March 19, 2019

Conversational hierarchy for interaction with virtual assistant

Alex Jacobson

Jens Nagel

Ye-Jeong Kim

Conversational hierarchy for interaction with virtual assistant

ABSTRACT

User interfaces for conversations, e.g., with virtual assistants, typically model conversation as a back-and-forth exchange between participants, e.g., without regard to the type or topic of conversation. Under this user interface, a filter, refinement, or query by the user is added as a new inline block or a full-page refresh, making the user interface cumbersome to use.

This disclosure describes techniques that model conversation as a hierarchy, e.g., with major and minor conversational turns. The user interface is designed around this hierarchy, and a flow between the front-end and the back-end of conversations informs user interactions and client-side page interactions.

KEYWORDS

- conversational hierarchy
- conversational turn
- conversation modeling
- conversational UI
- chat UI
- virtual assistant
- bot interaction
- page refresh

BACKGROUND

User interfaces for conversations, e.g., with virtual assistants or bots, typically model conversation as a back-and-forth exchange (as a chat or messaging user interface) between participants without regard to the type or topic of conversation. In such a user interface, each

user query and the corresponding virtual assistant response is shown as a different conversational turn. This results in many elements of the UI being fully reloaded. Further, such UI has a dense history of queries that is difficult to read. For example, filtering a list of items 3 times results in the chat history having the same list 3 times. Alternately, a filter, refinement, or query by the user is added as a new inline block or a full-page refresh, making the user interface cumbersome to use. A full-page refresh also results in high latency, as all contents of the screen need to be reloaded for each additional query. This includes structural information around a task, e.g., the preamble, user selection, transcription, and response. For multi-step queries, this makes each refinement step very slow. For example, making a reservation may require full screen load to input time, location, date, number of people, and other requirements.

DESCRIPTION

This disclosure describes techniques that model a conversation as a hierarchy, e.g., with major and minor conversational turns. The user interface is designed based on the hierarchy, and a flow between the front-end and the back-end of conversations informs user interactions and client-side page interactions.

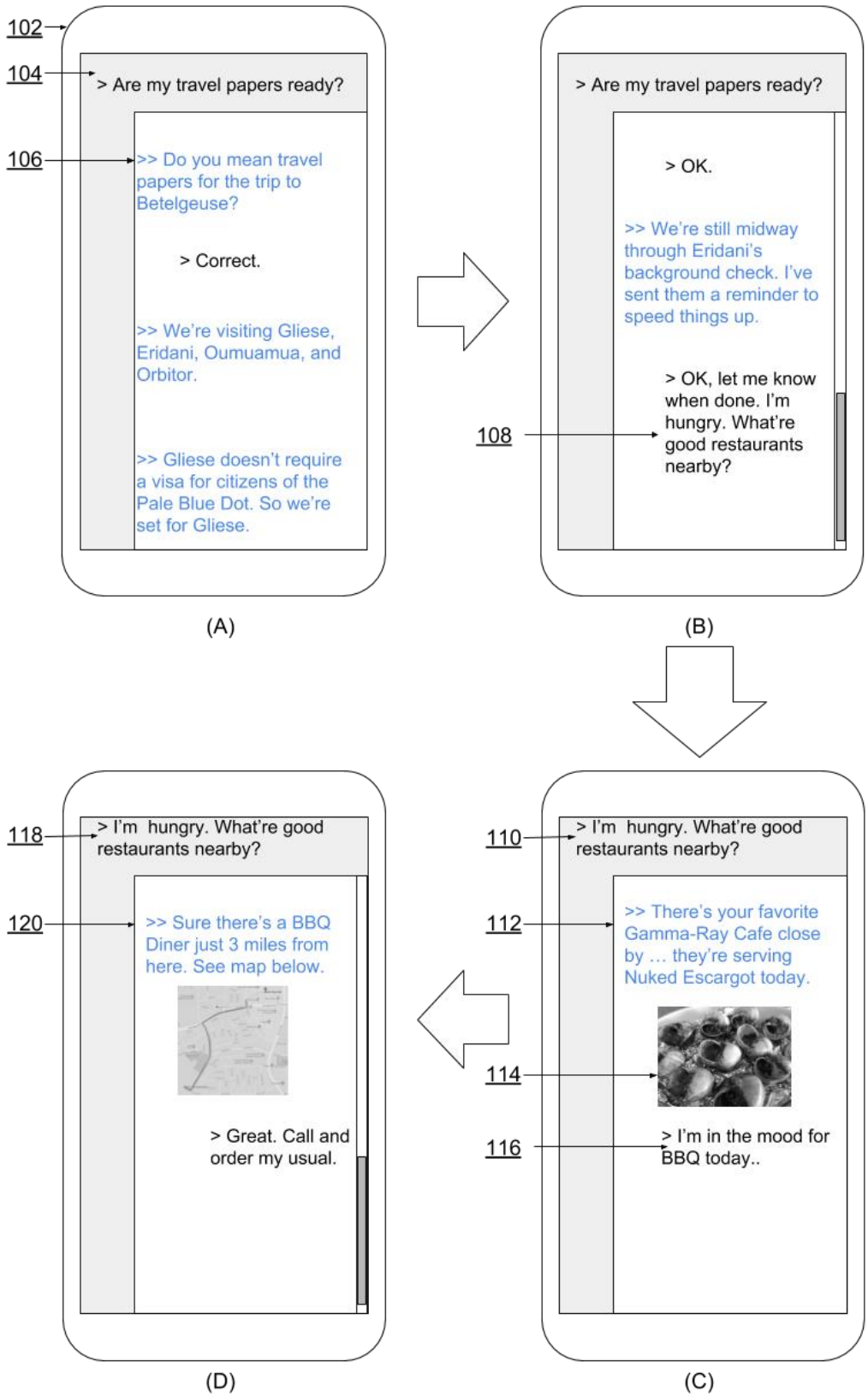

Fig. 1: Conversational hierarchy for interaction with virtual assistant

The modeling conversation as a hierarchy is illustrated by the example of Fig. 1. A user is conducting a conversation with a virtual assistant on a device with a screen (102), e.g., smartphone, tablet, laptop, etc. The virtual assistant may be an on-device application, a server-based application, or a combination. The virtual assistant renders answers to user queries in a displayed user interface, as shown in in Fig. 1. With user permission, the virtual assistant interacts with a server to retrieve answers to user queries and render the user interface. The user interface may include multiple elements that are utilized to display user queries and corresponding responses. The multiple elements may be displayed as part of a page, e.g., a web page.

In Fig. 1A, the conversation starts with a travel enquiry ("Are my travel papers ready?") from the user (104), which is detected as a major turn in the conversation. A major turn causes a full-page refresh, and subsequent lines of conversation that belong to this major turn are organized within a frame (106). As the conversation proceeds along this major turn, the contents of the frame are updated (Fig. 1B) and no full-page refresh is performed. The major turn in the conversation, e.g., the original travel enquiry, remains displayed as the conversation undergoes minor turns.

At a point in the conversation, the conversation deviates from travel to another topic. For example, in Fig. 1B, the user issues a restaurant-related query (108) - "I'm hungry. What're good restaurants nearby?" This query is detected as a new major turn in the conversation. The new major turn triggers a full-page refresh (Fig. 1C). Minor turns within this new major turn are organized within another frame (112). The full-page refresh includes display of the user request (110) and responses from the server, e.g., images and follow-on recommended content (114). The user requests a filter for restaurants serving barbeque (116) - "I'm in the mood for BBQ

today" - which is detected as a minor conversational turn. A selective, rather than full-page, refresh is performed, such that the selective refresh provides details (120), e.g., map and recommended content, pertaining to barbeque restaurants. High-level hierarchical information around the originating user query, page framing, and user selection remains static (118).

By selectively updating the page at major conversational turns, the techniques of this disclosure provide the following benefits:

- Latency is reduced during client rendering and data-fetch sizing.
- The number of server calls is reduced.
- Client and server have a clear organizational handoff.
- Expectations around the elements to update and expect refreshes for are based upon user interaction. This informs both UI content selection, e.g., on-screen taps, as well as follow-on voice queries that are used to update screen content.
- History bucketing becomes much clearer in organization. By hiding the full history of queries, e.g., in minor conversational turns, much of the noise of confirmation dialogues, disambiguation, and refinement is hidden from the conversation.
- By modeling major conversational turns as history-relevant, minor conversational turns automatically tuck into conversational history, following the same hierarchy.

Alternatively, at each response the client and the server can try to determine on the fly the contents to the user interface that need updating upon data fetch.

CONCLUSION

This disclosure describes techniques that model conversation as a hierarchy, e.g., with major and minor conversational turns. A user interface, e.g., a displayed UI for a virtual

assistant, is designed around this hierarchy. The flow between the front-end and the back-end of conversations informs user interactions and client-side page interactions.