LadderBot:
A requirements self-elicitation system

Authors Name/s per 1st Affiliation *(Author)*
Dept. name of organization *(Line 1 of Affiliation - optional)*
Name of organization - acronyms acceptable *(line 2)*
City, Country *(line 3)*
name@xyz.com – optional *(line 4)*

Authors Name/s per 2nd Affiliation *(Author)*
Dept. name of organization *(Line 1 of Affiliation - optional)*
Name of organization - acronyms acceptable *(line 2)*
City, Country *(line 3)*
name@xyz.com – optional *(line 4)*

***Abstract*—[Context] Digital transformation impacts an ever-increasing amount of everyone's business and private life. To successfully design information systems (IS), it is imperative to incorporate user requirements in the development process. Hence, requirements elicitation (RE) is increasingly performed by users that are novices at contributing requirements to IS development projects. [Objective] It is becoming more and more important to provide RE systems that directly assist novice users in communicating their needs and requirements. To design such a requirements self-elicitation system, we integrate insights from the fields of requirements engineering and human-computer interaction (HCI). [Research Method] We present the prototypical self-elicitation system "LadderBot". Using a conversational agent (CA), it enables novice users to articulate needs and requirements on basis of the laddering method. A CA is used to mimic 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 tool to guide novice users during requirements self-elicitation using the laddering technique. Furthermore, we present the design of an experimental study and outline next steps and a vision for the future.**

***Index Terms*—User, Requirements Elicitation, Conversational Agent, Self-Elicitation, Laddering, Experimental Study.**

I. Introduction

Digital transformation has brought a variety of information systems into everyone's business and private life with strong impact on business and society [1]. Literature refers to these changes as a transformation towards a digital society, stressing the influence of the Internet on many traditional services and advocating 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 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 askes 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 required to enable requirements elicitation with a wide range and number of users [8]. Therefore, this work-in-progress presents a requirements self-elicitation system for novice users.

Several tools to aid with elicitation have been proposed over the years. 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 [9]. AnnotatePro allows users to submit requirements that can be drawn on their screens [10]. These approaches, amongst other such as WinWin [11] or EasyWinWin [12], allow users to communicate requirements. However, these tools do not consider the particular level of experience of users, limiting the utility of such tools for novice users. Tools such as FAME [13] and ASSERT [14] cater to novices, but only on the side of a novice analyst, not novice users, hence not enabling self-elicitation. To enable elicitation of high-quality requirements from novice users, guidance and assistance are necessary [15], [16]. We utilize a conversational agent (CA) to mimic a human interviewer's capability to provide guidance during an interview [9]. In contrast to graphical user interfaces, CAs use natural language conversations to interact with users. As such, CAs allow us to include a wide range of users by providing guidance using the laddering technique.

The remainder of this paper is as follows. In Section II, we introduce conceptual foundations for our proposed RE system. In Section III, we present the prototypical RE system "LadderBot" and propose an experimental study to evaluate the system against an established questionnaire-based laddering approach. The paper is concluded with a discussion of future work in Section IV.

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 own 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 [17]. 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.

Particularly, one of the most frequently observed downfalls in elicitation performed with novice users or by novice analysts is a lack of structure [18]. This is reflected by interviewers not digging deep enough when conducting interviews, impacting requirements correctness [16]. Since especially novice users are not familiar with communicating requirements, which may be rooted in an incomplete understanding of their own needs, the tasks 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 [14]. Without uncovering the cause of or foundation for user needs, the development of disruptive solutions stagnates. Common mistakes of novice analysts that happen during interviews, such as question formulation, ordering and question omission may be avoided through a pre-defined interview structure [17]. Furthermore, an analyst's behavior, such as lack of confidence or unprofessionalism, or inadequate time management, has substantial impact on the results of an interview [21]. Hence, bot structural and behavioral interview guidelines are necessary for eliciting high-quality requirements.

To increase performance of requirements analysis, analysts are suggested to be educated in thinking in relations, hence applying model-based reasoning rather than object-attributes [19]. We suggest that by using an elicitation structure following the laddering technique, we can enable users to generate requirements in an already relation-focused fashion, contributing to the eventual 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 requirements elicitation

For a chatbot to act as a substitute for a human interviewer in guiding users to generate quality requirements, the interface needs to incorporate a specific elicitation structure. Following this specific structure, users should be guided to communicate their requirements or needs on a level of detail that allows analysts to understand the essential aspects of the self-elicitation interview, e.g. to formulate user stories.

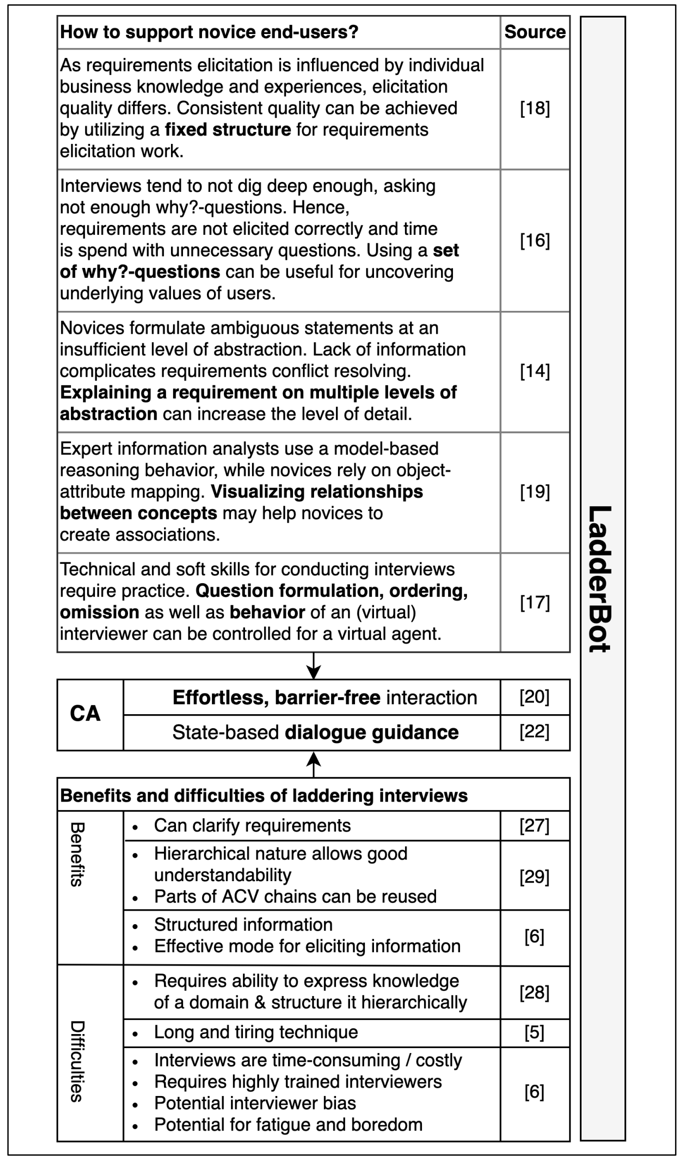

Fig. 1. Overview of the conceptual foundations of LadderBot

Laddering is an interview technique introduced in personality psychology to utilize a structured approach to data-gathering [6]. It 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 [26]. As such, the technique has become a tool for the means-end theory of the marketing domain. The means-end theory distinguishes three levels of abstraction of meaning that users obtain from purchase and/or consume [6]. These three levels are described as ACV chains: attributes – consequences – values [27]. *Attributes* as the least abstract level describe "concrete, physical or observable characteristics" of products. Despite the notion originally describing physical products, we my use the notion for digital products like software, too [28]. *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)* [28].

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?". The analysis process of laddering interviews is usually initiated with a content coding procedure. 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, the aggregate implication matrix is visualized 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 [28]–[30]).

C. The chatbot characteristics of form and function

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 [20]. 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 the 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 [20]. Multiple variants of chatbots have seen use over the years, which can be differentiated according to form and function [21]. 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 [22].

Despite a renewed research interest in chatbots, due to advances in artificial intelligence [23], the integration of CAs into RE remains spare. Derrick et al. investigated the effect of a simple scripted agent in facilitating group elicitation sessions with users [9] while other studies developed prototypes for frame-based agents in interview scenarios [24], [25]. While these studies evaluated the general applicability of CAs as facilitators of elicitation processes, to the best of our knowledge, no evaluation of CA-assisted requirements elicitation specifically targeting common quality criteria for elicitation techniques (effectiveness, efficiency, completeness) has been conducted [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, users are welcomed and receive a short explanation of the interface and the interview process. Afterwards, they are presented with a list of (seed) attributes. The list of attributes is generated in a pre-test session through triadic sorting[1]. LadderBot asks the user to identify the most relevant attribute. This attribute is subsequently used as seed for the first ACV chain. The following process is then repeated until participants constructed three chains for the three subjectively most relevant attributes. The user is asked an initial question to elicit the first consequence for the chosen attribute:

> *LadderBot: "You selected 'salary' as attribute most relevant to you when choosing an employer. Why is that?"*
> *User: "Because I want to make a lot of money of my education"*

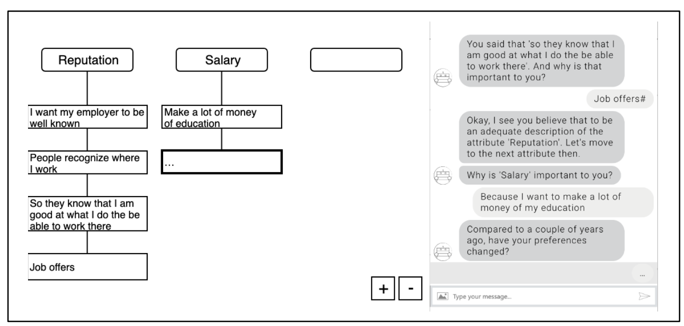

Fig. 2. Exemplary LadderBot interface

Rather than asking a default initial question, LadderBot integrates the specific attribute that users selected into question formulation. The line of questioning for consequences and values is the 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

[1] Participants receive three cards with company names and short descriptions. They are asked to identify ways in which two are similar but different from a third. The process is then repeated with other sets of three cards. The responses are used as attributes for the laddering interview [6]

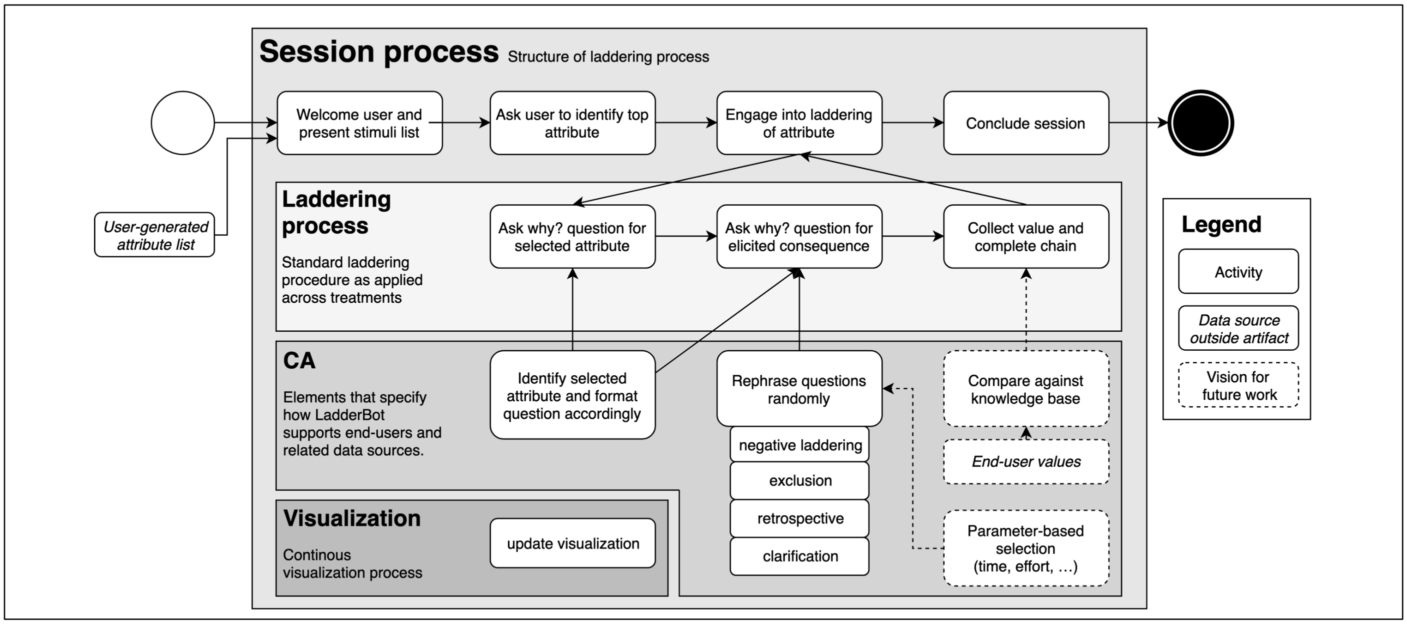

Fig. 3. Activity map of LadderBot

questions to help and guide the user. The techniques are identified 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 visualization of the current status of the interview on the left side is updated for each elicited consequence and the next question is asked. To end the elicitation for a specific attribute, or the interview in general, a human interviewer would need to identify if the user has described the value that they satisfy through an attribute (e.g. [29]). As the current iteration of LadderBot is not capable of recognizing whether a user has already described a final value, 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. The laddering questioning process is continued until the user performs such indication. In that case, LadderBot concludes the session.

As technological foundation of LadderBot, we use the Microsoft Bot Framework as well as LUIS as natural language processor. To visualize elicited ACV chains, we integrate the bot into a web application build on the VUE.js framework.

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 for evaluating elicitation techniques [31], [32] or applied the laddering technique as part of their experiment design [30].

The study is conducted with students from a large university in Germany in an experimental lab designed for conducting scientific studies. We choose *designing a recommender system for identifying suitable employers for a graduate position* as topic for the laddering self-interview. As such, we aim to identify requirements for an IS that supports finding an ideal entry position according to the graduates' selection criteria. Hence, students that are about to obtain their degree classify as novice users of the prospective system. To develop such a recommender system, it is mandatory to understand the attributes of employers that are most relevant for students, as well as the students' values explaining their priorities. Throughout the self-elicitation process, we expect participants to explore their priorities regarding attributes such as salary, working environment, training offerings, among others. Around 200 students in the final year of their respective program are invited as participants, randomly selected from a pool of potential participants.

The experimental study features a between-subject design with three treatments. Across treatments, participants are asked to conduct a self-elicitation of requirements for prospective employers. Treatments are characterized by the tool support available to participants. In treatment (1), participants use an established version of a "pencil-and-paper" laddering questionnaire [30]. However, a digital questionnaire is used, to increase comparability with the other treatments. In treatment (2), participants

TABLE I. Question rephrasing techniques

Technique	Description	Example
Negative laddering	Ask the user why they *do not* do something or *do not* want to feel a certain way	Why would you not apply for a job where overtime work is not tracked?
Exclusion	Ask the user to imagine a situation where an attribute or consequence does not exist	What would you base your decision on if you could not choose an employer with over 100 employees?
Retrospective	Ask the user to imagine their behavior in the past and compare it to now	Compared to a couple of years ago, have your preferences changed?
Clarification	Repeat a reply back to the user and ask for clarification	Please allow me to clarify. You said that 'You want to make a lot of money of your education'. So, why is that important to you?

use the LadderBot tool. In treatment (3), participants are given the choice to either use the laddering questionnaire or LadderBot. Participants may switch between the two methods during the experiment. By applying the presented treatments, we pursue two goals: firstly, to compare LadderBot with a traditional laddering approach that can be used to reach a high number of users: pencil-and-paper laddering; secondly, to evaluate participants preferences by including a treatment where participants themselves can select their preferred tool *in vivo*.

We evaluate the treatments using a combination of quantitative measurements. We rely on the established procedure for analyzing the results of the laddering interviews [6]. Abstractness and centrality are calculated on basis of an aggregate implication matrix, which represents direct and indirect linkages between attributes, consequences and values. Abstractness indicates whether constructs are predominantly at the beginning of a chain (means) or at their end (ends). Constructs become increasingly abstract from means to ends.

As such, it is a measure of importance in the means-ends structure [6]. Additionally, we use the number of direct / indirect links, the number of elicited consequences and values and time taken for comparing treatments [31]. Furthermore, subsequent to the treatments, we use a questionnaire to collect the participants perceptions regarding the following constructs on a 7-point Likert scale: Understandability, Learnability, Efficiency, Effectiveness and Enjoyment [32].

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: *Seed attribute ranking* - LadderBot initially requires the user to rank a list of attributes for the observed domain, that were previously generated through triadic sorting; *Elicitation guidance & assistance* - the user is supported through randomized rephrasing of questions based on an established guideline for interviewers. Answers of the users are used to formulate more lifelike questions; *Dynamic visualization* - Elicited attributes, consequences and values are visualized for the user and continuously updated throughout the interview process. The graphical representation of ACV chains may assist users in structuring their thoughts and uncovering new relations [19].

We propose an experimental study design to evaluate LadderBot against the established 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 requirements elicitation. Several comparisons of elicitation techniques have identified laddering as 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. For example, Jung (2014) applies a combination of laddering interviews and pencil-paper laddering, which is a questionnaire-based approach to laddering interviews for identifying user values of smartphone usage while investigating the means-end chain approach in the context of IT-user studies [29]. However, no considerations are presented for supporting the interviewer or automating the interview process entirely. This holds true for Tunnanen and Rossi (2004) as well, who propose a method for wide audience requirements elicitation that requires human conducted interviews for the generation of ACV chains [26]. 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 [33]. 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-elicit requirements creates the potential to come in contact with a broader 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.

We are currently working on finalizing the LadderBot artifact and setting up a pre-test for the initial evaluation of the tool and the generation of an (seed) attribute list for the main experiment. 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 [34]. For example, should a user diverge a specified amount from an average response time, the bot may provide additional assistance through question reformulation. As such, the bot would try to identify whether a user has problems with answering a question, and provide assistance similar to a human interviewer.

Furthermore, we plan on exploring adaptive interfaces that evolve as novice users build experience with LadderBot and with the process. Consequently, as novices become intermediates or experts, the tool should adapt its assistance and interface to the requirements of experienced users, see e.g. [33]. For example, the user may need less assistance through a chatbot, and rather use graphical input modalities more frequently. The adapted interface should then allow users to make use of the input modalities as they see fit.

References

[1] K. Villela et al., "Towards ubiquitous RE: A perspective on requirements engineering in the era of digital transformation," in Proceedings - 2018 IEEE 26th International Requirements Engineering Conference, RE 2018, 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] D. C. Derrick, A. Read, C. Nguyen, A. Callens, and G. J. De Vreede, "Automated group facilitation for gathering wide audience user requirements," in Proceedings of the Annual Hawaii International Conference on System Sciences, 2013, pp. 195–204.

[10] A. Rashid, D. Meder, J. Wiesenberger, and A. Behm, "Visual requirement specification in user participation," First Int. Work. Multimed. Requir. Eng. MeRE'06, 2006.

[11] B. Boehm, A. Egyed, J. Kwan, D. Port, A. Shah, and R. Madachy, "Using the WinWin spiral model: A case study," Comput. Pract., vol. 31, no. 7, pp. 33–44, 1998.

[12] P. Grünbacher and B. Boehm, "EasyWinWin: A Groupware-supported methodology for requirements negotiation," in Proceedings of the 8th European software engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of software engineering - ESEC/FSE-9, 2001, p. 320.

[13] M. Oriol et al., "FAME: Supporting continuous requirements elicitation by combining user feedback and monitoring," in Proceedings - 2018 IEEE 26th International Requirements Engineering Conference, RE 2018, 2018, vol. 0, pp. 217–227.

[14] A. Moitra et al., "Towards development of complete and conflict-free requirements," in Proceedings - 2018 IEEE 26th International Requirements Engineering Conference, RE 2018, 2018, pp. 286–296.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] L. C. Klopfenstein, S. Delpriori, S. Malatini, and A. Bogliolo, "The Rise of Bots: A Survey of Conversational Interfaces, Patterns, and Paradigms.," in Proceedings of the 2017 Conference on Designing Interactive Systems, 2017, pp. 555–565.

[21] T. Rietz, I. Benke, and A. Maedche, "The Impact of Anthropomorphic and Functional Chatbot Design Features in Enterprise Collaboration Systems on User Acceptance," in Proceedings of the 14th International Conference on Wirtschaftsinformatik, 2019, pp. 1656–1670.

[22] M. F. McTear, "Spoken dialogue technology: enabling the conversational user interface," ACM Comput. Surv., vol. 34, no. 1, pp. 90–169, 2002.

[23] U. Gnewuch, S. Morana, and A. Maedche, "Towards Designing Cooperative and Social Conversational Agents for Customer Service," in Proceedings of the International Conference on Information Systems (ICIS), 2017, pp. 1–13.

[24] 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.

[25] 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.

[26] T. Tuunanen and M. Rossi, "Engineering a method for wide audience requirements elicitation and integrating it to software development," in 37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the, 2004, vol. 00, no. C, pp. 1–10.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] C. R. Coulin, "A Situational Approach and Intelligent Tool for Collaborative Requirements Elicitation," Ph.D. University of Technology, Sydney, 2007.

[33] 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.

[34] 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.