Conversational Requirements Engineering: Pinpointing Requirements-Relevant Information in Conversations

Tjerk Spijkman[1,2]

[1] *fizor., Utrecht, The Netherlands*
[2] *Dept. of Information and Computing Sciences, Utrecht University, The Netherlands*

Abstract

Discussions about system requirements, for instance interviews, analysis workshops and customer meetings, are essential in the process of gathering requirements and contribute significantly to the creation of requirements specifications. In the majority of cases, the tasks in this process – preparation, elicitation, note-taking, and post-processing – are manually performed by the practitioners. Consequently, important information could be overlooked or omitted in the requirements specification. The unavailability of data and the challenges associated with transcribing conversational data have resulted in very limited research until today. However, by leveraging recent advancements in meeting technology, such as Microsoft Teams, we conduct empirical research to investigate conversational artifacts. The resulting findings are then utilized in the design science-oriented development of solutions aimed at assisting practitioners in the requirements engineering process.

Keywords

Requirements Elicitation, Natural Language Processing, Conversational RE, Requirements-Relevant Information

1. Context and Motivation

Requirements engineering (RE) is a critical aspect in software and information systems design and development. It is necessary in order to achieve a comprehensive understanding of the application domain, stakeholders, and system objectives [1]. The process of gathering system requirements is typically carried out through conversations, with interviews and facilitated meetings being the most commonly used techniques [2].

Despite being common in practice, the activities involved in requirements elicitation have received limited attention in research. We have previously argued that this is partly due to the unstructured nature of the data and limited availability of research data, stemming from confidentiality concerns or lack of recordings [3]. However, the COVID-19 pandemic has brought about changes in the practitioner domain and online meeting tools, providing an

In: A. Ferrari, B. Penzenstadler, I. Hadar, S. Oyedeji, S. Abualhaija, A. Vogelsang, G. Deshpande, A. Rachmann, J. Gulden, A. Wohlgemuth, A. Hess, S. Fricker, R. Guizzardi, J. Horkoff, A. Perini, A. Susi, O. Karras, A. Moreira, F. Dalpiaz, P. Spoletini, D. Amyot. Joint Proceedings of REFSQ-2023 Workshops, Doctoral Symposium, Posters & Tools Track, and Journal Early Feedback Track. Co-located with REFSQ 2023. Barcelona, Catalunya, Spain, April 17, 2023.

tjerk@fizor.io (T. Spijkman)
0000-0003-2726-3065 (T. Spijkman)

opportune time for advancements in this area of research. Requirements elicitation sessions are now frequently conducted digitally, and meeting tools such as Teams and Zoom Meetings have rapidly evolved to meet the increased demand. Notably, these developments have enabled the application of neural network approaches to improve the quality of transcriptions [4].

Our research is focused on the domain of Conversational Requirements Engineering, defined in previous works as: "The analysis of requirements elicitation conversations aimed at identifying and extracting requirements-relevant information" [3]. The objectives of this research domain are two-fold: first, to enhance understanding available in the research domain for the content of these conversations, such as identifying information that is relevant to requirements engineers, classifying the various topics that are discussed, the evolution of requirements information during the conversations, and the impact of conversation structures. Second, we aim to leverage the knowledge gained from this research to support the requirements engineering process by developing tools that can assist practitioners.

2. Related Work

There is very little research available on requirements conversations as an artifact. Ferrari *et al.* [5] conducted research on ambiguity in a set of 34 simulated interviews, and identified four facets of ambiguity. In a more recent study, Ferrari *et al.* [6] investigated conversational artifacts, with a focus on voice and biofeedback, to recognize engagement. Alvarez and Urza [7] explored the role of stakeholders and clients through manual review of interview transcripts.

A related research field is pre-Requirements Specification (pre-RS) traceability, which aims to trace requirements back to artifacts prior to the creation of the specification. Although pre-RS was recognized for its potential in the 90s by Gotel [8], it remains an underexplored area of research [9]. Another adjacent field is the extraction of declarative process models from natural language. Aa *et al.* [10] introduced an approach for the automatic extraction of declarative process models in the Declare language. The automation process faces several challenges, including the use of synonymous terms and phrases, discrepancies in order, noun-based actions, and negation. Some of these challenges are also present in our artifacts.

Several studies have also focused on analyzing and processing existing requirements specifications. For example, Abualhaija *et al.* proposed a natural language processing pipeline for detecting and delineating requirements in a document containing natural language requirements specifications. Kurtanović and Maalej [11], explored the automated classification of requirements into functional and non-functional categories.

Although the research on conversation structures extends beyond the domains of Requirements Engineering and Information Systems, it presents useful insights for our investigation. One such field is that of Conversation Analysis, the systematic analysis of conversations produced in everyday human interaction [12]. This field provides us additional avenues to consider in our research, like the artifacts used (screens, whiteboards, video game events etc.), overlapping talk and tempo of speech. Similarly, Speech Act theory [13] provides valuable knowledge about the use of language and how small differences can change the meaning of an utterance.

There are existing studies on the summarization of non-Requirements Engineering (RE) domain conversations. For instance, Fabbri *et al.* provide benchmark datasets for summarization tools and test state-of-the-art models against these [14]. And Chen and Yang, present a model for abstractive summarization that includes discourse relations [15].

3. Research Method

The PhD track is focused on the investigation of a relatively unexplored domain, Conversational Requirements engineering. In the broadest sense, we have two research questions;

RQ1 What is the relevance of information contained in Requirements Engineering conversations to practitioners, and what are the specific use cases where this information can be applied effectively?

RQ2 How can we effectively support practitioners in the identified use cases with requirements engineering focused software tooling?

This means that there are two aspects in our research; theory building and design science. On the one hand we must gain an understanding of the human process. On the other hand we aim to design these processes through automation and providing important information to the practitioners at the right time and place. To achieve these goals, we initiated a collaboration between research and practice, more specifically fizor., a consultancy company focused on the low-code domain, and Utrecht University. This enables us to gather both practitioner data and input, knowledge and state-of-the-art insights on the gathered data from a research perspective.

The dual nature of our research requires the use of different research methods in various research steps. For the design and prototyping phases, we primarily rely on the Design Science methodology proposed by Wieringa [16] to provide a structured approach to our research. In contrast, for theory building, we employ a range of research methods, such as exploratory case studies, action research [17] , student experiments, and grounded theory [18]. Gathering of data is predominantly through real-world cases provided by fizor. and student experiments through Utrecht University. Additionally, we utilize tooling like Microsoft Teams to generate the transcriptions in most scenarios to minimize intrusion in the practice. We have published works related to *RQ 1* [19, 20, 21] and *RQ 2* [22, 3, 23] as part of the PhD track, providing a preliminary understanding of the research domain and artifacts. However, our knowledge is limited to specific elicitation methods, conversation contexts, and use cases. Additionally, we aim to collect the prototypes developed for *RQ 2* in a user friendly toolkit and need to evaluate the usefulness and effectiveness.

4. Solution Proposal

The foundation of the research is based on a set of observations made from Requirements in Practice during the application of the RE4SA (Requirements Engineering for Software Architecture) model [19] on a set of case studies for a software product in the ERP domain. We then extended this approach by introducing metrics that make the link between requirements and architecture measurable [20]. In practice, we found that Agile methods tend to result in limited creation and maintenance of documentation. As a consequence, agreements are mostly based on a shared project understanding, and design artifacts created by practitioners are often limited to a set of user stories with minimal context. Additionally, we found a difference in the source of requirements and whether they were related to configuration or customization.

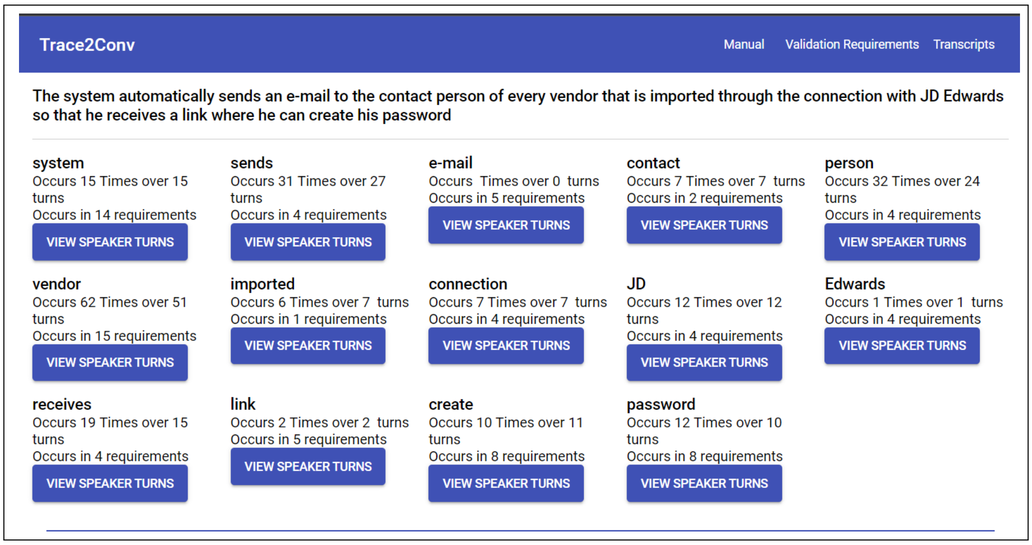
In software development, initial stages of a projects involve an analysis, scoping, or a sprint 0 in Agile. During this phase, requirements are discussed and gathered through analysis meetings, brainstorming sessions and discussed with stakeholders for validation and revision. Meetings may also be held to discuss the initial design, user journeys or data-models. What all these have in common, is human communication, either during collaborative design, or when validating documentation. These stages present a source of knowledge that is mostly untapped in the software tooling domain. With our research we set out to specify the knowledge contained in these conversations, and work on ways to make it easily accessible.

To this end, we started theory building through grounded theory research to find patterns and make observations on the content of fit-gap analysis conversations. In this work, we relate customization and configuration to fit-gap analysis, an analysis method that compares system capabilities to the customer requirements. We determined a categorization, and performed a validation of the perceived importance of these categories [21]. Similar research was performed as part of a master thesis on the specific context of pre-sales conversations, but has remained unpublished. This enabled us to develop a foundation and comprehension of the human processes involved, which can serve as a guide to the design of support tools.

The research team, where I am the primary conceptual contributor, developed three prototypes to assist with requirements engineering tasks. The first prototype is a concept extraction tool that utilizes existing Python packages to extract relevant concepts from a conversation transcript [22]. This tool scans the transcript of a conversation and compares it to an ontology of a software product (or domain). It then specifies the most discussed concepts, and categorizes them as known or unknown. These known concepts can be used to locate important topics for configuration, for instance different approval options in an invoice automation software. In contrast, the unknown concepts can either help recognize important context from the domain of the customer, or indicate customization. An unknown concept can for example indicate a remittance process, which specifies a set of payments through a single document. If its not part of the application scope of an invoice automation system, it might be required to change the process, or change the tooling to support the existing process. The outputs of the concept extraction tool were collected in a dashboard mockup and presented to domain experts. Although they found the information valuable, they also expressed a desire to see the context of the conversation related to the identified concepts. This feedback highlighted the need to collect and display more content from the transcripts, which guided our later prototypes.

The second prototype, Trace2Conv, focuses on backward traceability from a requirements specification and a transcript. Trace2Conv links the requirements in the specification to the relevant speaker turns in the transcript. This is achieved through token matching algorithms [3] that identify the speaker turns that are more likely to be relevant to a requirement. The outputs are displayed in a user interface to enable users to explore the transcript from their specification.

A front-end interface is provided to users to select a requirement and view the speaker turns in which a specific concept, such as email notifications, was mentioned. Users can select relevant speaker turns to see the conversation surrounding the concept (configured to 5 minutes before and after the selected speaker turn). This prototype provides a means to explore conversations for additional context related to a requirement. It can be used by developers to extract information from conversations they were not present in, or by analysts to review requirements during the revision of specification documents to identify missed requirements. A screenshot of the prototype is shown in Fig. 1. The tool is currently being extended to suggest relevant speaker turns on a user story level.



The system automatically sends an e-mail to the contact person of every vendor that is imported through the connection with JD Edwards so that he receives a link where he can create his password

system
Occurs 15 Times over 15 turns
Occurs in 14 requirements

sends
Occurs 31 Times over 27 turns
Occurs in 4 requirements

e-mail
Occurs Times over 0 turns
Occurs in 5 requirements

contact
Occurs 7 Times over 7 turns
Occurs in 2 requirements

person
Occurs 32 Times over 24 turns
Occurs in 4 requirements

vendor
Occurs 62 Times over 51 turns
Occurs in 15 requirements

imported
Occurs 6 Times over 7 turns
Occurs in 1 requirements

connection
Occurs 7 Times over 7 turns
Occurs in 4 requirements

JD
Occurs 12 Times over 12 turns
Occurs in 4 requirements

Edwards
Occurs 1 Times over 1 turns
Occurs in 4 requirements

receives
Occurs 19 Times over 15 turns
Occurs in 4 requirements

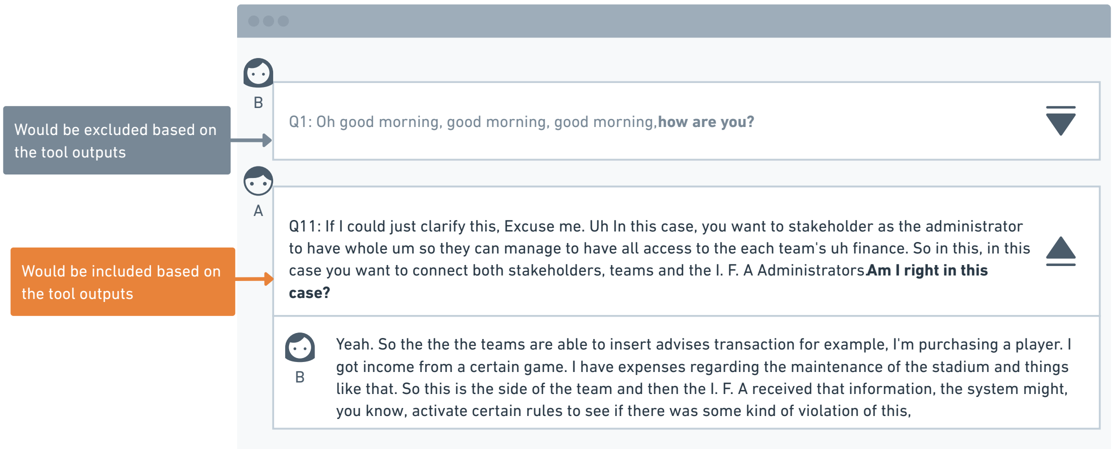
link
Occurs 2 Times over 2 turns
Occurs in 5 requirements

create
Occurs 10 Times over 11 turns
Occurs in 8 requirements

password
Occurs 12 Times over 10 turns
Occurs in 8 requirements

Figure 1: User Interface for the Requirements Token Review in Trace2Conv

Building on the ideas of the concept extraction tool, the third prototype REConSum was developed to perform extractive summarization of a requirements conversation [23]. This prototype filters speaker turns in a conversation transcript to only keep questions that are expected to contain or answer requirements-relevant information. A mock-up of the user interaction for REConSum can be seen in Fig. 2.



Would be excluded based on the tool outputs

Would be included based on the tool outputs

Q1: Oh good morning, good morning, good morning,**how are you?**

Q11: If I could just clarify this, Excuse me. Uh In this case, you want to stakeholder as the administrator to have whole um so they can manage to have all access to the each team's uh finance. So in this, in this case you want to connect both stakeholders, teams and the I. F. A Administrators**Am I right in this case?**

Yeah. So the the the teams are able to insert advises transaction for example, I'm purchasing a player. I got income from a certain game. I have expenses regarding the maintenance of the stadium and things like that. So this is the side of the team and then the I. F. A received that information, the system might, you know, activate certain rules to see if there was some kind of violation of this,

Figure 2: Mockup visualization of the outputs of REConSum

RECONSUM achieves this through a implementation that utilizes Part of Speech tagging, and Dialog Act recognition to recognize speaker turns containing a question. These are then filtered by utilizing TF-IDF against a general corpus to see if they contain domain specific terms, which indicates their relevance. This enables practitioners to review a conversation into a FAQ like interaction used in web-design. The outputs can be further classified in future iterations in the different topics discussed for increased ease of use.

5. Research Plan

For our research in Conversational RE, our goal is to develop a toolkit that can be applied in diverse use-cases. However, several challenges need to be addressed in order to achieve this goal. These challenges include creating a user-friendly interaction that aligns with the real-world process, reducing the effort required to use the tools, integrating all components into a single technology stack, and defining key concepts such as requirements-relevance.

Within the context of this PhD project in conversational requirements engineering, we have planned the following efforts: *(1) Investigating the evolution of requirements* to understand patterns and where they occur in subsequent meetings, documentation, chats, or project management tooling. This expands our understanding beyond single sessions and introduces more factors relevant to managing requirements from a conversational perspective. *(2) Extending the usability of our TRACE2CONV tool* [3], which currently relates requirements to a single transcription, to be project-based. This would provide practitioners with relevant information for a requirement from all recorded meetings of the project, ordered based on likely relevance. This would enable the tool to be used in action research and further review its usability. *(3) Utilizing conversational artifacts to generate a data model*, especially with advances in model driven design and low-code platforms in the market, which could transform a conversation into an early version of the application and present helpful concepts to the practitioner. This would also allow us to utilize low-code tooling and address the limited creation and maintenance of documentation in agile methods.

In the context of the Conversational Requirements Engineering (RE) field, our team is extending both the theory-building and design aspects. In theory-building, our focus is on expanding the scope of our understanding beyond conversations within a specific project or individual recordings. In the design aspect of our research, we are refining existing prototypes for ease of use in practical settings, while also designing new tools. We are also exploring the potential of state-of-the-art technologies, such as ChatGPT and OpenAI's endpoints, for our research.

References

[1] C. Jones, Software project management practices: Failure versus success, J. Defense Softw. Eng. 17 (2004) 5–9.

[2] S. Wagner, et al., Status quo in requirements engineering: A theory and a global family of surveys, ACM Trans. Softw. Eng. Methodol. 28 (2019).

[3] T. Spijkman, F. Dalpiaz, S. Brinkkemper, Back to the roots: Linking user stories to requirements elicitation conversations, in: Proc. of IEEE RE, 2022.

[4] M. M. Archibald, R. C. Ambagtsheer, M. G. Casey, M. Lawless, Using zoom videoconferencing for qualitative data collection: perceptions and experiences of researchers and participants, IJQM 18 (2019).

[5] A. Ferrari, P. Spoletini, S. Gnesi, Ambiguity and tacit knowledge in requirements elicitation interviews, Requirements Engineering 21 (2016) 333–355.

[6] A. Ferrari, T. Huichapa, P. Spoletini, N. Novielli, D. Fucci, D. Girardi, Using voice and biofeedback to predict user engagement during requirements interviews, arXiv:2104.02410 (2021).

[7] R. Alvarez, J. Urla, Tell me a good story: Using narrative analysis to examine information requirements interviews during an ERP implementation, ACM SIGMIS Database 33 (2002).

[8] O. C. Gotel, Requirements traceability, Technical Report, Oxford University, 1992.

[9] J. Krause, A. Kaufmann, D. Riehle, The Code System of a Systematic Literature Review on Pre-Requirements Specification Traceability, Technical Report, 2020.

[10] H. van der Aa, C. Di Ciccio, H. Leopold, H. A. Reijers, Extracting declarative process models from natural language, in: P. Giorgini, B. Weber (Eds.), CAiSE, 2019, pp. 365–382.

[11] Z. Kurtanović, W. Maalej, Automatically classifying functional and non-functional requirements using supervised machine learning, in: Proc. of IEEE RE, 2017, pp. 490–495.

[12] T. Stivers, J. Sidnell, The handbook of conversation analysis, John Wiley & Sons, 2012.

[13] J. R. Searle, J. R. Searle, Speech acts: An essay in the philosophy of language, volume 626, Cambridge University Press, 1969.

[14] A. R. Fabbri, F. Rahman, I. Rizvi, B. Wang, H. Li, Y. Mehdad, D. Radev, Convosumm: Conversation summarization benchmark and improved abstractive summarization with argument mining, arXiv preprint arXiv:2106.00829 (2021).

[15] J. Chen, D. Yang, Structure-aware abstractive conversation summarization via discourse and action graphs, arXiv preprint arXiv:2104.08400 (2021).

[16] R. J. Wieringa, Design science methodology for information systems and software engineering, Springer, 2014.

[17] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, Empirical Software Engineering 14 (2009) 131–164.

[18] K.-J. Stol, P. Ralph, B. Fitzgerald, Grounded theory in software engineering research: A critical review and guidelines, in: Proc. of ICSE, 2016, pp. 120–131.

[19] T. Spijkman, S. Brinkkemper, F. Dalpiaz, A.-F. Hemmer, R. van de Bospoort, Specification of requirements and software architecture for the customisation of enterprise software, in: Proc. of RE Workshops, 2019, pp. 64–73.

[20] T. Spijkman, S. Molenaar, F. Dalpiaz, S. Brinkkemper, Alignment and granularity of requirements and architecture in agile development: A functional perspective, Information and Software Technology 133 (2021) 106535.

[21] T. Spijkman, F. Dalpiaz, S. Brinkkemper, Requirements elicitation via fit-gap analysis: A view through the grounded theory lens, in: Proc. of CAiSE, Springer, 2021, pp. 363–380.

[22] T. Spijkman, B. Winter, S. Bansidhar, S. Brinkkemper, Concept extraction in requirements elicitation sessions: Prototype and experimentation, in: Proc. of REFSQ Workshops, 2021.

[23] T. Spijkman, X. de Bondt, D. Fabiano, S. Brinkkemper, Summarization of elicitation conversations to locate requirements-relevant information, in: Proc. of REFSQ, 2023.