Branching Preferences: Visualizing Non-linear Topic Progression in Conversational Recommender Systems

Lovis Bero Suchmann
lovis.suchmann@uni-due.de
University of Duisburg-Essen
Duisburg, Germany

Nicole Krämer
nicole.kraemer@uni-due.de
University of Duisburg-Essen
Duisburg, Germany

Jürgen Ziegler
juergen.ziegler@uni-due.de
University of Duisburg-Essen
Duisburg, Germany

ABSTRACT

Recent advances in AI allow complex, natural user–system dialogue flow in NLP-based conversational recommender systems (CRS). While this enables users to express complex intents to the system, its usual linear GUI representation as a chat log fails to account for two non-linear aspects of natural conversation: humans can switch between topics as customary; and, especially in decision-making contexts, topics discussed are structurally related. As early work, we motivate and present a GUI design approach that aims to exploit these phenomena for CRS by conveying topic progression, and discuss several design variants, their trade-offs, and open questions. Our approach aims to help users orientate while exploring and comparing multiple preference model variants and corresponding recommendations in complex, natural ways, also accounting for different explanation types. Such orientation could benefit users for achieving complex goals using CRS, like thoroughly-informed decision making, getting inspiration for novel consumable items, and exploring their own preferences.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Human-centered computing** → *Information visualization*; • **Computing methodologies** → Cognitive science.

KEYWORDS

Conversational Recommender Systems, Topic Progression, Mental Models, Branching Diagrams

ACM Reference Format:
Lovis Bero Suchmann, Nicole Krämer, and Jürgen Ziegler. 2023. Branching Preferences: Visualizing Non-linear Topic Progression in Conversational Recommender Systems. In *UMAP '23 Adjunct: Adjunct Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization (UMAP '23 Adjunct), June 26–29, 2023, Limassol, Cyprus.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3563359.3597380

1 INTRODUCTION

In the light of recent advances in AI and large language models (LLMs), conversational recommender systems (CRS) based on natural-language processing (NLP) can now facilitate increasingly complex dialogue flow between user and system. If a system is capable of handling complex and diverse forms of conversation, users are able to express equally complex intents and thought processes to the system directly, by simply verbalizing them in a way similar to natural communication between humans. One aspect of this beneficial complexity in natural conversation is the *non-linearity* of discussed topics: while conversational utterances—especially written ones—are ordered linearly (namely, chronologically), this does not necessarily hold for the topics addressed in a conversation. We identify at least two kinds of non-linearity regarding topic progression in complex natural conversation between humans:

- *Chronological non-linearity.* Rather than discussing one topic after the other in a strict order, humans are able to switch back and forth between topics as they see fit, remember which topics have been discussed, as well as return to a previous topic and expand on it.
- *Structural non-linearity.* Conversation topics are typically semantically and pragmatically related to each other, which holds particularly in joint decision-making contexts. These relations between topics form a non-linear structure, e.g., a hierarchy or network, that overlays the linear sequence of utterances in the dialog. In decision-making, topics can include, inter alia, naming preferences, comparing options, choosing between alternatives, finding commonalities, explaining rationales, or making plans. For instance, two conversation topics "comparing options for alternative A" and "comparing options for alternative B" can be regarded as having a prequisite relation to the topic "choosing between alternatives A and B." Such structural relations between topics help users organize the conversation and consider what to discuss next.

NLP-based CRS often serve as decision-making aids, helping to navigate a large set of options or items by filtering it according to users' preferences. While using CRS, users may benefit from exploiting chronological and structural non-linearity of topics to make well-informed decisions, especially for item domains that they are already willing to explore thoroughly, like computers, cars, or hotel rooms. Aside from well-informed decision making, users could also exploit non-linear topic progression to get inspiration for novel consumable items (e.g., in the movie, music, or scientific literature domains) or to discover their own preferences. To pursue these and other goals, users might like to investigate different *scenarios*—characterized by variations of the preference model—and compare the resulting recommendations, have recommendations and preference models explained, and more.

While LLMs and AI-based intent detection will likely further increase the capabilities of CRS to both understand and initiate

UMAP '23 Adjunct, June 26–29, 2023, Limassol, Cyprus

ACM ISBN 978-1-4503-9891-6/23/06…$15.00
https://doi.org/10.1145/3563359.3597380

non-linear topic progression on the content level, such conversations still have linear textual GUI representations, usually in the style of a digital chat application, or chat log, but with no further structural organization. This can pose a challenge to the user: In simple cases, like when an alternative scenario is investigated using a single user utterance that expresses the desired variation in the preference model (thus being conceptually equivalent to a *What If?* type explanation request), the conversation can immediately return to the previous topic after the system reply; thus a linear representation might well suffice. But in complex cases, for example when the user begins to investigate multiple scenarios in parallel, makes subsequent scenario-specific adjustments or explanation requests, and draws comparisons, both the number of topics and their non-linearity might increase strongly; this eventually makes it hard for the user to keep track of what topics have been explored and which topic an utterance corresponds to. This cognitive gulf of the user is present even if we assume perfect capabilities of the system regarding NLP and user intent detection, and likely gets even larger when the system makes errors.

In this paper, we aim to mitigate these limitations by suggesting a system that provides orientation to users regarding the following three questions: (1) What is the set of topics that has been explored, and what are its internal structure and relations? (2) How do these topics relate to individual utterances and to other elements like recommendations and preference model visualizations? (3) How can this information help me, as a user, to achieve my goals? Ideally, this orientation enables users to build up and navigate a coherent *mental model* of how both linear and non-linear information relate to each other that allows users to exploit complex topic progression, or reduces users' mental effort required for doing so.

To this end, we motivate and present a GUI design approach that aims to provide orientation to the user by (1) modeling the *topic structure* by conceptually organizing CRS-related topics based on the scenario they belong to and their function within that scenario, (2) employing a *branch-and-merge design paradigm* to relate topics to utterances and visualize topic progression, and (3) displaying *suggestions* for next user utterances in a specific way. We present an exemplary prototype and discuss possible design variants as early ongoing research. While the prototype is only one out of many ways to realize our design approach, we opted to present a concrete example before discussing variants, as such a prototype requires multiple interrelated design decisions.

The remaining paper is structured as follows. In Section 2, we present existing work from a number of related research strands. In Section 3, we introduce the exemplary GUI design prototype. In Section 4, we conclude by discussing possible design alternatives, open questions, and limitations.

2 RELATED WORK

In this section, we first review the state of research in CRS, focusing on concepts related to topic progression. We then look into examples from information visualization that indicate the feasibility of a *branch-and-merge design paradigm* for our purpose. Finally, we shortly address existing work on mental models that motivates our overall goal to help users integrate linear and non-linear information into a coherent mental representation.

2.1 Conversational Recommender Systems

CRS are recommender systems (RS) that support multi-turn interaction, usually towards the user's goal of deciding for a certain product or consumable item. While CRS with a *form-based interface* require users to enter text into forms and select options via classical GUI components, an *NLP-based interface* allows interaction via natural written or spoken conversation [14]. NLP-based CRS also qualify as a type of dialog system, or conversational agent. In particular, they fall into the category of goal-oriented conversational agents [2], as user and system work towards the goal of finding a suitable recommendation. While numerous distinctions have been made in the literature between (NLP-based) CRS and conversational search [16], as well as between dialog systems and chat bots [28], recent developments have resulted in complex systems that blend the borders between these concepts [29]. These developments also allow complex, natural conversation that can flexibly jump between preference elicitation, recommendation, opening the preference model [12], critiquing [8], and providing explanations, and could equally facilitate exploring multiple preference models.

Exploring alternative scenarios is conceptually similar to providing *What If?* type explanations known as counterfactuals [31, 35]. Explainability as a research area has recently gained a lot of interest. For RS, explanations can relate to the recommended items, the preference model, or the algorithms involved. Recent work also considers personalized explanations. An extensive taxonomy has been provided in [1]. Regarding CRS, existing work has investigated, e.g., explanatory needs of users in the hotel domain [13] as well as high-level user needs called meta-intents [21]. However, it is still unexplored how to organize multiple explanatory and other conversational needs in a CRS dialog without the user losing track of what they intended to do.

On the system side, user intents and dialog flow are typically managed in specific subsystems of the conversational agent in question [9, 20]. While conversation topics are related to user intents, topics usually span multiple utterances from both user and system. Recently, state-of-the-art LLMs have been employed for CRS via prompt engineering [11]. Here, while LLMs are also able to summarize a conversation and thus can serve as a tool to identify topics, topics are not used explicitly to steer the dialog flow.

Topic classification, in contrast to intent classification, has already been investigated in detail for conventional search queries (for an overview, see [4]), but not conversational agents; in addition, as conventional search is a single-turn process, existing topic taxonomies are usually content-related, while we also motivate differentiating topics based on their function in the recommendation process.

NLP-based CRS are usually presented in the style of a digital chat application. Recommendations and other aspects like visualizations of the preference model ("profile visualizations") can either be displayed within the chat log (e.g., [6]) or in a separate interface area (e.g., [13]). Existing work has looked into ways to integrate classical GUI elements into the chat log [22], but to our knowledge GUI approaches for organizing the dialog itself do not exist yet, even in research on dialog systems outside the field of CRS.

2.2 Information Visualization

Following Chen [7], we regard information visualization as the research area "concerned with the design, development, and application of computer generated interactive graphical representations of information", aiming to "communicate complex ideas to its audience and inspire its users for new connections." Thus, our aim of visualizing the complexity of non-linear topic progression to help users orientate and explore new scenarios meets this definition. According to the author, it usually takes a creative process to find ways how abstract data without intrinsic spatial representation can be brought into an understandable visual form. In this spirit, we transfer a *branch-and-merge design paradigm* from existing domains to the novel application of visualizing topic progression in CRS.

Branching diagrams are used in several domains to organize linearly ordered elements into groups with chronological and structural non-linearity. One common use case is with version control systems such as Git[1] ("`git log --tree`"). Git commits and code branches are conceptually similar to utterances and topics in that each commit belongs to a single code branch, branches originate from previous commits in existing code branches, and work on code branches can occur concurrently, i.e. chronologically non-linearly. In public transportation, *strip maps* [3] visualize a single transit line as a straight line, but may still contain branching and merging paths if the line splits up for a part of the track. Genealogical graphs also use branching to visualize evolution over time [24], similar to how topics evolve during conversation. TRACTUS [33] is a software tool helping data scientists organize code blocks in a linearly ordered source file into a branching structure that reflects their thought processes during data exploration and hypothesis testing.

Branching is also used to visualize data flow, often in programming languages such as Max/MSP/Jitter [23] for multimedia or recently Rapsai [10] for machine learning, but also in the context of statistical testing [34]. Previous work has found that such interactive visualizations are helpful for domain novices, such as high-school students learning how to program embedded systems [5].

2.3 Mental Models

Depending on the research discipline, the term *mental model* denotes a number of different concepts. For this work, we draw on two kinds of mental models, which we dub mental models of *how things work* and mental models of *how things relate to each other*.

We use the first expression to refer to mental models as they are presented in the works of Norman (e.g., [26]). Following his definition, a mental model is a simplified mental representation of a system, formed by experience or instruction and used to predict future system behavior [26]. For RS, providing structural knowledge of their inner workings has proven beneficial for mental model soundness, which can enable users to achieve more satisfying results [17]. Other work has investigated preexisting mental models of RS [19, 25]; systems were perceived as more transparent and competent by users whose mental models were structured by the sequential steps of the algorithm, rather than by the involved concepts alone [19].

Our GUI design prototype (cf. Section 3) can implicitly help the user form a (first-kind) mental model of how the recommendation process works by conveying, e.g., that recommendations are always based on preference models and that preference models evolve over time.

The second kind of mental models dates from the works of Johnson-Laird (e.g., [15]). Such mental models represent spatial and abstract relations and allow reasoning [32]. These models are built up sequentially; thus if new relations can be integrated into a user's current mental model directly, the model is more likely to stay coherent [15]. Such mental models have also been applied to information visualization [30]. Existing work on RS has, e.g., looked into benefits of visualizing preference models via spatial arrangements of items (e.g., [18]) or users (e.g., [27]), but connections to this kind of mental models are rarely made.

As utterances in CRS appear sequentially, we suggest that topic visualizations, especially if updated gradually with each new utterance as in the case of our GUI design prototype (cf. Section 3), can help users form a coherent mental model of how topics, utterances, recommendations, and other aspects relate to each other. As these (second-kind) mental models allow reasoning, such visualizations could make it easier for users to reason about the explored items and conversation topics and make well-informed decisions.

3 GUI DESIGN PROTOTYPE

This section explains in detail the rationales behind our prototype design. For this demonstration, we chose the item domain of song playlists for a music streaming service. We also created an exemplary conversation that can be inspected in full in Fig. 1–3, while the complete prototype interface is shown in Fig. 4. Following our introductory questions, we first describe our initial model of *topic structure* that conceptually organizes CRS-related topics based on the scenario they belong to and the function they have in that scenario, followed by design decisions regarding how to convey the chronological and structural non-linearity of topic progression, and finally ways to convey system capabilities regarding these aspects to users. Section 4 discusses possible alternative designs, open questions, and limitations.

3.1 Modeling Topic Structure

Organizing CRS dialogue by a list of topics already allows conveying chronological non-linearity by visualizing for each utterance which (single) topic it is related to. As there is no obvious unique—or best—way to define what counts as a topic, we suggest our own initial model of *topic structure* that we deem suited to a general complex recommendation process. This model organizes conversation topics based on two dimensions: the *scenario* they belong to, and their *function* within that scenario.

We postulate a set of (at least) five coarse *functions* that an exchange within a CRS dialogue can serve: (P) eliciting a preference model, (R) generating a recommendation, (RX) explaining a recommendation, (V) opening/visualizing a preference model, (VX) explaining a preference model (based on its visualization). *Scenarios*, on the other hand, are characterized by different variations of a preference model: The *initial scenario* uses the preference model that is originally elicited at the beginning of the conversation (cf. Fig. 1). An *alternative scenario* is created if the user desires to modify the

[1]https://git-scm.com/ (accessed Apr 29, 2023)

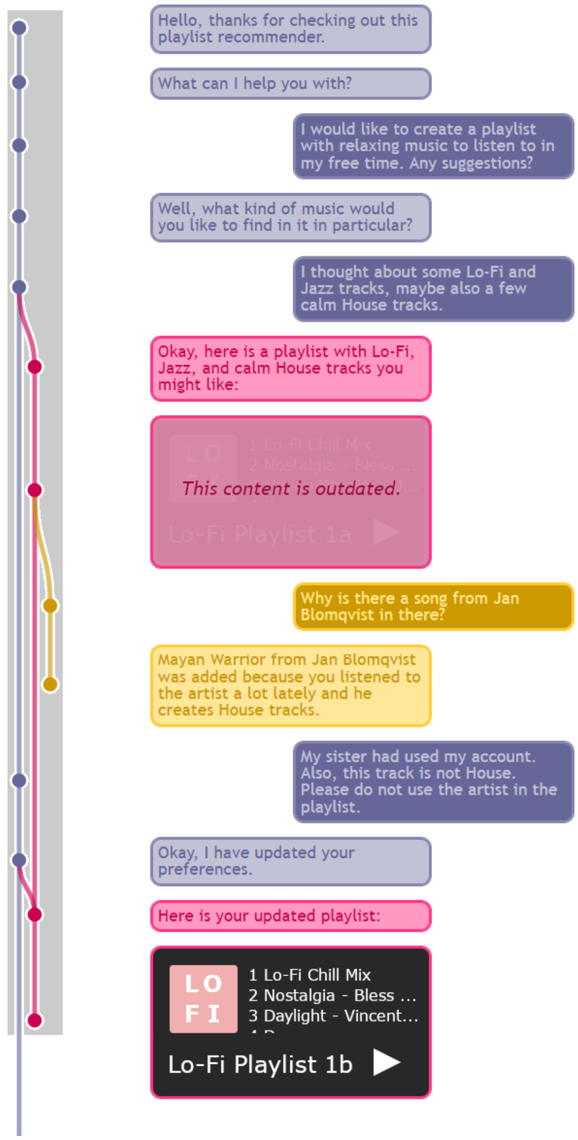

Figure 1: Exemplary conversation, part 1/3. Each branch represents a conversation topic. The grey background indicates that all three branches belong to one *scenario*. Within a scenario, branches are organized based on topic function: we use blue for preference elicitation (P), red for recommendation (R), and yellow for recommendation explanation (RX). The user can subsequently adjust preferences using information obtained in conversation (last four messages).

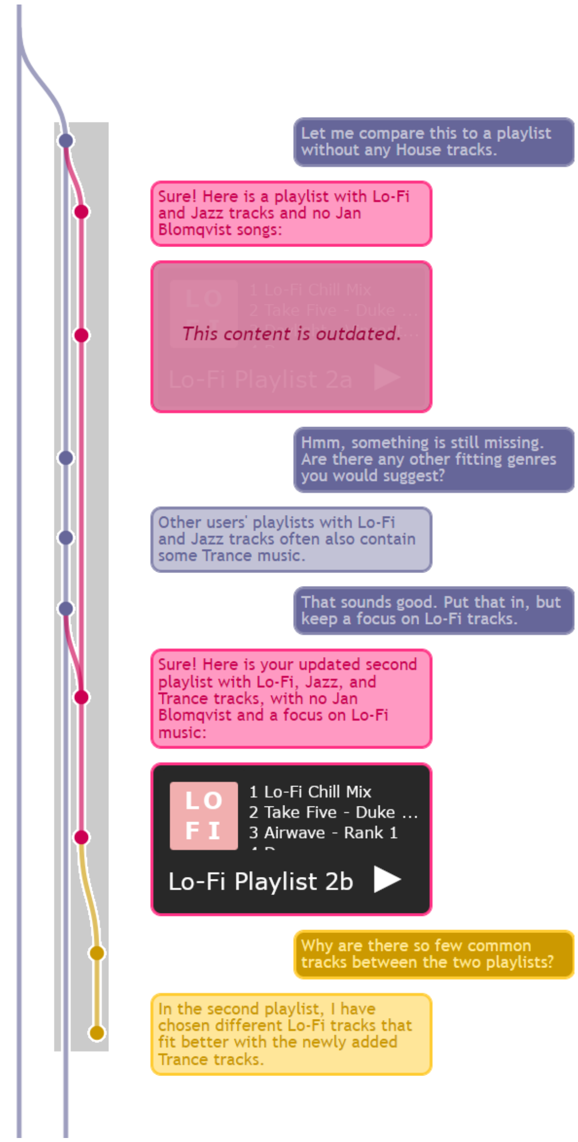

Figure 2: Exemplary conversation, part 2/3. Here, a second scenario (indicated by the grey background) is created as an *alternative* to the existing one when the user requests to compare the result of the existing elicitation branch to a modified version. Further adjustments and explanations are made for this scenario as well.

preference model of an existing scenario and compare recommendations and explanations between the two (cf. Fig. 2). A *merged scenario* is be created by combining the preference models of multiple existing scenarios (cf. Fig. 3). In a single scenario, utterances with all of the five functions can occur. We assume that each utterance is related to exactly one scenario and one function.

Our model assumes relations between topics in the sense of structural non-linearity. Each elicitation topic in (P) serves as the root of a scenario that the user might like to investigate. Thus, topics in (P) are related to each other, just like the corresponding scenarios are: The elicitation topic of an *alternative scenario* has a "modifies" relation to another elicitation topic; the elicitation topic of a *merged scenario* has a "combines" relation to a set of other elicitation topics. Within each scenario, topics are hierarchically structured based on their function: As each recommendation or visualization is based

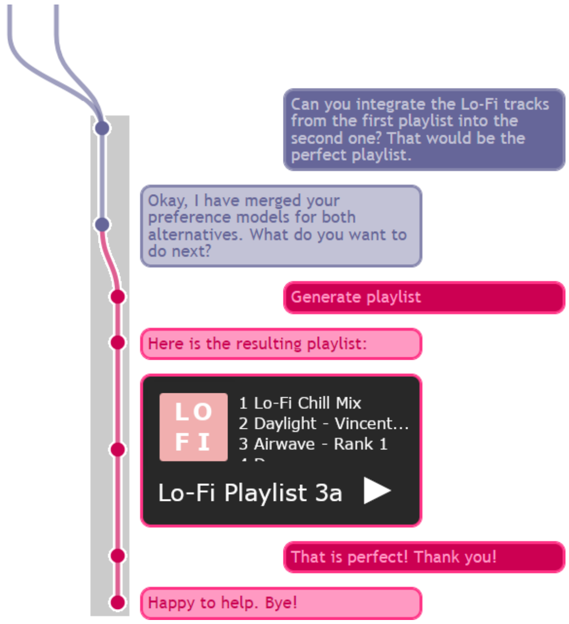

Figure 3: Exemplary conversation, part 3/3. Here, a third scenario (indicated by the grey background) is created by *merging* the final preference models of both previous scenarios into a combined preference model as requested by the user. This time, the user is satisfied with the final recommendation given in the recommendation branch of this scenario (last five messages).

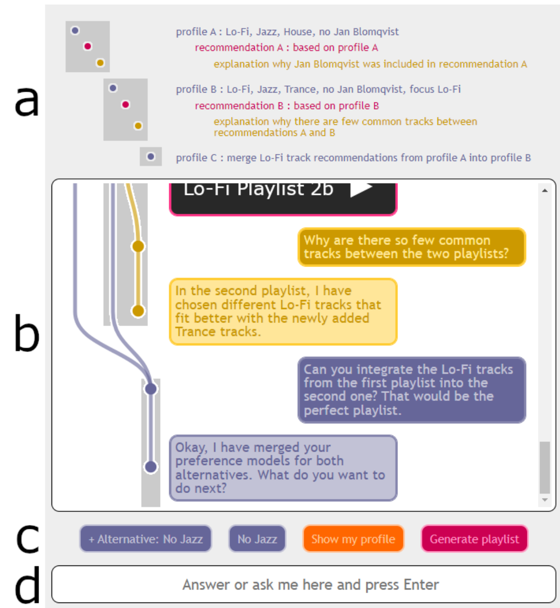

Figure 4: The full GUI prototype in an intermediate state of our exemplary conversation. (a) The *topic list* provides an overview and textual descriptions of all discussed topics. (b) The *chat log* allows to scroll through all utterances and the juxtaposed branching visualization. (c) Several *suggestions* for next user utterances give the user ideas how to continue the conversation; a "+" indicates the creation of a new scenario. (d) The *text field* enables the user to enter utterances.

on an already elicited preference model, we consider topics in (R) and (V) subordinate to the corresponding topic in (P). Also, as each topic in (RX) and (VX) explains a particular recommendation or visualization, these topics are subordinate to the corresponding topic in (R) and (V).

While this model is likely subject to future changes, additions, and refinements, it demonstrates how a CRS dialogue can be organized into a set of topics that have structurally non-linear relations.

3.2 Conveying Topic Progression in the GUI

We make use of several means to visualize how utterances relate to topics and show the non-linearity of topic progression in our GUI design prototype (cf. Fig. 4). The resulting system can convey to the user, e.g., which preference model a recommendation is based on, which recommendation or profile visualization an explanation refers to, or which scenarios have been explored. It might also help form a mental model of how the recommendation process works by conveying, e.g., that recommendations are always based on preference models and preference models evolve over time.

As a first means, in the chat log (cf. Fig. 4 b), we visualize discussed topics as a collection of vertical-line *branches* that appear to the left of the sequence of chat messages. The topic an utterance belongs to is indicated by a dot placed to the left of the utterance on the corresponding branch. Using this visualization of *chronological non-linearity*, the user can immediately see that the conversation has switched to a new topic if a new branch occurs, or to a previous topic if a new utterance is not on the same branch as previous ones and thus not directly connected to them by a vertical line.

Above the chat log, a list of all topics is displayed, each with a dot indicating the horizontal position of the corresponding branch in the chat log. This topic list (cf. Fig. 4 a) allows to convey the set of discussed topics, as well as a textual description of individual topics, to the user. The listed topics are grouped by scenario and subsequently ordered by first appearance.

We also convey the *structural non-linearity* of topics by visualizing which scenario they belong to and which function they have in that scenario: Branches whose topics belong to the same scenario are always placed next to each other. Boxes with a grey background are used to group the branches for each scenario as seen in Fig. 1–3. Similar boxes are used to group the dots in the topic list (cf. Fig. 4 a). We also color branches and topic list entries based on function: blue for (P), red for (R), orange for (V), yellow for (RX) and (VX). The function is also named for each topic list entry. Structural non-linearity is further conveyed by how the branches split and merge in the chat log. New subordinate branches (according our model of topic structure) always originate from the latest previous utterance of the topic they are subordinate to; alternative preference elicitation branches always originate from the latest previous utterance of the elicitation branch which they modify; elicitation branches that merge the preferences of multiple existing elicitation branches always originate from multiple utterances, one per involved branch. The topic list does not visualize how elicitation branches (and thus scenarios) relate to each other, but only shows subordination relations, by indenting subordinate topics.

3.3 Conveying System Capabilities in the GUI

To give the user ideas how to continue the conversation and reduce mental load, a number of suggested next user utterances is displayed below the chat log (cf. Fig. 4 c). Clicking one of these has the same effect as entering the utterance via the text field (cf. Fig. 4 d). Conveying system capabilities via "clickable preformulated user inputs" is nowadays a common approach in AI-based systems[2].

Due to the novelty of our approach, we consider it necessary to invest additional effort to convey to the user how a suggested utterance would affect the topic structure. To this end, we augment suggestions by (1) again applying the color mapping for topic functions and (2) showing a "+" icon if a suggestion will create a new elicitation branch (and thus scenario). For suggestions regarding elicitation, the difference between creating an alternative branch and adjusting an existing branch is emphasized ("Alternative: No Jazz" compared to "No Jazz"). With these augmentations, the user can anticipate when an utterance will switch the current branch or even create a new scenario. Our suggestions also convey structural non-linearity: e.g., the fact that explanatory suggestions (in yellow) are only shown in a recommendation or profile visualization branch conveys that explanations refer to such entities and not to the elicitation branch itself.

If an existing elicitation branch is adjusted, recommendations and profile visualizations subordinate to the branch are no longer valid. This is conveyed by retroactively overlaying an invalid element with a text box stating "This content is outdated." (cf. Fig. 1–2). In each scenario, at most one recommendation and one profile visualization remain valid.

4 ALTERNATIVES AND LIMITATIONS

This section concludes our paper by addressing potentially feasible design alternatives and their trade-offs, open questions, and limitations of our approach. In future user studies, we aim to answer these questions and verify that our general approach supports users in building up and navigating a coherent *mental model* of how both linear and non-linear information relate to each other, helping them achieve complex goals when using CRS.

One limitation of our approach is its inherent complexity, which results from its aim to support users follow complex thought processes. To be beneficial to users, it is vital to keep a balance between reducing the complexity to not overwhelm users, and showing enough information that enables users to orientate. As it is not clear where this balance lies, the complexity of the branching visualization in the chat log could be varied. For instance, it might confuse users or require too much mental load to pay attention to both the scenarios (grey boxes) and the individual topics (branches) at the same time. This could be resolved either by only visualizing how the individual topic branches evolve, or by instead getting rid of the branching of topics within a scenario and only visualizing the branching and merging of (alternative and merged) scenarios overall. In the latter case, the color mapping for topic functions could still be applied to utterances within a scenario.

A branch-and-merge paradigm could alternatively be employed by moving new (alternative or merged) scenarios into their own chat logs, and connect these chat logs similar to how data flow is visualized in [23, 34]. This emphasizes that each scenario has its individual scope and reduces the complexity of the branching visualization in each chat log, as it now only communicates how the topic functions evolve. Also, this design variant always keeps in view how the corresponding preference models are related, but results in a larger interface footprint that is undesirable, e.g., on mobile devices. Besides, by having multiple chat logs, the linear ordering of all utterances is lost, which might make it more difficult for users to backtrack. If a branch-and-merge paradigm should generally prove too complex to understand, only a color mapping could be applied in the chat log. Then, by clicking a topic in the topic list, the chat log could be filtered to only show messages of the selected topic. We aim to compare paradigms and degrees of complexity in future work.

Another limitation of our approach is that the preference elicitation within a scenario is regarded as a single topic and thus still unstructured. It could be helpful to visually further organize this process into topics based on item features, but it might also be challenging for users to handle the increased number of topics shown in the GUI.

Existing research on CRS has investigated how to integrate classical GUI components and inline critiquing into the conversation [22]. This could be integrated with our approach, e.g., by providing toggle buttons to switch between alternative scenarios. However, it is an open question how to do so in a way beneficial to the user.

So far, we have implicitly assumed that the system is perfectly capable of detecting user intents and classifying conversation topics, which will not hold in a working system. Further interface elements and/or conversational abilities need to be employed to allow the user to intervene when the system has not understood or classified an utterance correctly. How this should be handled in detail is unclear and another question subject to future research.

[2]https://www.microsoft.com/en-us/haxtoolkit/pattern/g1-d-demonstrate-possible-system-inputs/ (accessed Apr 29, 2023)

REFERENCES

[1] Qurat Ul Ain, Mohamed Amine Chatti, Mouadh Guesmi, and Shoeb Joarder. 2022. A Multi-Dimensional Conceptualization Framework for Personalized Explanations in Recommender Systems.
[2] Merav Allouch, Amos Azaria, and Rina Azoulay. 2021. Conversational Agents: Goals, Technologies, Vision and Challenges. *Sensors* 21, 24 (Dec. 2021), 8448. https://doi.org/10.3390/s21248448
[3] Silvania Avelar and Lorenz Hurni. 2006. On the Design of Schematic Transport Maps. *Cartographica: The International Journal for Geographic Information and Geovisualization* 41, 3 (Sept. 2006), 217–228. https://doi.org/10.3138/A477-3202-7876-N514
[4] David J. Brenes, Daniel Gayo-Avello, and Kilian Pérez-González. 2009. Survey and Evaluation of Query Intent Detection Methods. In *Proceedings of the 2009 Workshop on Web Search Click Data*. ACM, Barcelona Spain, 1–7. https://doi.org/10.1145/1507509.1507510
[5] Anke Brocker, René Schäfer, Christian Remy, Simon Voelker, and Jan Borchers. 2023. Flowboard: How Seamless, Live, Flow-Based Programming Impacts Learning to Code for Embedded Electronics. *ACM Transactions on Computer-Human Interaction* 30, 1 (Feb. 2023), 1–36. https://doi.org/10.1145/3533015
[6] Wanling Cai, Yucheng Jin, Xianglin Zhao, and Li Chen. 2023. "Listen to Music, Listen to Yourself": Design of a Conversational Agent to Support Self-Awareness While Listening to Music. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, Hamburg Germany, 1–19. https://doi.org/10.1145/3544548.3581427

[7] Chaomei Chen. 2010. Information Visualization. *Wiley Interdisciplinary Reviews: Computational Statistics* 2, 4 (July 2010), 387–403. https://doi.org/10.1002/wics.89

[8] Li Chen and Pearl Pu. 2012. Critiquing-Based Recommenders: Survey and Emerging Trends. *User Modeling and User-Adapted Interaction* 22, 1-2 (April 2012), 125–150. https://doi.org/10.1007/s11257-011-9108-6

[9] Yinpei Dai, Huihua Yu, Yixuan Jiang, Chengguang Tang, Yongbin Li, and Jian Sun. 2020. A Survey on Dialog Management: Recent Advances and Challenges. (2020). https://doi.org/10.48550/ARXIV.2005.02233

[10] Ruofei Du, Na Li, Jing Jin, Michelle Carney, Scott Miles, Maria Kleiner, Xiuxiu Yuan, Yinda Zhang, Anuva Kulkarni, Xingyu Liu, Ahmed Sabie, Sergio Orts-Escolano, Abhishek Kar, Ping Yu, Ram Iyengar, Adarsh Kowdle, and Alex Olwal. 2023. Rapsai: Accelerating Machine Learning Prototyping of Multimedia Applications through Visual Programming. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, Hamburg Germany, 1–23. https://doi.org/10.1145/3544548.3581338

[11] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-REC: Towards Interactive and Explainable LLMs-Augmented Recommender System. (2023). https://doi.org/10.48550/ARXIV.2303.14524

[12] Mouadh Guesmi, Mohamed Amine Chatti, Yiqi Sun, Shadi Zumor, Fangzheng Ji, Arham Muslim, Laura Vorgerd, and Shoeb Ahmed Joarder. 2021. Open, Scrutable and Explainable Interest Models for Transparent Recommendation.. In *IUI Workshops*.

[13] Diana C. Hernandez-Bocanegra and Jürgen Ziegler. 2023. Explaining Recommendations through Conversations: Dialog Model and the Effects of Interface Type and Degree of Interactivity. *ACM Transactions on Interactive Intelligent Systems* 13, 2 (June 2023), 1–47. https://doi.org/10.1145/3579541

[14] Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2022. A Survey on Conversational Recommender Systems. *Comput. Surveys* 54, 5 (June 2022), 1–36. https://doi.org/10.1145/3453154

[15] Philip N Johnson-Laird. 1980. Mental Models in Cognitive Science. *Cognitive science* 4, 1 (1980), 71–115.

[16] Tina Klüwer. 2011. From Chatbots to Dialog Systems. In *Conversational Agents and Natural Language Interaction: Techniques and Effective Practices*. IGI Global, 1–22.

[17] Todd Kulesza, Simone Stumpf, Margaret Burnett, and Irwin Kwan. 2012. Tell Me More?: The Effects of Mental Model Soundness on Personalizing an Intelligent Agent. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Austin Texas USA, 1–10. https://doi.org/10.1145/2207676.2207678

[18] Johannes Kunkel, Benedikt Loepp, and Jürgen Ziegler. 2017. A 3D Item Space Visualization for Presenting and Manipulating User Preferences in Collaborative Filtering. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*. ACM, Limassol Cyprus, 3–15. https://doi.org/10.1145/3025171.3025189

[19] Johannes Kunkel, Thao Ngo, Jürgen Ziegler, and Nicole Krämer. 2021. Identifying Group-Specific Mental Models of Recommender Systems: A Novel Quantitative Approach. In *Human-Computer Interaction – INTERACT 2021*, Carmelo Ardito, Rosa Lanzilotti, Alessio Malizia, Helen Petrie, Antonio Piccinno, Giuseppe Desolda, and Kori Inkpen (Eds.). Vol. 12935. Springer International Publishing, Cham, 383–404. https://doi.org/10.1007/978-3-030-85610-6_23

[20] Jiao Liu, Yanling Li, and Min Lin. 2019. Review of Intent Detection Methods in the Human-Machine Dialogue System. *Journal of Physics: Conference Series* 1267, 1 (July 2019), 012059. https://doi.org/10.1088/1742-6596/1267/1/012059

[21] Yuan Ma, Tim Donkers, Timm Kleemann, and Jürgen Ziegler. 2023. An Instrument for Measuring Users' Meta-Intents. In *Proceedings of the 2023 Conference on Human Information Interaction and Retrieval*. ACM, Austin TX USA, 290–302. https://doi.org/10.1145/3576840.3578317

[22] Yuan Ma, Timm Kleemann, and Jürgen Ziegler. 2021. Mixed-Modality Interaction in Conversational Recommender Systems.. In *IntRS@ RecSys*. 21–37.

[23] Vincent J Manzo. 2016. *Max/MSP/Jitter for Music: A Practical Guide to Developing Interactive Music Systems for Education and More*. Oxford University Press.

[24] M.J. McGuffin and R. Balakrishnan. 2005. Interactive Visualization of Genealogical Graphs. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005*. IEEE, Minneapolis, MN, USA, 16–23. https://doi.org/10.1109/INFVIS.2005.1532124

[25] Thao Ngo, Johannes Kunkel, and Jürgen Ziegler. 2020. Exploring Mental Models for Transparent and Controllable Recommender Systems: A Qualitative Study. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*. ACM, Genoa Italy, 183–191. https://doi.org/10.1145/3340631.3394841

[26] Don Norman. 2013. *The Design of Everyday Things: Revised and Expanded Edition*. Basic books.

[27] John O'Donovan, Barry Smyth, Brynjar Gretarsson, Svetlin Bostandjiev, and Tobias Höllerer. 2008. PeerChooser: Visual Interactive Recommendation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Florence Italy, 1085–1088. https://doi.org/10.1145/1357054.1357222

[28] Diana Perez-Marin and Ismael Pascual-Nieto (Eds.). 2011. *Conversational Agents and Natural Language Interaction: Techniques and Effective Practices*. IGI Global. https://doi.org/10.4018/978-1-60960-617-6

[29] Filip Radlinski and Nick Craswell. 2017. A Theoretical Framework for Conversational Search. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*. ACM, Oslo Norway, 117–126. https://doi.org/10.1145/3020165.3020183

[30] Guenther Schreder, Florian Windhager, Michael Smuc, and Eva Mayr. 2016. A Mental Models Perspective on Designing Information Visualizations for Political Communication. *JeDEM - eJournal of eDemocracy and Open Government* 8, 3 (Dec. 2016), 80–99. https://doi.org/10.29379/jedem.v8i3.443

[31] Ruoxi Shang, K. J. Kevin Feng, and Chirag Shah. 2022. Why Am I Not Seeing It? Understanding Users' Needs for Counterfactual Explanations in Everyday Recommendations. In *2022 ACM Conference on Fairness, Accountability, and Transparency*. ACM, Seoul Republic of Korea, 1330–1340. https://doi.org/10.1145/3531146.3533189

[32] Jan Frederik Sima, Holger Schultheis, and Thomas Barkowsky. 2013. Differences between Spatial and Visual Mental Representations. *Frontiers in Psychology* 4 (2013). https://doi.org/10.3389/fpsyg.2013.00240

[33] Krishna Subramanian, Johannes Maas, and Jan Borchers. 2020. TRACTUS: Understanding and Supporting Source Code Experimentation in Hypothesis-Driven Data Science. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–12. https://doi.org/10.1145/3313831.3376764

[34] Krishna Subramanian, Johannes Maas, Michael Ellers, Chat Wacharamanotham, Simon Voelker, and Jan Borchers. 2018. StatWire: Visual Flow-based Statistical Programming. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, Montreal QC Canada, 1–6. https://doi.org/10.1145/3170427.3188528

[35] Kun Xiong, Wenwen Ye, Xu Chen, Yongfeng Zhang, Wayne Xin Zhao, Binbin Hu, Zhiqiang Zhang, and Jun Zhou. 2021. Counterfactual Review-based Recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. ACM, Virtual Event Queensland Australia, 2231–2240. https://doi.org/10.1145/3459637.3482244