that should be given to a node which appears to be potentially important, and at the same time, how should different factors (e.g. the type and status of node, its position in a graph, etc.) be weighted, when evaluating the potential relevance of a node. As discussed above, if a keyword (or a group of keywords) appears in the top node of a graph, this graph must be very relevant to the information needs of the user. If this inference holds, it is clear that if a keyword (or a set of keywords) appears in several nodes in a graph, then this graph is very likely to be of interest to the user. A second problem is thus raised by these inferences; that is, how to determine whether and how a keyword (or a set of keywords) appears in several nodes, i.e. what searching strategy should be developed. A third problem is how to evaluate the similarity between two graphs or sub-graphs, given that a set of words appears in both of them.

3.3 Utilising the implicit structure in DRed graphs

As DRed nodes have different types, a first step is to classify them in terms of these types. Such a classification can not only filter the nodes found in response to a query, but can also help users to submit queries that better reflect their information needs. For example, if a designer wants to find out whether a particular issue has been addressed in any previous projects, they can simply inform the retrieval system that only 'issue' nodes need to be inspected. However, there still exist some circumstances under which designers do not know exactly what to search for. Therefore they need to have the option of not filtering the DRed nodes, so as to achieve a better possibility of finding useful information. In this research, we assume that there are two main reasons for design engineers to search DRed graphs: firstly, they have a problem but do not know how to resolve it, so they wonder whether similar problems have been identified and resolved in previous projects; and secondly, they have a problem together with an answer but are not sure about the effectiveness of the answer, so they would like to see whether similar problems have been encountered before – and if so, whether similar solutions were used on those occasions.

Therefore, 'issue' nodes will be given highest priority as it is believed that engineers tend to work in a problem-oriented way. 'Answer' nodes will be given medium priority as solutions are also things that engineers would like to find and analyse. 'Argument' nodes will be given lowest priority, with pro-arguments and con-arguments being treated equally. Apart from grouping DRed nodes based on their different types, there are two more methods which can also be used to classify DRed nodes. The first one can be done automatically by calculating the similarity between two DRed nodes. Another method for classifying DRed nodes is to attach them to specific subjects that are created by users. For example, users can create subjects like 'manufacturing', 'service', 'combustion', 'cooling', etc., and then assign different subjects (each of which corresponds to a specific subject) to each DRed node they read.

The statuses of DRed nodes can also be utilised in the classification discussed above. For example, 'issue' node groups can be classified as 'resolved issues' or 'open issues', and be assigned different priorities (e.g. the retrieval system may deem resolved issues to be more important than open ones). Information about either the position of a node in a DRed graph or other nodes interconnected with it can also be used to give different priorities to DRed nodes. Another method of using the status information is to generate a summary for each node, in addition to its text. This summary might include the type of the node, its status, and some important nodes connected to it. Such summary can not only help users to understand the significance of a particular DRed node better, but also help them by telling them more stories. For example, a summarisation might include some text such as 'this is an issue node about dust cap; please note that this issue was proved to be irresolvable'. In this way, users can quickly ignore this node if they are only looking for solvable issues, whilst still being able to get further details if they would like to see why no effective solutions were found for this issue.

As introduced above, design rationale in DRed is represented as a directed graph with dependencies and therefore any DRed node has a specific level in a graph whilst having interconnections with other nodes. The levels of DRed nodes essentially reflect the stages of a problem-solving process. Higher levels mean earlier stages of the process whereas lower levels correspond to later stages. Nodes at high levels tend to describe a design problem with a high level of abstraction whereas those at low levels are usually aimed at describing solutions and arguments with more details. The interconnections between DRed nodes not only indicate their dependencies but also reflect the complexity of a particular node. Specifically, a DRed node is very likely to describe a complex issue if it has many successors derived from it. If a node is derived from two or more predecessors, it can be inferred that either this node describes a common fact (or argument) or its predecessors all pertain a similar topic. This is useful for the retrieval system which can highly recommend the nodes with more interactions. Another usage of the interconnections of a DRed node is to put it in context like putting a few sentences in a paragraph. A single DRed node does not contain much information, consisting of a few sentences used to introduce an issue, to describe a solution, as well as to make arguments. When users navigate a DRed graph, they look at a node, try to understand its contents, and quickly move to another one to reason using information accumulated. Therefore, a group of interrelated DRed nodes, i.e. a sub-graph in a DRed graph, can help put any single node in context and make the information easier to understand. If the retrieval system is able to try to understand the information contained in a DRed graph (or sub-graph) and match it with the query, it will find more precise information for users. In this way, a sub-graph can be used as the response to users' query. If this can be implemented, users can even submit a piece of design rationale as a query to better express their information needs.

4 DEVELOPMENT AND EVALUATION

4.1 Development of the prototype system

To verify the proposed retrieval methods, a prototype system is being developed based on an earlier keyword-based system, and some preliminary evaluation has now been performed. The prototype is implemented using Java and can run on multiple platforms. A snapshot of the GUI of this prototype system is shown in Fig. 1, with some annotations to some specific parts of the GUI. Though it is currently developed as a standalone application, it can be extended as a Web-based collaborative tool and integrated with DRed once the methods are deemed to be valid and helpful. A detailed demonstration of the functionality of the keyword-based system is beyond the scope of this paper, and is published elsewhere [24]. As shown in the figure, there is an option for users to choose what kind of DRed nodes (i.e. issues, answers, pro and con arguments) they would like to find. Currently, the summary generated simply includes the type and status information of a node, which enables users to ignore some results at a first glance. Below the contents of DRed nodes, there are two links. The first can be clicked to open the DRed file where the node exists. The second link is used to show other DRed nodes correlated with the one currently viewed by users. Searching nodes groups in response to queries is also implemented in the current version of the prototype system.

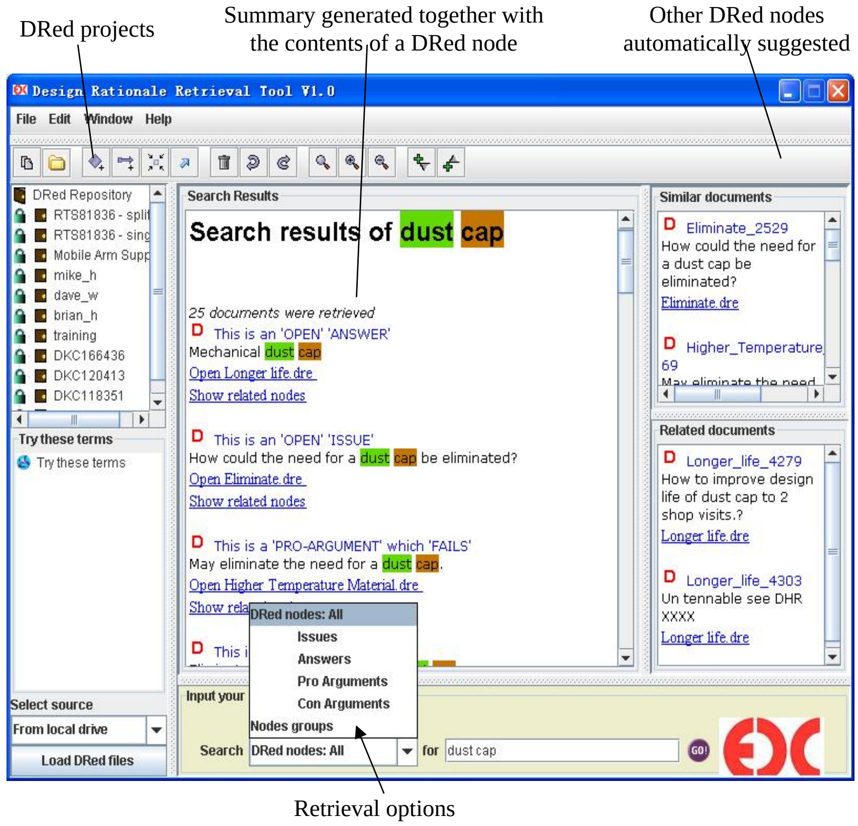

Figure 1. Graphical user interface of the retrieval tool

4.2 Evaluation of the prototype system

An evaluation of the keyword-based retrieval method for both its functionality and performance was given in [24]. In this section, the prototype system will be evaluated to see whether the utilisation of implicit structures can improve the retrieval. As shown in Fig. 1, using summaries for each DRed node can provide better information for users, compared with the method used by the keyword-based prototype. Furthermore, retrieval results can be filtered by using the type information. The retrieval results for a number of randomly formed queries are shown in Fig. 2. With the support of this functionality, users can focus on the particular kinds of DRed nodes of interest. The usefulness of this functionality is especially apparent when large numbers (e.g. several hundreds) of DRed nodes are found in response to a query. In Fig.3, the results (each of them is called a nodes group) found for the query “dust cap” are shown, each of which consists of more than one node and is displayed on the GUI of DRed. Using the nodes groups, the contents of a DRed node together with its context (other nodes connected with it) are all clearly shown to users. This achieves a much better performance than showing lots of DRed nodes listed one by one, such that the useful ones cannot be easily found. As evidenced in this evaluation, the new methods are useful and the algorithms used for implementing them are not difficult to implement. Currently, we are developing the functionality of supporting the using of complex queries to better express users’ needs and help find more precise results. More details about these methods and algorithms used will be presented in our other publications in the future.

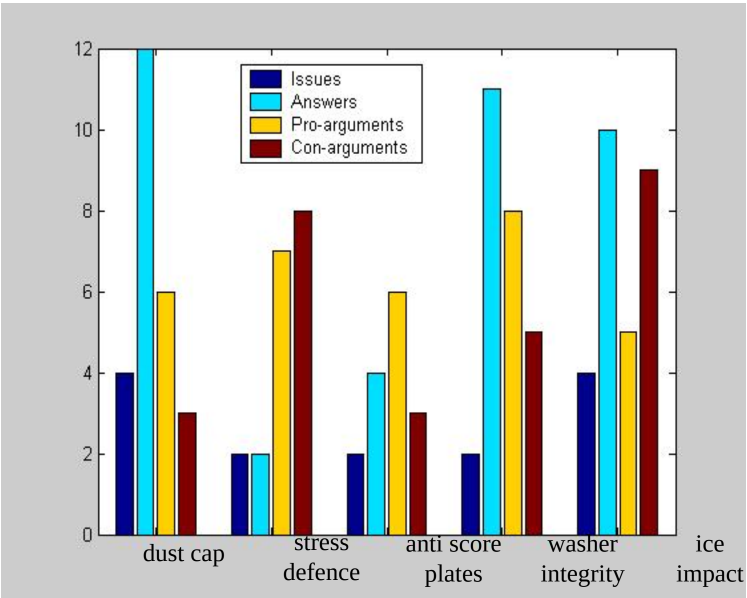

Figure 2. Retrieval results when different queries are used

5 CONCLUSION AND FUTURE WORK

An important advantage of using design rationale as a source of engineering knowledge is that it contains a lot of engineering ‘know-why’ which is captured as a by-product of the normal process of designing. The wide adoption of design rationale capturing tools in industry makes it very desirable to develop methods and tools for the retrieval and re-use of these records. A keyword-based retrieval method can be effective, as such records are mainly stored as plain text. However, such a method is not very efficient when large numbers of design rationale records have been captured and stored. : the results of interest will be hard to find, amongst the large quantity of results returned. Moreover, the contents of a single node are often not sufficient to allow users to understand the issues properly. As evidenced in the development and evaluation of a prototype system, the utilisation the of implicit structures in DRed graphs is useful and can help improve retrieval performance. In our current research, the structured information is utilised mainly in four ways. Firstly, retrieval results can be filtered by using the structured information. Secondly, a summary can be generated for each DRed node to clearly inform users about its type, status, and other features. Thirdly, since the information in a single DRed node is not enough, a group of nodes can be found and returned as the retrieval result. Fourthly, complex queries can be formed and used to better express users’ information needs, and thus find better results. The methods proposed are shown to be useful and the algorithms used are feasible. As any design knowledge captured is in essence structured, the methods developed can be generalised and utilised for the retrieval of other structured design knowledge. In our future work, we will study how to enable users to formulate complex queries and how to understand their contexts of working so that relevant design rationale records can be recommended automatically.

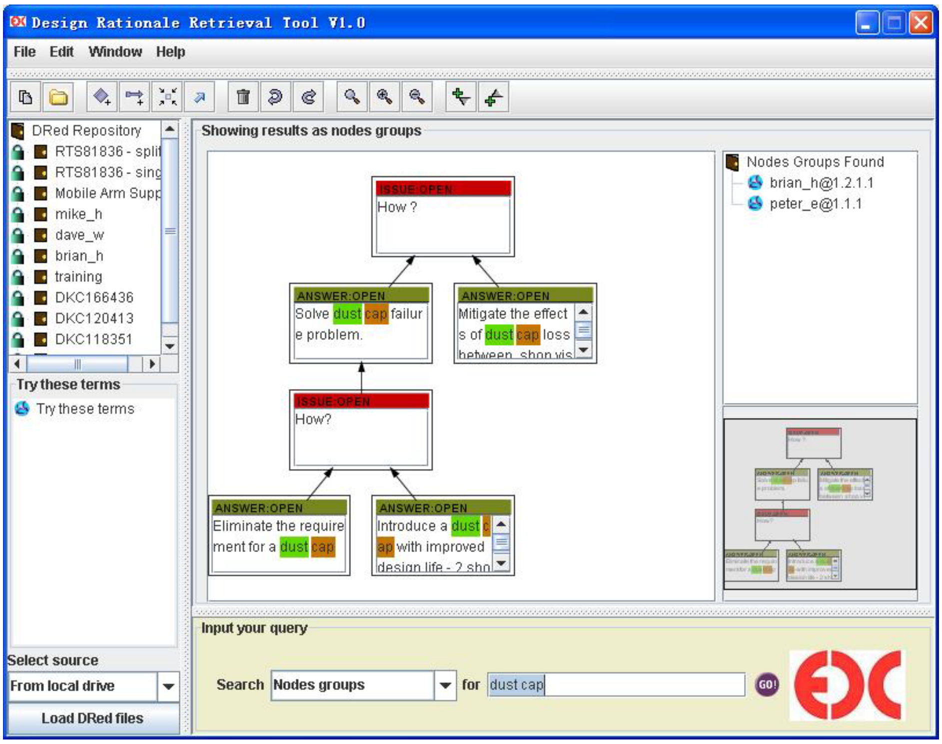

Figure 3. Nodes groups found in response to queries

REFERENCES

[1] Regli W.C., Hu X., Atwood M. and Sun W. A survey of design rationale systems: approaches, representations, capture and retrieval. *Engineering with Computers*, 2000, 16(3-4), 269-286.

[2] Jarczyk A. P. J., Loffler P., Shipman III F.M. Design rationale for software engineering: a survey. In *Twenty-Fifth Hawaii International Conference on System Sciences*, 1992.

[3] Giess M. D., Goh Y. M., Ding L. and McMahon C. A. Improved product, process, and rationale representation and information organisation to support design learning. In *International Conference on Engineering Design, ICED'07*, Paris, August 2007.

[4] Arora V., Greer E.J., Tremblay P. A framework for capturing design rationale using granularity hierarchies. In *Fifth International Workshop on Computer-Aided Software Engineering*, 1992.

[5] Lee J. Design rationale systems: understanding the issues. *IEEE Expert*, 1997, 12(3), 78-85.

[6] Klein M. Capturing design rationale in concurrent engineering teams. *IEEE Computer*, 1993, 26(1), 39-47.

[7] Burgess K.C., Conklin E.J. Crisfield M.A. Report on a development project use of an issue-based information system. In *International Conference on Computer Supported Cooperative Work, CSCW 90*, 1990.

[8] Burge J.E., Brown D.C. SEURAT: integrated rationale management. In *International Conference on Software Engineering, ICSE'08*, Leipzig, Germany, 2008.

[9] Bracewell R. H. and Wallace K. M. A tool for capturing design rationale. In *International Conference on Engineering Design, ICED'03*, Stockholm, August 2003, pp.185-186.

[10] Ahmed S., Wallace K.M. Understanding the knowledge needs of novice designers in the aerospace industry. *Design Studies*, 2004. 25(2), 155-173.

[11] McMahon C., Lowe A., Culley S. Knowledge management in engineering design: personalization and codification. *Journal of Engineering Design*, 2004. 15(4), 307-325.

[12] Szykman S., Sriram R.D., Regli W.C. The role of knowledge in next-generation product development systems. *Journal of Computing and Information Science in Engineering*, 2001.

1(1), 3-11.
[13] Susca L., Mandorli F., Rizzi C., Cugini U. Racing car design using knowledge aided engineering. *Journal of Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 2000. 14(3), 235-249.
[14] Marx W.J., Mavris D.N., Schrage D.P. A knowledge-based system integrated with numerical analysis tools for aircraft life-cycle design. *Journal of Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 1998. 12(3), 211-229.
[15] Skarka W. Application of MOKA methodology in generative model creation using CATIA. *Journal of Engineering Applications of Artificial Intelligence*, 2007. 20(X), 677-690.
[16] Brazier F.M.T., Van Langen P.H.G., Treur J. A compositional approach to modelling design rationale. *Journal of Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 1997. 11(2), 125-139.
[17] Ganeshan R., Garrett J., Finger S. A framework for representing design intent. *Design Studies*, 1994. 15(1), 59-84.
[18] Shipman F.M., McCall R.J. Integrating different perspectives on design rationale: Supporting the emergence of design rationale from design communication. *Journal of Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 1997. 11(2), 141-154.
[19] Charlton C.T. The retrieval of mechanical design information. *PhD Thesis*, Department of Engineering, University of Cambridge, 1998.
[20] Giess M.D., Goh Y.M., Ding L., and McMahon C.A. Improved product, process and rationale representation and information organisation to support design learning. In *International Conference on Engineering Design, ICED'07*, Paris, August 2007.
[21] Huet G., Curley S.J., McMahon C.A., Fortin C., Sellini F. Communication, information and knowledge processes observed during engineering design reviews. In *International Conference on Engineering Design, ICED'07*, Paris, August 2007.
[22] Kim S., Bracewell R.H., Wallace K.M. A framework for design rationale retrieval. In *International Conference on Engineering Design, ICED'05*, Melbourne, August 2005.
[23] Kim S., Bracewell R.H., Wallace K.M. Improving design reuse using context. In *International Conference on Engineering Design, ICED'07*, Paris, August 2007.
[24] Wang H., Johnson, A., Bracewell R. Supporting design rationale retrieval for design knowledge re-use. In *International Conference on Engineering Design, ICED'09*, Stanford, USA, August 2009.
[25] Bracewell R., Wallace K., Moss M., Knott D. Capturing design rationale. *Journal of Computer-aided Design*, 2009. 41(3), 173-186.
[26] Bracewell R.H., Gourtovaia M., Wallace K.M., Clarkson P.J. Extending design rationale to capture an integrated design information space. In *International Conference on Engineering Design, ICED'07*, Paris, August 2007.

Contact: Hongwei Wang
University of Cambridge
Engineering Design Centre, Engineering Department
Cambridge, CB2 1PZ
UK
Tel: Int +44 1223 748568
Fax: Int +44 1223 332804
Email: hw308@cam.ac.uk
URL: http://www-edc.eng.cam.ac.uk/people/hw308.html

Hongwei Wang's research interests are mainly on the methods and tools to support engineering design processes, in particular how to manage design rationale to facilitate knowledge re-use and how to evaluate design concepts using simulation techniques in a distributed environment.