

LadderBot: A requirements self-elicitation system

Tim Rietz

Institute of Information Systems and Marketing (IISM)
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
tim.rietz@kit.edu

Alexander Maedche

Institute of Information Systems and Marketing (IISM)
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
alexander.maedche@kit.edu

Abstract—[Context] Digital transformation impacts an ever-increasing amount of everyone’s business and private life. It is imperative to incorporate user requirements in the development process to design successful information systems (IS). Hence, requirements elicitation (RE) is increasingly performed by users that are novices at contributing requirements to IS development projects. [Objective] We need to develop RE systems that are capable of assisting a wide audience of users in communicating their needs and requirements. Prominent methods, such as elicitation interviews, are challenging to apply in such a context, as time and location constraints limit potential audiences. [Research Method] We present the prototypical self-elicitation system “LadderBot”. A conversational agent (CA) enables end-users to articulate needs and requirements on the grounds of the laddering method. The CA mimics a human (expert) interviewer’s capability to rephrase questions and provide assistance in the process. An experimental study is proposed to evaluate LadderBot against an established questionnaire-based laddering approach. [Contribution] This work-in-progress introduces the chatbot LadderBot as a tool to guide novice users during requirements self-elicitation using the laddering technique. Furthermore, we present the design of an experimental study and outline the next steps and a vision for the future.

Index Terms—User, Requirements Elicitation, Wide Audience, Conversational Agent, Self-Elicitation, Laddering

I. INTRODUCTION

Digital transformation has brought a variety of information systems into everyone’s business and private life with a substantial impact on business and society [1]. We observe a transformation towards a digital society, stressing the influence of the Internet on many traditional services, which advocates a power shift towards the user [2]. In the face of persistently high failure rates of IS development projects, it is imperative that an increasing number of users is involved in RE processes, with a varying degree of technological and methodological expertise [3]. The scalable elicitation of user requirements is crucial for developing software that meets needs and demands and to reduce project failure [4]. Consequently, RE needs to be performed with a wide range of users that are novices at contributing requirements to development projects [1].

For requirements elicitation, interviews have been used most widely [5]. Especially the laddering interview is considered a very effective technique for eliciting relevant information for

articulating requirements [5]. Laddering produces comprehensive and structured insights due to the method’s hierarchical nature. In laddering, an interviewer identifies a seed attribute, an initial topic, and asks a series of “why...?” questions to uncover and clarify needs and related attitudes [6]. While having its roots in personality psychology, laddering has already seen usage for requirements elicitation [4] (e.g., to elicit Customer Attribute Hierarchies [7]). Essentially, requirements are elicited as attribute-consequence-value (ACV) chains [6]. Since laddering interviews require highly trained and experienced interviewers, the availability of suitable interviewers imposes a bottleneck onto elicitation interviews [6]. Tool support is necessary to enable requirements elicitation with a wide range and number of users [8].

Several tools to aid with wide audience elicitation have been proposed over the years. AnnotatePro allows users to submit requirements that can be drawn on their screens [9]. Given the common problems with requirements quality, such as completeness and ambiguity, exploring natural-language (NL) based elicitation systems gained traction. Pérez and Valderas (2009) combine visualization-based RE with NL to reduce ambiguity and inconsistency in end-user RE. Derrick et al. (2013) evaluated an embodied conversational agent to facilitate a group workshop that used prompts to guide and assist during user story formulation [10]. However, these tools do not suffice in providing a solution to both challenges introduced: Annotation-based tools primarily enable RE for iteratively improving existing systems; NL tools commonly require a requirements engineer to facilitate the process, hence retaining a bottleneck for wide audience integration [11]; additionally, existing research rarely considers (methodological) guidance for novice end-users. Tools such as FAME [12] and ASSERT [13] cater to novices, but only on the side of a novice analyst, not novice users, hence not enabling self-elicitation. A literature gap remains in extending RE techniques to wide audiences. Guidance and assistance are necessary to facilitate the elicitation of high-quality requirements from novice users [14], [15]. We utilize a conversational agent (CA) in the form of a chatbot to mimic a human interviewer’s capability to guide an interview [10]. Chatbots allow us to include a wide audience of users, independent of personal, time, or location restrictions and may guide novice users through laddering interviews. Therein, we extend our previous research on (semi-)automated RE by explicitly focusing on the collection of unstructured data on the basis of self-elicitation interviews [16], [17].

II. CONCEPTUAL FOUNDATIONS

A. Common issues in user elicitation interviews

To understand the implications for a novice-centric self-elicitation system, we need to understand the characteristics of the requirements (self-)elicitation behavior of novices. In this article, we refer to self-elicitation of requirements rather than self-service RE system. As the user is guided in uncovering their requirements, rather than being enabled to create a service with a direct benefit for themselves, we argue that self-elicitation serves as a better term to describe the process.

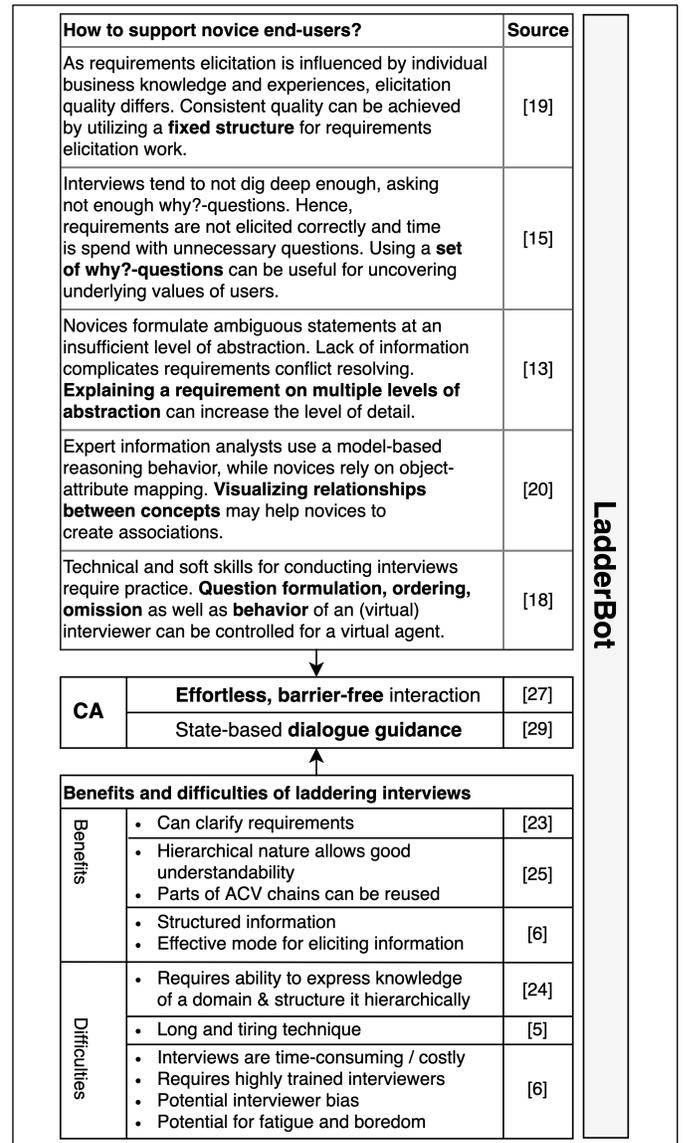
So far, RE literature rarely focuses on characteristics of novice users to be supported in elicitation processes [1]. Commonly, novice RE analysts are the focus of supporting activities [18]. However, insights from analyzing the behavior of novice analysts in elicitation processes may serve as a guideline for how to provide appropriate support for requirements self-elicitation.

Notably, one of the most frequently observed downfalls in elicitation performed with novice users or by novice analysts is a lack of structure [19]. A lack of structure results in interviewers not digging deep enough when conducting interviews, impacting requirements correctness [15]. Since especially novice users are not familiar with communicating requirements, which may be rooted in an incomplete understanding of their own needs, the task of uncovering the cause of a need or requirement falls to the interviewer. Otherwise, interviews lead to ambiguous user statements at the wrong level of abstraction [13]. Without uncovering the cause of, or foundation for user needs, the development of disruptive solutions stagnates. We can avoid common mistakes of novice analysts that happen during interviews, such as question formulation, ordering, and question omission through a pre-defined interview structure [18]. Furthermore, the analyst's behavior, such as lack of confidence or unprofessionalism, or inadequate time management, has a substantial impact on the results of an interview [21]. Hence, bot structural and behavioral interview guidelines are necessary for eliciting high-quality requirements.

Analysts should be educated in thinking in relations, hence applying model-based reasoning rather than object-attributes to increase the performance of requirements analysis [20]. We propose that by using an elicitation structure following the laddering technique, we can enable users to generate requirements in a relation-focused fashion, contributing to the quality of requirements specification. Fig. 1 provides an overview of how the conceptual foundations feed into the development of LadderBot.

B. The laddering interview technique for RE

Laddering is a cognitive interview technique with its roots in personality psychology that utilizes a structured approach for data-gathering [6]. For RE, cognitive techniques, in comparison to traditional, collaborative, or contextual techniques, are commonly used to acquire knowledge. As such, requirements are not directly communicated but extracted from the structure and content of user knowledge based on rich enough information [21]. Herein, cognitive techniques provide the most natural interaction with end-users [21].



LadderBot

Fig. 1. Overview of the conceptual foundations of LadderBot

Laddering was introduced as a method to elicit superordinate items from subordinate ones, to clarify the relations between items obtained using the repertory grid method, with its origin in personal construct theory. However, the laddering technique has primarily been used for knowledge-elicitation in marketing and advertising [22]. As such, the technique has become a tool for the means-end theory in marketing. The means-end theory distinguishes three levels of abstraction of meaning that users obtain from a purchase or consume [6]. These three levels are described as ACV chains: attributes – consequences – values [23]. *Attributes* as the least abstract level describe “concrete, physical, or observable characteristics” of products. Despite the notion initially describing physical products, we may use the idea for digital products like software, too [24]. *Consequences* constitute the second level of abstraction. They describe what a product provides a user with, either on the positive (benefits) or negative side (costs). A product can have functional or non-functional, e.g., psychosocial, consequences. *Values* are the most abstract

level. They represent a user’s wishes, goals, and needs and are the end state a customer is trying to achieve through a purchase. An exemplary ACV chain in a software development context has the following form: *Providing default values (A) – No need to fill out data repeatedly (C) – Happiness (V)* [24].

The laddering technique usually comprises three steps: elicitation of attributes, a laddering interview, and representing and analyzing the results. Attributes serve as the seed for the interview, in the form of lower order characteristics with implications for higher-order cognitive processes and determine the direction of the interview. As such, multiple methods of generating attributes have been used, depending on the purpose of related study. The laddering interview itself follows a straightforward structure. Participants are asked why a particular attribute is important to them, using a series of “why...?” questions while navigating through the ACV chains. E.g., an interviewer might ask, “why is starting process X from the landing page is important?”. A content-coding procedure initializes the analysis process of laddering interviews. These codes are then used to build a summary matrix, visualizing each chain from each participant, showing the included codes per chain. Subsequently, an aggregate implication matrix is formed, showing the aggregated information across interviews. This matrix contains all direct and indirect relations between attributes, consequences, and values. Finally, we can visualize the aggregate implication matrix as a hierarchy value map, a tree diagram showing either only direct or both direct and indirect relations at a specified cut-off value (for examples, see [24]–[26]).

C. Form and Function of Chatbots

The goal of CAs, as McTear (2002) puts it, is the “[...] effortless, spontaneous communication with a computer”. Klopfenstein et al. (2017) conducted a systematic analysis of one of the instantiations of CAs, chatbots, categorizing advantages for users and developers [27]. They find *instant availability*, a *gentle learning curve*, and *platform independence* to be among the most prominent benefits. Hence, we argue that chatbots serve as a promising form of CAs for approaching a large number of users. Instant availability and platform independence enable barrier-free interaction with the system. A gentle learning curve, resulting from an interaction mode that is familiar to novice users, texting, creates an effortless experience [27]. Multiple variants of chatbots have seen use over the years, which can be differentiated according to form and function [28]. The form of a chatbot describes the arrangement of aspects that do not primarily contribute to the utility of the bot (similar to non-functional requirements). For example, anthropomorphism comprises methods for making the appearance and behavior of a bot more human-like. Function describes aspects related to general performance, such as the bot’s dialogue control strategy. A frame-based bot uses question templates to provide information back to a user. These systems do not have pre-determined dialogue flows but adapt to user input, e.g., a software problem reparation tool [29].

Despite a renewed research interest in chatbots, due to advances in artificial intelligence [30], the integration of CAs into

RE remains spare. Derrick et al. (2013) investigated the effect of a simple scripted agent in facilitating group elicitation sessions with users [10] while other studies developed prototypes for frame-based agents in interview scenarios [31], [32]. While these studies evaluated the general applicability of CAs as facilitators of elicitation processes, to the best of our knowledge, no evaluation of chatbot-based requirements elicitation with a wide audience of end-users has been conducted, comparing the performance of a system with established processes on the basis of measures such as performance and perception [5].

III. LADDERBOT

LadderBot uses a two-column visualization, with a graphical representation of ACV chains on the left and a frame-based chatbot on the right, as shown in fig. 2. Initially, LadderBot welcomes users and provides a short explanation of the interface and the interview process. We adapted the subsequent laddering interview structure from Jung (2014). To begin the interview, LadderBot asks the user to state the three most frequently used features of a system as seed attributes for each chain. The following process is then repeated until participants constructed three chains. At the beginning of each chain, LadderBot asks an initial question to elicit the first consequence for the current attribute:

LadderBot: “As 2. example, you said Email. Why do you use Email? What do you obtain by using the function?”

User: “I need to know if someone needs something from me, and see if I got any updates from the services I signed up for.”

Rather than asking an initial default question, LadderBot integrates the specific attribute that users selected into question formulation. The line of questioning for consequences and values is repeated until a value is identified, or the user is unable to provide a more precise answer. When asking why-questions repeatedly, the chatbot will rely on four techniques for rephrasing questions to help and guide the user. We adapted these techniques from suggestions for human interviewers on how to conduct laddering interviews [6], as described in table 1. Fig. 3 depicts a visual overview of how the solution works in a laddering interview. For now, the four techniques are applied by LadderBot randomly. The rephrasing techniques primarily incorporate the seed attribute of the current ladder into the question formulation.

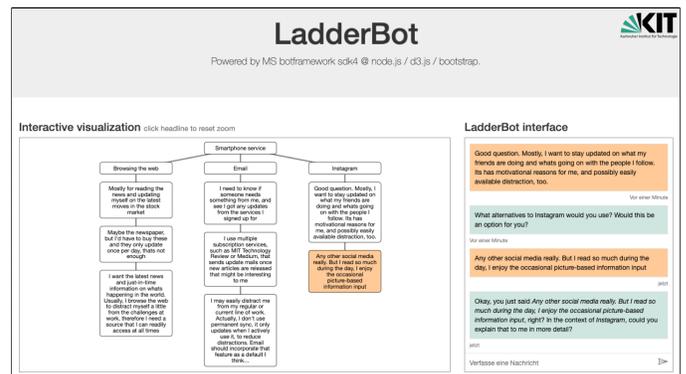


Fig. 2. The interface of LadderBot

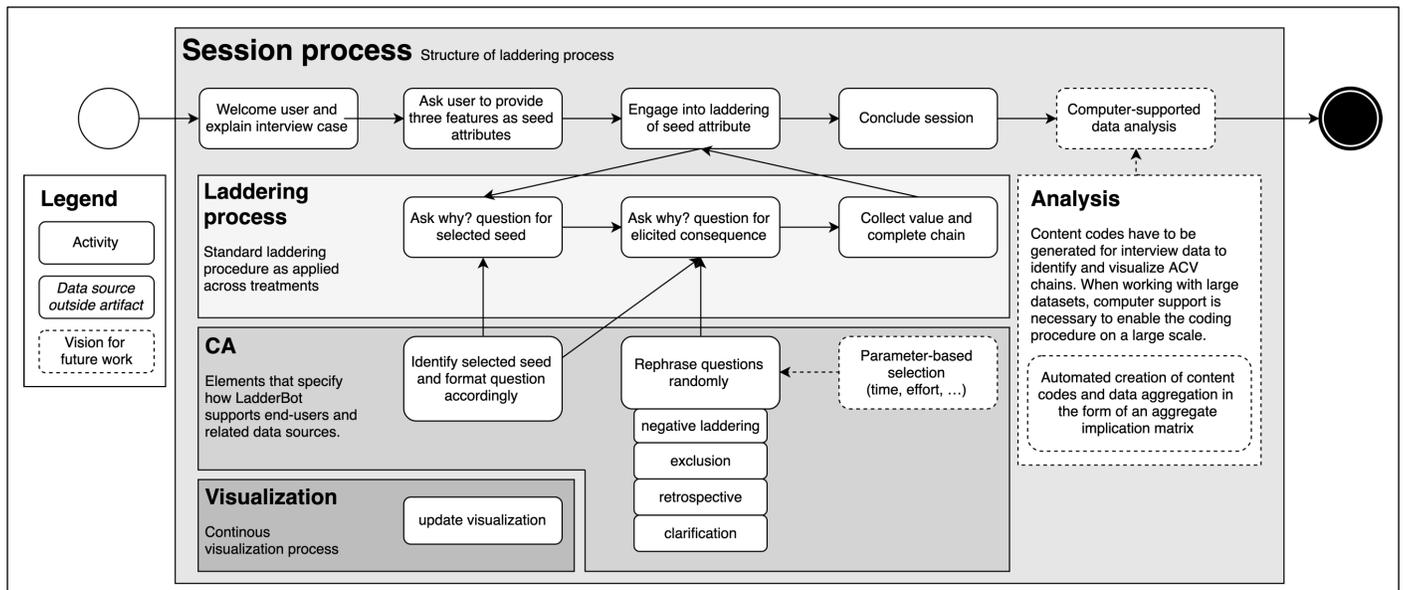


Fig. 3. Activity map of LadderBot

User replies are used for rephrasing only in the form of quotes, for ensuring that the resulting question makes sense. The visualization of the current status of the interview on the left side updates for each elicited consequence. The graphical representation of ACV chains may assist users in structuring their thoughts and uncovering new relations [20]. When asking a series of questions, a human interviewer would need to identify if the user has described the value that they satisfy through an attribute to end the elicitation for a specific attribute or to end the interview in general (e.g. [25]). As the current iteration of LadderBot is not capable of recognizing whether a user has already described a final value on its own, the bot requires the user to indicate if they want to continue the laddering process for the current attribute, or switch to the next chain. The user can make this indication with a predefined command (“stop”). The questioning process for each of the three ladders is continued until the stop command is given and LadderBot concludes the session.

The current implementation of LadderBot does not impose restrictions on the length of an answer of a user, to keep the interaction with the chatbot as natural as possible. Long replies impose a challenge for LadderBot in formulating an appropriate question as a response. As such, user replies are incorporated in questions only as complete references. Furthermore, LadderBot uses the three features provided by the users at the start of the interview to formulate more direct questions, as we identified these replies to be rather short. Users are currently not capable of making changes to previous answers. However, we plan to include this functionality in future iterations. As the technological foundation of LadderBot, we use the Microsoft Bot Framework on node.js. To visualize elicited ACV chains, we integrate the bot into a web application build on the frameworks d3.js and bootstrap. This architecture allows for a straightforward reconfiguration of the artifact to change the laddering use case or the interview structure.

IV. EXPERIMENTAL STUDY DESIGN

To evaluate LadderBot, we will conduct an experimental study. The experiment procedure and the applied measurements will partially build on previous studies that evaluated elicitation techniques [33], [34] or used the laddering technique as part of their experiment design [26].

We will conduct the study with students from a large university in Germany in an experimental lab designed for conducting scientific studies. As laddering case, we recreate the laddering structure applied by Jung (2014) to elicit the users’ goals for smartphone use. Such results may be used to uncover requirements to develop or improve an IS for smartphones. Similar to the original study, we will invite students as participants, while controlling for the participants’ experience with development projects and laddering interviews. Around 200 students will be invited, randomly selected from a pool of potential participants.

TABLE I. QUESTION REPHRASING TECHNIQUES

Technique	Description	Example
Negative laddering	Ask the user why they <i>do not</i> do something or <i>do not</i> want to feel a certain way	What problems could be caused by Email? How would Email have to change to mitigate these problems?
Exclusion	Ask the user to imagine a situation where an attribute or consequence does not exist	Imagine you could not use Instagram. What alternatives to Instagram would you use and why?
Retrospective	Ask the user to imagine their behavior in the past and compare it to now	Has your perception of this changed compared to a couple of years ago? If so, why is that and what changed?
Clarification	Repeat a reply back to the user and ask for clarification	Okay, you just said “I want a real-time newsfeed”, right? In the context of Instagram, could you explain that to me in more detail?

The experimental study will use a between-subject design with three treatments. Across treatments, participants will be asked to conduct a self-elicitation of their goals in smartphone use. Treatments will be characterized by the available interview tool and the interview visualization. In treatment (1), participants will use an established version of a “pencil-and-paper” laddering questionnaire [26]. However, a digital questionnaire will be used to increase comparability with other treatments. In treatment (2), participants will use the same questionnaire as in treatment (1) but augmented with the visualization used in LadderBot to keep track of already elicited ladders. In treatment (3), participants will use LadderBot to complete the laddering interview. As such, only one of either the visualization or the interview tool presented to participants is changed between treatments. Thereby, we aim to increase the comparability of results between treatments while being able to evaluate the visualization and chatbot interface features of LadderBot separately.

We will evaluate the treatments using a combination of quantitative measurements. Herein, we rely on the established procedure for analyzing the results of the laddering interviews [6]. We will calculate abstractness and centrality based on an aggregate implication matrix, which represents direct and indirect linkages between attributes, consequences, and values. Abstractness indicates whether constructs are predominantly at the beginning (attributes) or ends (values) of a chain. Constructs become increasingly abstract from means to ends. As such, it is a measure of importance in the means-ends structure [6]. Centrality measures the extent to which a concept is connected to all other concepts in the matrix and is used to evaluate the importance of a concept. Additionally, we will use the amount of direct/indirect links, the number of elicited consequences and values, and the time taken for comparing treatments [33]. Furthermore, after the treatments, we will apply a self-reporting questionnaire to collect the participants’ perceptions regarding the following constructs on a 7-point Likert scale: Understandability, Learnability, Efficiency, Effectiveness, and Enjoyment [34], as well as multiple constructs from the Big Five personality test. The self-reported measurements allow us to compare the perception of LadderBot against the established computer-based laddering questionnaire. Finally, we will incorporate multiple control questions, to evaluate the influence of experience, age, or gender, amongst others, on the experiment results.

V. ROADMAP AND CONCLUSION

This paper presents our work-in-progress for building LadderBot, a requirements self-elicitation system capable of guiding a novice user through a laddering interview to generate attribute-consequence-value chains as follows: *Elicitation guidance & assistance* - the user is supported through randomized rephrasing of questions based on an established guideline for interviewers; *Dynamic visualization* - Elicited attributes, consequences, and values are visualized for the user and continuously updated throughout the interview process.

We propose an experimental study design to evaluate LadderBot against the traditional approach of pencil-and-paper laddering using a digital questionnaire. As we will use the proposed structure for the evaluation of LadderBot and its subsequent

iterations, the scenario and the generated dataset might be helpful for other researchers for comparing CA-driven tool support for RE. Several comparisons of elicitation techniques have identified laddering as a very potent technique. However, only a limited amount of research describes approaches to creating tool support for laddering, especially for tool-supported self-elicitation of user requirements. A similar approach to our work-in-progress comes from Kassel and Malloy (2003), who attempt to automate requirements elicitation through combining domain knowledge, a software requirements specification (SRS) template and user needs as XML in a tool-based approach [35]. However, their focus lies on closed-ended questions, while the laddering tool proposed in our article relies on the detail introduced by open-ended questions.

Overall, we expect LadderBot to allow the elicitation of requirements from users without the need for highly qualified interviewers. Furthermore, enabling users to self-elicite requirements creates the potential to come in contact with a broad range of users, hopefully improving software development projects through detailed insights. In the spirit of “RE for everyone” [1], tool support for users enables developers to get an idea of the expectations of society and supports the end-to-end value co-creation between an outer- and an inner circle of systems development teams: between users and system engineers, analysts and developers. Additionally, with LadderBot, we also wish to show a proof-of-concept for using chatbots for RE, which may inspire the usage of the technology with elicitation techniques other than laddering in the future (e.g., 5W2H).

We are currently working on finalizing the LadderBot artifact and setting up a pre-test for the initial evaluation of the tool. Moving forward, we envision multiple adjustments to LadderBot, which will be evaluated in future studies: Enable the tool to use an interviewing technique (retrospective, ...) not randomly but based on measurements from the interview process, such as time since asking a question or based on user characteristics, e.g. cognitive styles [36]. For example, should a user diverge a specified amount from the average response time, the bot may provide additional assistance through question reformulation. Furthermore, future iterations of LadderBot will explore ways of generating content codes for the analysis of laddering interviews automatically. When dealing with a large number of self-elicitation interviews, it becomes necessary to provide requirements analysts with support in generating aggregate implication matrices and hierarchy value maps, ideally through an automated aggregation of results as well as an interactive visualization.

REFERENCES

- [1] K. Villela et al., “Towards ubiquitous RE: A perspective on requirements engineering in the era of digital transformation,” in 2018 IEEE 26th International Requirements Engineering Conference (RE’18), 2018, pp. 205–216.
- [2] J. M. Leimeister, H. Österle, and S. Alter, “Digital services for consumers,” *Electron. Mark.*, vol. 24, no. 4, pp. 255–258, 2014.
- [3] J. Jia and L. F. Capretz, “Direct and mediating influences of user-developer perception gaps in requirements understanding on user participation,” *Requir. Eng.*, vol. 23, no. 2, pp. 277–290, 2018.

- [4] H. F. Hofmann and F. Lehner, "Requirements Engineering as a Success Factor in Software Projects," *IEEE Softw.*, vol. 18, no. 4, pp. 58–66, 2001.
- [5] O. Dieste and N. Juristo, "Systematic review and aggregation of empirical studies on elicitation techniques," *IEEE Trans. Softw. Eng.*, vol. 37, no. 2, pp. 283–304, 2011.
- [6] G. M. Breakwell, *Doing Social Psychology Research*, 1st ed. The British Psychological Society and Blackwell Publishing Ltd, 2004.
- [7] C. H. Chen, L. P. Khoo, and W. Yan, "A strategy for acquiring customer requirement patterns using laddering technique and ART2 neural network," *Adv. Eng. Informatics*, vol. 16, no. 3, pp. 229–240, 2002.
- [8] O. Dieste, M. Lopez, and F. Ramos, "Updating a Systematic Review about Selection of Software Requirements Elicitation Techniques," in *11th. Workshop on Requirements Engineering Updating*, 2008.
- [9] A. Rashid, D. Meder, J. Wiesenberger, and A. Behm, "Visual requirement specification in end-user participation," in *First International Workshop on Multimedia Requirements Engineering, MeRE'06*, 2006.
- [10] D. C. Derrick, A. Read, C. Nguyen, A. Callens, and G. J. De Vreede, "Automated group facilitation for gathering wide audience end-user requirements," in *Annual Hawaii International Conference on System Sciences (HICSS'13)*, 2013, pp. 195–204.
- [11] F. Pérez and P. Valderas, "Allowing end-users to actively participate within the elicitation of pervasive system requirements through immediate visualization," *2009 4th Int. Work. Requir. Eng. Vis.*, pp. 31–40, 2009.
- [12] M. Oriol et al., "FAME: Supporting continuous requirements elicitation by combining user feedback and monitoring," in *2018 IEEE 26th International Requirements Engineering Conference (RE'18)*, 2018, pp. 217–227.
- [13] A. Moitra et al., "Towards development of complete and conflict-free requirements," in *2018 IEEE 26th International Requirements Engineering Conference (RE'18)*, 2018, pp. 286–296.
- [14] I. Mohedas, S. R. Daly, and K. H. Sienko, "Requirements Development: Approaches and Behaviors of Novice Designers," *J. Mech. Des.*, vol. 137, no. 7, pp. 1–10, Jul. 2015.
- [15] J. Kato et al., "A model for navigating interview processes in requirements elicitation," in *Proceedings of the Asia-Pacific Software Engineering Conference and International Computer Science Conference, APSEC and ICSC*, 2001, pp. 141–148.
- [16] H. Meth, M. Brhel, and A. Maedche, "The state of the art in automated requirements elicitation," *Inf. Softw. Technol.*, vol. 55, no. 10, pp. 1695–1709, 2013.
- [17] H. Meth, B. Mueller, and A. Maedche, "Designing a requirement mining system," *J. Assoc. Inf. Syst.*, vol. 16, no. 9, pp. 799–837, 2015.
- [18] M. Bano, D. Zowghi, A. Ferrari, P. Spoletini, and B. Donati, "Learning from mistakes: An empirical study of elicitation interviews performed by novices," in *Proceedings - 2018 IEEE 26th International Requirements Engineering Conference, RE 2018*, 2018, pp. 182–193.
- [19] T. Yamanaka, H. Noguchi, S. Yato, and S. Komiya, "A proposal of a method to navigate interview-driven software requirements elicitation work," *WSEAS Trans. Inf. Sci. Appl.*, vol. 7, no. 6, pp. 784–798, 2010.
- [20] I.-L. Huang and J. R. Burns, "A Cognitive Comparison of Modelling Behaviors Between Novice and Expert Information Analysts," in *Sixth Americas Conference on Information Systems (AMCIS 2000)*, 2000, pp. 1316–1322.
- [21] T. Tuunanen, "A new perspective on requirements elicitation methods," *J. Inf. Technol. Theory Appl.*, vol. 5, no. 3, pp. 45–72, 2003.
- [22] T. Tuunanen and M. Rossi, "Engineering a method for wide audience requirements elicitation and integrating it to software development," in *2004 37th Annual Hawaii International Conference on System Sciences (HICSS'04)*, 2004, pp. 1–10.
- [23] M. S. Mulvey, J. C. Olson, R. L. Celsi, and B. A. Walker, "Exploring the Relationships between Means End Knowledge and Involvement," *Adv. Consum. Res.*, vol. 21, pp. 51–57, 1994.
- [24] C. M. Chiu, "Applying means-end chain theory to eliciting system requirements and understanding users perceptual orientations," *Inf. Manag.*, vol. 42, no. 3, pp. 455–468, 2005.
- [25] Y. Jung, "What a smartphone is to me: Understanding user values in using smartphones," *Inf. Syst. J.*, vol. 24, no. 4, pp. 299–321, 2014.
- [26] G. Botschen, E. M. Thelen, and R. Pieters, "Using means-end structures for benefit segmentation," *Eur. J. Mark.*, vol. 33, no. 1/2, pp. 38–58, 2004.
- [27] L. C. Klopfenstein, S. Delpriori, S. Malatini, and A. Bogliolo, "The Rise of Bots: A Survey of Conversational Interfaces, Patterns, and Paradigms," in *2017 Conference on Designing Interactive Systems (DIS'17)*, 2017, pp. 555–565.
- [28] T. Rietz, I. Benke, and A. Maedche, "The Impact of Anthropomorphic and Functional Chatbot Design Features in Enterprise Collaboration Systems on User Acceptance," in *2019 14th International Conference on Wirtschaftsinformatik (WI'19)*, 2019, pp. 1656–1670.
- [29] M. F. McTear, "Spoken dialogue technology: enabling the conversational user interface," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 90–169, 2002.
- [30] U. Gnewuch, S. Morana, and A. Maedche, "Towards Designing Cooperative and Social Conversational Agents for Customer Service," in *2017 International Conference on Information Systems (ICIS'17)*, 2017, pp. 1–13.
- [31] J. F. Nunamaker, D. C. Derrick, A. C. Elkins, J. K. Burgoon, and M. W. Patton, "Embodied Conversational Agent-Based Kiosk for Automated Interviewing," *J. Manag. Inf. Syst.*, vol. 28, no. 1, pp. 17–48, 2011.
- [32] M. Pickard, R. M. Schuetzler, J. Valacich, and D. A. Wood, "Next-Generation Accounting Interviewing: A Comparison of Human and Embodied Conversational Agents (ECAs) as Interviewers," *SSRN Electron. J.*, no. April, pp. 1–21, 2017.
- [33] C. Corbridge, G. Rugg, N. P. Major, N. R. Shadbolt, and A. M. Burton, "Laddering: technique and tool use in knowledge acquisition," *Knowledge Acquisition*, vol. 6, pp. 315–341, 1994.
- [34] C. R. Coulin, "A Situational Approach and Intelligent Tool for Collaborative Requirements Elicitation," *University of Technology, Sydney*, 2007.
- [35] N. W. Kassel and B. A. Malloy, "An Approach to Automate Requirements Elicitation and Specification," *Proc. 7th Int. Conf. Softw. Eng. Appl.*, pp. 544–549, 2003.
- [36] O. Blazhenkova and M. Kozhevnikov, "The new object-spatial-verbal cognitive style model: Theory and measurement," *Appl. Cogn. Psychol.*, vol. 23, no. 5, pp. 638–663, Jul. 2009.