From Discourse Analysis to Answering Design Questions

Sanghee Kim, Rob Bracewell and Ken Wallace

Engineering Design Centre, Department of Engineering,
University of Cambridge, UK
{shk32,rhb24,kmw}@eng.cam.ac.uk

Abstract. This paper discusses an approach of modelling design rational expressed in natural language sentences into a discourse model. The discourse model is used to classify captured rationale into discourse categories by taking into account semantic and pragmatic relationships between two design elements. It is expected that accessibility and reusability of captured rationale is improved since retrieval is supported within discourse contexts. A small dataset was collected to test whether selected discourse relations are extractable and whether a machine learning algorithm can generate rules under which appropriate relations can be automatically marked.

1 Introduction

The reuse of design knowledge depends on the successful retrieval of the required knowledge. Given a large number of information sources, it is a challenge for designers to search effectively for the answers they need. It is suggested that an efficient knowledge management tool needs to support the information needs of designers by answering their questions with concise responses and providing them with easy navigation for further information. Well defined semantics enable such knowledge seeking processes to be automated by making the information to be interpretable both by humans and machines.

Argumentation-based Design Rationale (DR) tools (e.g. DRed [2], Issue-Based Information System (IBIS) [9], or Questions, Options and Criteria (QOC) [15]) model the information space considered by a user when dealing with a specific design task. These tools are able to provide with not only factual information (e.g. definitions) but also explanations about how certain decisions were made. Finding optimum solutions for given problems involves considering alternatives and justifying the selected ones. Complex problems are decomposed into details and solutions are often dependent on each other. Alternatives can be compared for differences and similarities and the elaborations of problems for explaining how related problems and solutions are explored are common. The relationship between two typed elements (e.g. issues, answers, arguments) linked by arrows conveys a designer's problem solving preferences. Such a dependency within a design context is critical for structuring captured information for better accessibility and reusability.

Since these tools represent rationale with typed elements, the rationale can be retrieved by a question answering (QA) framework, where the answers to questions are identified by following the direct links between them. As such, a hypermedia-based question answering system can be implemented with which a user can browse to find out what kinds of issues were discussed, why certain answers were chosen, and who made such decisions. Although it is easy to annotate question-answer pairs, there are three challenges in retrieving answers against users' questions. First, it is feasible that users' questions and the existing questions are not exactly matched; they might differ in terms of sentence structures or semantic meanings. Secondly, since specific answers can be elaborated by creating further elements, more appropriate answers can be identified by following links rather than the answers already associated. Thirdly, whereas [14] argued that although typed elements are a sufficient means of indexing, users' requests are not necessarily restrained to the questions (e.g. appropriate answers are retrieved from arguments), so further analysis of other types of elements rather than answers is necessary.

Semiformal, argumentation-base design rationale (DR) tools can make use of typed elements for structuring captured rationale. For example, users can retrieve all the issues related to *turbine engine* and subsequently request the corresponding answers. It means that users can query over the organised structures of DRed graphs such that an element is retrieved although it does not match with the terms in the query submitted. One problem is that in general users of argumentation-based DR tools do not take into account how the represented issues would be accessed by others or what design decisions should be known to others [15]. In comparison to DR tools which focus on documenting critical decisions for future reference, background information is often missing. Therefore extracting implicitly conveyed context is important for better accessibility.

The argument above suggests that many answers to rationale questions are not retrieved simply by recognising similar questions or by following links to associated answers. As reported in [5], the explanations why certain options are chosen include better comprehensibility with the descriptions of design constraints or functionalities considered which are distributed across different elements. As such, in order to provide valid explanations, it is necessary to identify dependency between design knowledge elements and to integrate them when inferring appropriate answers. Finding such dependency can be supported by a discourse model that extracts semantic and pragmatic relations between two texts. In this paper, an approach of modelling design rationale into a discourse model for the purpose of better accessibility and reusability is discussed. The discourse model is expected to support sophisticated question answering through discourse annotations (e.g. difference, evidence), which might be infeasible with a keyword matching. For an example of argumentation, semiformal DR tool, DRed is chosen, and more details are shown in [2].

This paper is organised as follows: in section 2, reviews of design rationale retrieval and discourse analysis are presented; section 3 describes a proposed discourse model and a machine learning algorithm used for generating rules un-

der which appropriate relations can be automatically marked. An experimental result is presented in section 4 followed by conclusions and future work.

2 Related Work

Argumentation-based semiformal design rational (DR) tools use graphs in which design elements (i.e. nodes) are differentiated by colors and/or notations. There are at least three types of user interfaces in retrieving information from those DR tools. Users navigate the indexed elements in order to identify what kinds of issues were discussed or the reasons why certain answers were accepted [4, 6, 15]. Users add their ideas to the existing ones and new nodes are created in relation to the existing ones. A difficulty arises when the number of nodes is increased such that the information space which users navigate becomes too large. In addition, since the users of those tools do not take into account how the represented issues would be accessed by others or what design decisions should be known to others, it might be difficult to know which links to follow in order to search for required information [14, 15]. [15] conducted user studies of analysing cognitive issues caused by retrieving information from Questions, Options and Criteria (QOC) representation. Since the QOC is more concerned with answering why certain decisions were made, the content structures are mainly determined by the narrative aspects of the issues explored. As such it was difficult for users to comprehend why a specific issue was located under an issue which seems irrelevant to them. It was claimed that the coherence of a DR structure was critical for easy navigation, so that grouping elements across different graphs according to their generic task classifications could be useful.

A question-based retrieval is an alternative, and is more efficient than the navigation of rationale structures [12]. [5] analysed the questions asked by designers or derived from various design statements and classified them with various criteria (i.e. search strategies, answer types). One of the findings indicated that the questions related to rationale often needed to be inferred by considering dependencies among design elements (e.g. design decisions, artefact structures or expected behaviors). Automatic suggestions of necessary constraints or detections of design errors can minimise the passiveness of these two types of retrievals. CRACK actively notifies not-satisfying design solutions by monitoring the activities of designers and compares them with pre-defined design principles [4]. In PHIDIAS, users not only take suggested critics but also can argue with such critics and provide rationale why these are inaccurate for them [14].

A rhetorical structure theory (RST) is a well-known discourse theory and labels the two texts as nuclei (i.e. more essential to the text and independent on the satellites) and satellites [10]. Rhetorical relations are related to both the semantics of texts and the communicative purposes intended. A discourse model was used to assess student essays with respect to their writing skills, to analyse scientific papers for summarisations, and to generate personalised texts according to reading abilities of users [3, 16, 17]. In labelling discourse relations,

cue phrases can be used, for example, by detecting the word 'but', a 'contrast' relation between two adjacent texts is identified [8].

3 A Discourse Model for Design Rationale

Elements in DRed are a collection of natural language sentences and references to previously mentioned information in other elements is common. A semantic relation in which one element provides necessary information for interpreting another element suggests potential dependency relationships among the elements. A discourse model can be used for mining such semantic relations. A discourse analysis takes as inputs texts or dialogs and assigns discourse labels (e.g. elaboration, cause) to discourse units by taking into account semantic, pragmatic or syntactic features. Each unit is related to the previous one by a coherence discourse relation that determines its role to realise a speaker's or writer's intentions. Fig. 1 shows an example of a discourse model for a situation where a user considers various transportation methods for getting to a station. For example, given an issue of *'how to get to the station'*, one of the options considered is *'use the car'* which has two motivations (cheapest, quickest). With this model, users can ask sophisticated queries like *'how to check if there is petrol in the tank'* or *'what is the consequence of battery failure'*. In addition, captured rationale can be retrieved according to discourse relations, e.g. *'find all causes of car_brake_down'* or *'what solutions were considered'*.

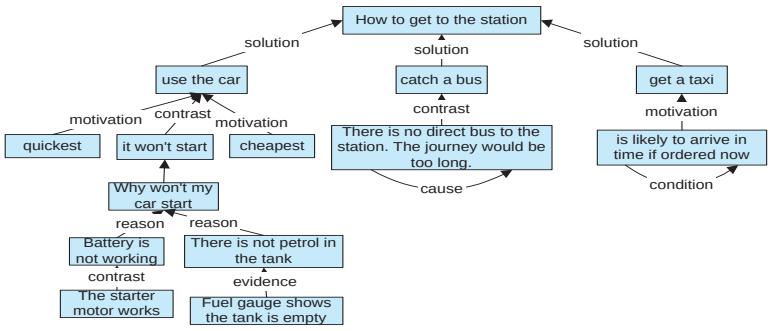

Fig. 1. An example of a discourse model

3.1 Discourse Relations

In order to create a discourse model suitable for DRed graphs, decisions about the number of the relations, relation types, and the methods of automatically annotating such relations have to be made. A set of relation types is determined in relation to the questions to be asked by users when they search or to the analysis of commonly used discourse labels in DRed graphs. Each relation is defined

according to its role of contributing to the communication purpose of associated design elements. Potential user interests in browsing rationale according to discourse types are also considered. Since too many relation types can increase disagreements among users (i.e. for some users, it may be hard to distinguish 'contrast' from 'concession'), and too few relations may be insufficient for capturing different types of discourses, it is critical to determine the right numbers of relations. As such, we cluster relations into seven categories, i.e. *elaboration, cause, problem-solution, motivation, question-answering, walk-around, and contrast*. An elaboration holds when one of two elements contains more details of the another. A cause includes result, purpose, consequence, and evidence relations. A problem-solution recognises solutions for given problems. A motivation provides positive opinions on ideas or solutions. A question-answering provides answers for given questions. A contrast favours one idea or solution but not the other. A walk-around is a question or statement which diagnoses given problems. Discourses like 'sequence', 'joint', 'conjunction', or related to 'temporal' or 'spatial' relations are not considered since these are less relevant to explicating design rationale and these are not commonly identifiable between two texts rather than between two clauses.

3.2 Progol: A Machine Learning Algorithm

Inducing rules from given examples can be supported by the inductive logic programming technique (ILP). In contrast with other learning methods, such as Decision Trees, using computational logic as the representational mechanisms means that ILP can learn more complex, structured, or recursive descriptions and generate the outputs in first-order logic. Progol is one of the ILP systems and selects one positive example, constructs the most specific clause and this becomes a search space for the hypotheses [11]. The use of Progol to analyse texts has been attempted in [1, 7]. The target clause to be learned is prediction (A,B), where B is the predicted relation type with which the link A is to be associated, e.g. prediction(link1, 'contrast'). Each link is represented with clauses as described in Table 1. For example, 'has_first_ele(+link,-element)' specifies a source element with which the link is associated and 'has_word(+element,-word)' specifies a word occurred in the element. Apple Pie parser[13] is used for syntactic taggings.

Table 1. Clauses used by Progol

has_first_ele(+link,-element).	has_second_ele(+link,-element).
has_modal_verb(+element,#modal).	has_first_word(+element,+word).
has_word(+element,-word).	has_verb(+link,#verb).
has_subject_word(+element,#word).	has_object_word(+element,#word).
has_prev_postag(+element,+postag,#postag).	has_prev_discourse(+link,#discourse).
has_subject_pos(+element,#postag).	has_object_pos(+element,#postag).
has_postag(+element,+word,#postag).	has_question_mark(+element,#mark).

4 Evaluation

It is hypothesized that texts in argumentation-based, semiformal DR tools can be modelled by discourse relations and these relations are consistently extractable across DR graphs. It is also hypothesized that the rules specifying under which conditions discourses are automatically marked can be derived from tagged examples. Precision and recall were used for measuring the performance of Progol. Precision is the proportion of the correctly predicted discourses by the Progol to the number of discourses predicted. Recall is the proportion of the number of the correctly predicted discourses to the total number of correct discourses. A total of 264 links (468 sentences) was collected from the seven graphs which were based on the design documents used by an engineering company. A total of 250 discourses was manually annotated. It was observed that cue phrases (i.e. [8]) were rarely used. Only 15% of 250 relations were identifiable by matching with the cue phrases and most common relation was *cause*.

10-fold cross-validation was used for dividing the dataset into training and testing examples. The following shows an example of generated rules for a 'cause' discourse. *prediction(A,cause) :- has_first_ele(A,B), has_subject_word(B,what), has_object_pos(B,'DT'), has_prev_discourse(A,'walk-around').* The rule defines that if the first element had the word 'what' in a subject and a word pos-tagged as 'determiner' in an object and if an antecedent link had a 'walk-around' discourse, then the link is predicted as related to 'cause'. The 'walk-around' was identified as commonly used with relation to 'cause' discourse. It was observed that when users faced with design problems, they often investigated what were the causes of the problems before searching for corresponding solutions. This observation is consistent with the fact that the DRed has been used by users who are familiar with problem solving procedures. On average, 83% precision and 74% recall were obtained, i.e., 'walk-around'(0.9, 0.79), 'cause' (0.78, 0.4), 'contrast' (0.88, 0.8), 'elaboration' (0.91, 0.67), 'motivation' (0.68, 0.74), 'question-answering' (0.83, 0.87), and 'problem-solution' (0.85, 0.89). 74% of testing relations were identified suggesting that the rules generated by Progol were more reliable than relying on the cue phrases.

5 Conclusion and Future Work

This paper has discussed an approach of modelling design rationale into a discourse model and a preliminary experiment with a small dataset was presented. Progol was able to generate a set of rules specifying under which conditions appropriate discourse labels could be extracted for untagged design elements. The dataset was manually annotated by one person and since the decisions on the use of labels depend heavily on the annotators preferences, it is planned to re-examine the generated rules with new datasets. A user interface based on the proposed discourse model will be developed to evaluate how the discourse model will be used by users when searching for required information. It is also planned to investigate how the discourse structures could be used for deriving design patterns in order to actively inform users of the existence of needed information. In

addition, the investigation of how the discourse structures could access graphs in terms of how well the issues and solutions are explored so that individual problem solving processes can be compared and evaluated will be carried out.

6 Acknowledgements

This work was funded by the University Technology Partnership for Design, which is a collaboration between Rolls-Royce, BAE SYSTEMS and the Universities of Cambridge, Sheffield and Southampton.

References

1. J. S. Aitken. Learning information extraction rules: An inductive logic programming approach. In *European Conf. on Artificial Intelligence*, pages 335–359, 2002.
2. R.H. Bracewell and K.M. Wallace. A tool for capturing design rationale. In *14th Int. Conf. on Engineering Design*, pages 185–186, 2003.
3. J. Burstein, D. Marcu, and K. Knight. Finding the write stuff: Automatic identification of discourse structure in student essays. *IEEE Intelligent Systems*, Jan/Feb:32–39, 2003.
4. G. Fischer, R. McCall, and A. Morch. Design environments for constructive and argumentative design. In *CHI 89*, pages 269–275, 1989.
5. T. Gruber and D. M. Russell. Derivation and use of design rationale information as expressed by designers. Technical Report KSL 92-64, Stanford University, 1992.
6. S. Isenmann and W. Reuter. Ibis - a convincing concept but a lousy instrument? *Symposium on Designing Interactive Systems*, pages 163–172, 1997.
7. S. Kim, P. Lewis, and K. Martinez. The impact of enriched linguistic annotation on the performance of extracting relation triples. In *Computational Linguistics and Intelligent Text Processing*, pages 547–558, 2004.
8. A. Knott and R. Dale. Using linguistic phenomena to motivate a set of coherence relations. *Discourse Processes*, 18(1):35–62, 1995.
9. W. Kunz and W. J. Rittel. Issues as elements of information systems. Technical Report 131, University of California, Berkeley, 1970.
10. W. C. Mann and S. A. Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281, 1988.
11. S. Muggleton. Inverse entailment and progol. *New Generation Computing*, 13:245–286, 1995.
12. W. C. Regli, X. Hu, M. Atwood, and W. Sun. A survey of design rationale systems: Approaches, representation, capture and retrieval. *Engineering with Computers*, 16:209–235, 2000.
13. S. Sekine and R. Grishman. A corpus-based probabilistic grammar with only two non-terminals. In *1st Int. Workshop on Multimedia annotation*, 2001.
14. F. Shipman and R. McCall. Integrating different perspectives on design rationale. *AIEDAM*, 11(2):141–154, 1997.
15. S. Shum. Qoc design rationale retrieval: A cognitive task analysis and design implications. Technical Report EPC-1993-105, Rank Xerox Ltd., 1993.
16. S. Teufel. Task-based evaluation of summary quality: Describing relationships between scientific papers. In *Workshop Automatic Summarization, NAACL*, 2001.
17. S. Williams and E. Reiter. A corpus analysis of discourse relations for natural language generation. In *Corpus Linguistics*, 2003.