Text Mining for Meeting Transcript Analysis to Extract Key Decision Elements

Caroline Chibelushi and Mike Thelwall

***Abstract*— The frequent but unfortunate need to rework software development projects may often be caused by inappropriate decision making. The first step in addressing this issue is to explore decision making processes and to extract the tangible elements of decision making within meetings. This paper explores the hypothesis that text mining techniques can be used to extract the elements of decision making from software development project meetings that can ultimately be used as a facility to develop a decision management system. Theories of discourse, lexical chaining and cohesion are presented and used as the basis for the analysis of meeting transcripts. Information retrieval and data mining methods are also used. To assess the performance of the algorithm the C99, and TextTiling algorithms are used as comparators. The evaluation results show that our method is able to identify and extract the needs and actions of decision making with a high recall of 85 – 95% at a precision of 54-68%.**

***Index Terms*— Decision making process, Rework, Software development projects, Text mining.**

I. Introduction

Many software development projects have been unsuccessful partially as a result of communication failures between decision makers and incorrect or inappropriate decision making [1]. Rework involves altering, revising or restarting certain project activities because the previous work was incorrect, incomplete or inconsistent. It is a major cost factor in system development projects and accounts for over 50% of additional effort and substantial costs, particularly for large projects [2, 3]. This research considers decision management systems as a means of controlling rework and failure in system developments projects.

Rework can be associated with incorrect or inappropriate decision making. However, decisions could be 'hidden' and difficult to identify, and the evidence of their existence is related to the issues discussed, the meeting participants' needs, the actions taken to satisfy the participants' needs, and the agents who will perform the chosen actions. Such detailed information is often not found in meeting minutes, and may result in unmanaged decisions. This research argues that by investigating recorded and transcribed meeting conversations, it is possible to identify and extract evidence of meeting decision making processes, referred to herein as the elements of decision making. By understanding the relationship between these elements, managers will be able to better detect incorrect decisions and communication failures, and to understand their impact on rework.

Many software development projects become successful and achieve quality after undergoing iterative processes [4], e.g., specification, design, coding and testing processes. However, in order to correctly perform these iterative processes, the decision makers have to communicate and relate their decisions, understand the impact of each decision made on each iterative process and the risks involved in making each decision. For example, a decision maker can decide on *design issues* without relating them to the decisions made on the *testing issues*, and without keeping detailed and accurate documentation. In addition, software development project meetings may contain decisions that lack an action, or decisions that contain an action but lack an agent to perform that action. As a result, costly rework in software development projects occurs, causing budget overruns.

An example of a catastrophic failure that occurred as a result of the above mentioned factors is the 4th June 1996 explosion of the Ariane 5 rocket whilst lifting-off on its maiden flight [5, 6]. Rework also occurs in many organisational systems, especially in the public sector [7]. For example, rework has recently occurred in major projects shown in Table I.

Manuscript received November 13, 2008. This research project was supported by the UK Engineering and Physical Sciences Research Council (EPSRC), grant GR/R12176/01.

C.Chibelushi and M. Thelwall are with the School of Computing and IT, University of Wolverhampton, City Campus - South, Wulfrana Street, Wolverhampton, WV1 1SB, UK. Email{c.chibelushi, m.thelwall}@wlv.ac.uk

Table I: Recent software failures in the UK public sector.

Software failure	Responsible software company	Failure date	Description	Reference
Educational maintenance allowance (EMA) system	Liberata	Sept 2008	Students were unable to receive their study grants due to the failure of an online application system and telephone helpline	www.kablenet.com
Standard Assessment Tasks (SATs) - marking system	ETS Europe	May 2008	School children's exam results were delayed due to Technical and operational problems associated with the systems and the management of information	www.bbc.co.uk
British Airways Terminal 5 failure	British Airports Authority (BAA) and British Airways (BA)	May 2008	A disastrous launch of terminal 5 was in part due to lack of software testing	www.computing.co.uk
A system to ensure that patient records are available electronically to all clinicians- National Health Service (NHS)	iSoft	August 2006 - ongoing	A £12.7bn NHS project was supposed to be implemented in 2004. But it is 4 years delayed as iSoft has financial difficulties	www.gurdian.co.uk www.telegraph.co.uk ww.computerweekly.co.uk
Passport office computer failure	Siemens	1999	The failure that led to a backlog of more than 500,000 unprocessed passport applications, costing trhe tax payer approximately £12m.	http://news.bbc.co.uk
National Insurance Recording System II (NIRS II) failed	Andersen Consulting	September 1998	Allowance and some incapacity benefits are being paid "blind" after the National Insurance Recording System II (NIRS II) failed, erasing personal records.	http://news.bbc.co.uk/1/hi/uk/168277.stm

Another factor that may cause software development complications could be *incorrect analyses* made by the requirements engineers. In the case of Ariane 5 for example, this factor led to *incorrect decisions* to re-use the Ariane 4 alignment code that was not needed by Ariane 5 after its lift-off. The code remained operational in Ariane 5 without satisfying any specific requirement. The Ariane 5 IRS (Inertial Reference System) shut down 37 seconds after launch as because some Ariane 4 values overflowed in a computation that was not required by Ariane 5. As random signals were being sent to the launcher engines, control was lost and the launcher self-destructed. This would have been prevented if there was communication between decision makers and engineers, as well as detailed and accurate documentation [8] on decisions made before launching Ariane 4 that could have been accessed by the integrators and users. If such documentation existed it would have been easier to be aware of the problems that may arise when adequate validation, verification, testing and review were not conducted on the system.

Despite the problem of rework being well recognised, there has been little research focused on the analysis of meeting transcripts with the intention to reduce rework and failure in software development projects. Instead, the problem of rework has been addressed through technical tools and techniques such as software language specifications or requirements management tools. More precisely, the problem of rework has been addressed through changing requirements. In addition, most tools support the process of carrying out rework, but do not help eliminate rework caused by inappropriate decision making. The first step in solving this problem is to explore decision making processes and propose a model for the tangible elements of decision making.

II. The Corpus

The research project used 17 transcripts recorded from three diverse meeting environments: industrial, organizational and educational, each involving a multi-party conversation with an accurate and unedited record of the meetings, corresponding speakers and no pre-set agendas. The meeting transcripts varied in size, ranging

from 2,479 to 25,670 words, posed various complexities due to their informal style, their lack of structure, their argumentative nature, and the usage of common colloquial words. The transcripts contain incomplete sentences, sentences related to social chatting, interruptions, and references by participants made to visual contexts. The total corpus with a total of 247,238 words was used to identify an appropriate model and to illustrate the algorithm proposed for the analysis of transcripts. However, in this paper, only a transcript with a total of 25,670 words is analysed. This was the longest and most argumentative meeting, hence its selection.

III. The ANIA Model

There have been various views on the number of distinct processes followed in organisational decision making, ranging from three to five main phases [9-12]. These were then developed by many other researchers resulting in different iterative models (example, [13-16]). Another 3 models of decision making which are commonly used in many organisations and share similar phases to the above iterative models include the rational [15, 17], garbage can [18], and political model [19]. The three models represent current thinking with regard to formal organisational decision making and hence are preferred by many organisations.

The existing models are simplified as some (e.g. the rational model) are impossible to implement because they require comprehensive knowledge of every facet of the problem [20]. However, the models are iterative and share some of the same decision making components [21].

Although it is obvious that the aims of the above models are to assist decision makers to make appropriate decisions, these models do not explicitly identify and relate the key features of a decision. The obvious requirement that can lead us to fully understand decision making or discriminate between different ways of managing decisions and subsequently rework is to identify the necessary features of decision making within the decision management domain; this has led this research to propose the ANIA model for decision making process.

The ANIA model is developed using the shared components of the above mentioned models. ANIA represents ***A**gents, **N**eeds, **I**ssues and **A**ctions* within decision making processes. In order to be able to develop the ANIA model, the research introduces the concept of '*decision making elements'* which are generated from the decision making processes.

The model perceives meetings as activities conducted because one or more '*agents'* have '*needs*' for a particular *issue* or a *topic*, and each need can be fulfilled by one or more '*actions'*. The *agents*, *needs*, *issues* and *actions* are referred to in this research as the elements of decision making.

Each element can be part of one or more decisions. The identification of these elements and the recording of them in the decision management system will enable decision makers to identify the actions, the information that led to them, the reason why the actions were taken and the result of the actions. Decision makers will thus be able to understand the decisions made by other decision makers, and why they were made.

IV. Text Analysis Theories Pertinent to the Analysis of Transcripts

Grosz and Sidner [22] have proposed a theory for discourse structure to understand and determine the relationships between sets of words uttered by different speakers across dialogue turns. Any attempt to automate the process of identifying Grosz and Sidner's discourse structure requires a method of identifying linguistic segments in text. In order to achieve this, Morris *et al.* [23] extended Grosz and Sidner's theory to implement a lexical chaining technique.

Discourse is made up of functional words and content words. Examples of functional words are: *'the', 'is', 'a', and 'for'*. These words are likely to be used in the text about any subject. Examples of content words are: '*software', 'application',* and *'date'*. These are mostly represented in the text as nouns. Researchers such as Hasan [24], Hearst [25], and Reynar [26] have observed that nouns and noun phrases, sometimes called content phrases, are mostly used in human language to convey the information in a text. Lexical cohesion is a result of identifying and relating content phrases that contribute to the continuity of lexical meaning, hence identifying issues of conversation in meetings. For this reason, finding text structure involves finding units of text (content phrases) that are about the same thing [23]. When these units are semantically clustered, they are referred to as *lexical chains.*

Morris and Hirst [23] hypothesised that in order to be able to capture discourse structure, they needed to divide the text into cohesive segments. They employed Halliday and Hasan's [24] theory to analyse the cohesiveness of a text segment. The first step in Morris and Hirst's [23] algorithm was to link sequences of related words from a document to form lexical chains. They believed that each segment would be represented by the span of a lexical chain in the text.

Halliday and Hasan [24] identify five (not always distinct) cohesive relations that contribute coherence to a document: conjunction, substitution, reference, ellipsis and lexical cohesion [27]. Lexical cohesion is a linguistic device for investigating the discourse structure of texts, and lexical chains have been found to be an

adequate means of exposing this structure. In the usual case where a discourse contains a set of related terms, Morris *et al.* [23] claim that cohesion is a useful sign of coherence in a text, especially since the identification of coherence itself is not computationally feasible at present. Stairmand [28] supports this claim by emphasizing that although cohesion fails to account for grammatical structure (i.e. readability) in contrast to coherence, cohesion can still account for the organisation of meaning in a text, and so, by implication, its presence corresponds to some form of structure in the text. This research uses lexical cohesion (as in Wordnet) to develop its lexical chains.

V. TEXT MINING

Text mining allows the extraction of significant features from unstructured textual data such a meeting transcript. This research adapts the CRISP-DM (**CR**oss-**I**ndustry **S**tandard **P**rocess for **D**ata **M**ining) methodology for text mining by presenting a new methodology called the **Dec**ision **M**anagement using **T**ext **M**ining (DecM-Text Mining). This methodology consists of three main phases as shown in Fig.1.

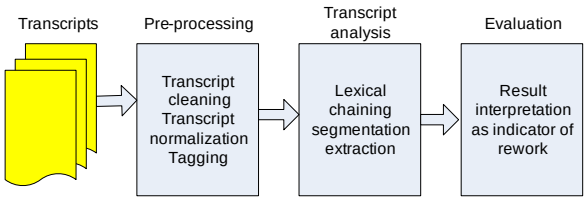

Fig.1. DecM-Text mining

The three stages of DecM-Text mining are as follows.

A. *Pre-processing*

This stage prepares the textual data to allow for more precise results and faster processing. It includes three steps: i) *transcript cleaning* involves removing signs and characters which can be irrelevant for text mining, ii) *text normalisation* involves tokenization (dividing transcripts into utterances), case folding (removing differences between capital and lower case words), identification of compound words and removal of stop words, and iii) part-of-speech tagging is done using the online tool Wmatrix [29].

B. *Transcript analysis*

The DecM-Text mining transcript analysis phase comprises three main tasks: lexical chaining, segmentation, and extraction.

Lexical chaining Lexical cohesion is used as a linguistic device for investigating the discourse structure of texts, and lexical chains have been found to be an adequate means of exposing this structure. Morris *et al.* [23] define *lexical chain* as a term used to identify sequences of cohesive words in a text, where lexical cohesive relationships between words are established using an auxiliary knowledge source such as a dictionary or a thesaurus.

Lexical chaining implements feature clustering, i.e. grouping together related features in a document or across document collections. In contrast to generic techniques like hierarchical clustering [30-32], k-means [33] and two dimensional clustering in Self Organizing Maps [34, 35], lexical chaining is less complex and takes into account the context of the document. In lexical chaining, words are clustered depending on the semantic relationships (synonym, hypernym, meronym etc.) between them, and hence the analysis is contextual.

Segmentation This is used to divide transcripts into their topical boundaries and includes temporary segmentation based on the cosine similarity measure, the application of lexical chains (which involves content) to identify topic boundaries, and final boundary identification refines further the new segments by searching for speech cue phrases to confirm or cast doubt upon the topic boundaries.

Extraction The aim of the DecM-Text mining Extractor (DTE) is to extract topics or issues, agents, actions, and needs, from meeting transcripts. To extract topics, Katz's theory [36] is implemented. In this, the extraction of agents, actions, and needs relies on linguistic pattern recognition methods. The extraction process depends on the appearance of elements related to decision making within a segment.

VI. EVALUATION

Evaluation is performed in two stages; segmentation and extraction. Segmentation was evaluated by comparing the **D**ecM-**T**ext Mining **T**ranscript **S**egmentation (DTTS) against the two standard methods: Textilling and C99. Three types of evaluation metrics were used, P_k [37], P_k', and *WindowDiff* [38]. Each metric measures values ranging from 0 to 1 inclusive. These measures aim to calculate the average amount of error from each approach, and so the approach with the lowest score in each metric is the best performing algorithm. The results are shown as shown in Fig. 2.

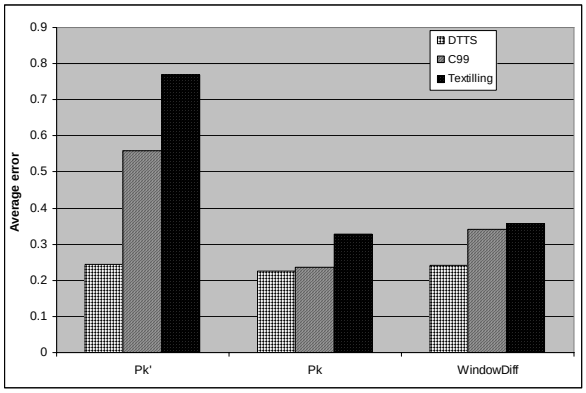

Fig. 2. DTTS, C99 and Textilling segmentation accuracy

Fig. 2 show that DTTS has considerably fewer errors than TextTiling and C99. The difference between the P_k and P_k' values of the TextTiling and C99 approaches is relatively large. The P_k' values in C99 and TextTiling are approximately twice as large as P_k, which means that these approaches suffer from false positives.

Since DTTS topic boundaries are relatively accurate, this implies that the discussed issues are also separated accurately. Similarly, because each utterance has the name of the corresponding speaker, when the utterance with a need or action is extracted, it will automatically have corresponding speaker (agent) associated with it. For this reason, this research focused on the evaluation of the needs and actions.

The extracted needs and actions were evaluated using precision and recall as shown in Fig. 3.

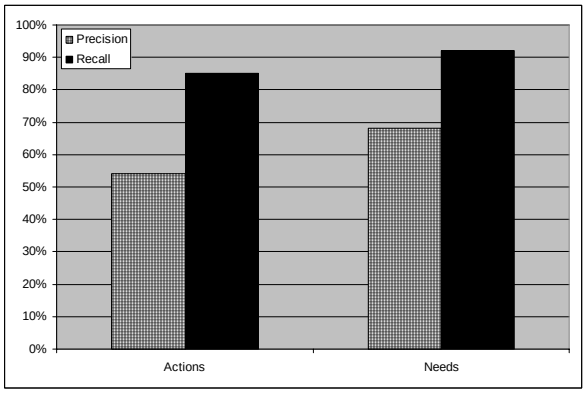

Fig. 3. Recall and precision of the extracted 'actions' and 'needs' from the corpus

The goal of the DTE approach is to extract as many needs and actions as possible from meeting transcripts. The linguistic patterns used include actions or needs with repetition, faulty tagged phrases and interruption – incomplete actions, or needs and actions which are not portraying any meaning without some cognitive knowledge. As a consequence, precision suffers due to extracting needs or actions which are not useful. However, in the extraction of needs, the system's performance is satisfying because the recall of two transcripts measured 100%. Apart from interruption and repetition, there was little irrelevant material. Most *needs* linguistic patterns do not have a wide variation in their appearances. Test results used to calculate recall and precision were identified by a group of 6 Natural language processing researchers.

VII. Conclusions

This paper describes a new text mining approach to extract key information regarding decisions from software development meeting transcripts. The approach combine aspects from many disciplines: decision theories, linguistics, natural language processing, information retrieval, text mining and statistics. These are incorporated in text mining approach called **Dec**ision **M**anagement using **T**ext mining, or DecM-Text mining. This was applied to software development project meeting transcripts to identify and extract the relevant elements of the decision making process. DecM-Text mining achieves this by providing the contextual analysis that is necessary to understand thematic aspects within software development project meeting conversations. The approach is an adaptation of the CRISP-DM methodology for mining unstructured text by combining algorithms from different fields.

The results of both the segmentation and extraction tasks were obtained and evaluated. Three metrics were applied to two different lexical cohesion based segmentation algorithms: C99 [39] and TextTiling [25]. The results show that DTTS outperforms the C99 and TextTiling algorithms. The results of the Dec-Text mining extraction (DTE) task were also evaluated using precision and recall measures. The average measure for recall on the extracted 'needs' is 92%, and its precision is 68%, while the measure for recall on the extracted 'actions' is 85%, while its precision is 54%. This suggests that the approach correctly detects most of the cohesive relationships necessary to identify the topic regions and shifts within meeting transcripts, and has extracted the relevant related elements of decision making found in each region.

Text mining is still evolving and this research has contributed some insights into methodologies and algorithms. This is the first study to use text mining to analyse meeting transcripts, to extract the elements of decision making which can be used as a basis for the development of a decision management system.

REFERENCES

[1] P. Rayson, B. Sharp, A. Alderson, J. Cartmell, C. Chibelushi, R. Clarke, A. Dix, V. Onditi, A. Quek, D. Ramduny, A. Salter, H. Shah, I. Sommerville, and P. Windridge, "Tracker: A Framework to Support Reducing Rework Through Decision Management.," Proceedings of 5th International Conference on Enterprise Information Systems ICEIS2003, Angers - France, 2003.

[2] B. Boehm and V. R. Basil, "Software Reduction Defects Top 10 List," IEEE Computer, vol. 34, pp. 135-137, 2001.

[3] T. Jones, "Software Productivity," McGraw-Hill, 1986

[4] W. Melo, L. Briand, and V. Basili, "Measuring the Impact of Reuse on Quality and Productivity in Object-Oriented Systems," Department of Computer Science, University of Maryland, College Park, MD, Technical Report CS-TR-3395, 1995.

[5] ESA, "Ariane 5 - Flight 501 Failure," European Space Agency, 1996.

[6] B. Nuseibeh, "Ariane 5: Who Dunnit?," IEEE Software, vol. 14, pp. 15-16, 1997.

[7] S. Cross, "Message from the Director," Software Engineering Institute, Annual Report 2002.

[8] J. Goodman, "Lessons Learned From Flights of "Off the Shelf" Aviation Navigation Units on the Space Shuttle," presented at Joint Navigation Conference, Orlando, Florida, 2002.

[9] H. Simon, The new Science of Management Decision. New York: Harper & Row, 1960.

[10] I. L. Janis, Stages in the Decision Making Process. Chicago, 1968.

[11] L. P. Schrenk, "Aiding the Decision Maker - A Decision Process Model," Ergonomics, pp. 543-557, 1969.

[12] E. Witte, "Field Research on Complex Decision-Making Processes - The Phase Theorem," International Studies of Management and Organisation, pp. 156-182, 1972.

[13] B. Henrikke, Life cycle assessment and Decision Making-theories and practices. GÖteborg: Chalmers University of Technology, 1998.

[14] J. Adair, Effective Decision-Making. London: Pan Books, 1985.

[15] L. Fulop, S. Linstead, S. Lilley, and R. J. Clarke, "Decision Making in Organizations," in Management: A Critical Text. Australia: Macmillan Education, 1999, pp. 462-494.

[16] C. Pescatore and A. Vári, "Stepwise Approach to Decision Making for Long-term Radioactive Waste Management," The NEA Radioactive Waste Management Committee NEA No. 4429, 2004.

[17] H. Mintzberg and F. Westley, "Decision Making: It's Not What You Think," MIT Sloan Management Review, vol. 42, pp. 89-93, 2001.

[18] J. G. March, Theories of choice and making decisions, 1984.

[19] W. Hoy and C. Miskel, Educational Administration: Theory, Research, and Practice, 4 ed. New York: McGraw-Hill, 1991.

[20] J. Chen, "A Comparative Study on Decision Making Approach in a Dilemma between Taiwanese and Italian Business," in Graduate Institute of Management Sciences. Taiwan: Tamkang University, 2002.

[21] F. Harrison, The Managerial Decision-making Process. USA: Houghton Mifflin Company, 1999.

[22] B. J. Grosz and C. L. Sidner, "Attention, Intentions, and the Structure of Discourse," Computational Linguistics, vol. 12, pp. 175-204, 1986.

[23] J. Morris and G. Hirst., "Lexical Cohesion Computed," Computational Linguistics, vol. 17, pp. 21-48, 1991.

[24] M. Halliday and R. Hasan, Cohesion in English: Longman, London, 1976.

[25] M. Hearst, "TextTilling: Segmenting Text into Multi-Paragraph Subtopic Passages.," Computational Linguistics, vol. 23, pp. 33-64, 1997.

[26] J. Reynar, "Topic Segmentation: Algorithms and Applications," in Computer and Information Science. College Park, Maryland: University of Pennsylvania, 1998.

[27] N. Stokes, "Applications of Lexical Cohesion Analysis in the Topic Detection and Tracking Domain.," in Department of Computer Science: University College Dublin, 2004.

[28] A. M. Stairmand, "A Computational Analysis of Lexical Cohesion with Applications in Information Retrieval," in Department of language Engineering: University of Manchester Institute of Science and Technology, 1996.

[29] P. Rayson, "Wmatrix: a Web-based Corpus Processing Environment.," presented at ICAME 2001 Conference, Université Catholique de Louvain, Belgium., 2001.

[30] F. Mandreoli, R. Martoglia, and P. Tiberio, "Text Clustering as a Mining Task," in Text Mining and its Applications to Intelligence, CRM and Knowledge Management A. Zanasi, Ed. GB: WIT Press, 2005, pp. 75-108.

[31] C. Manning and H. Schütze, Foundations of Natural Language Processing, 6 ed: The MIT Press London, 2003.

[32] J. Meij and A. Bosch, "Text Mining Techniques," in Dealing with the Data Flood, J. Meij, Ed. The Hague, Netherland: STT/Beweton, 2002, pp. 746-753.

[33] M. Steinbach, G. Karypis, and V. Kumar, "A Comparison of Document Clustering Techniques," presented at Proceedings of the KDD Workshop on Text Mining, Boston, MA, USA, 2000.

[34] K. Lagus, "Text mining with the WEBSOM," in Computer Science and Engineering. Helsinki: Helsinki University, 2000, pp. 58.

[35] T. Kohonen, Self Organization and Associative Memory. Berlin, Heidelberg, New York: Springer Verlag 1989.

[36] S. M. Katz, "Distribution of Context Words and Phrases in Text and Language Modelling," Natural language Engineering, vol. 2, pp. 15-59, 1996.

[37] D. Beeferman, A. Berger, and J. Laffety, "Text Segmentation Using Exponantial Models," presented at Proceedings of EMNLP-2, Madrid, Spain, 1997.

[38] L. Pevzner and M. Hearst, "evaluation Metric for Text Segmentation," Computational Linguistics, vol. 28, pp. 19-36, 2002.

[39] F. Y. Choi, "Advances in Domain Independent Linear Text Segmentation," presented at Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics (ANLP-NAACL-00), Seattle, 2000.