Interaction Design Patterns of Web Chatbots

Chapter · June 2024

DOI: 10.1007/978-3-031-62362-2_22

2 authors:

Verena Traubinger
Chemnitz University of Technology
10 PUBLICATIONS **17** CITATIONS
SEE PROFILE

Martin Gaedke
Chemnitz University of Technology
298 PUBLICATIONS **3,588** CITATIONS
SEE PROFILE

Interaction Design Patterns of Web Chatbots

Verena Traubinger ✉[0000−0002−1786−672X] and Martin Gaedke[0000−0002−6729−2912]

Faculty of Computer Science, Chemnitz University of Technology, Germany
verena.traubinger@informatik.tu-chemnitz.de
martin.gaedke@informatik.tu-chemnitz.de

Abstract. Chatbots are often used in the web as an additional user interface which offers different modalities for users. Still, there is not yet an established catalogue of interaction patterns across different chatbot implementations, while this is the case for graphical user interfaces. Such a catalogue will give web engineers an orientation in developing chatbots, which will in turn lead to users more easily recognizing common functionalities of chatbots. This short paper is mapping known chatbot functionalities to already established interaction design patterns from graphical user interfaces. For this, these frameworks were assessed on their relevance towards web interfaces before systematically mapping them onto each other by regarding their similarities in the categories *Situation*, *Intention* and *Implementation*. From the result, we discuss similarities and differences which are influencing the specific use of interaction design patterns in web chatbots and formulate the ensuing 12 chatbot interaction patterns in a catalogue for web engineers.

Keywords: Web interfaces · Human-Computer-Interaction (HCI) · Web Engineering · interaction design patterns · User Interfaces.

1 Introduction and Related Work

The design of graphical user interfaces (GUIs) in the web has a history of change and evolution [4] until a common ground was found and guidelines were established. A pattern language, as introduced by [1] was the basis for [7] to describe interaction design patterns, which are often already used by designers as best practices, like Input Prompts, Pagination or Error Messages. To become a pattern, single visual elements or actions like buttons, toolbars or hovering have to be used with a specific intention and context by the users. The authors concentrate on web and mobile interfaces, usually with a large weight on graphical representations of content. Other interfaces are explicitly excluded by the authors, among them chatbots and other conversational designs, as they are both an emerging technology.

Chatbots are often used in web environments, like e-commerce and customer service. They are a form of Conversational User Interface (CUI) with a visual-centric style, which means that GUI elements are combined with natural language processing [5]. To the best of our knowledge, only two publications regard

how chatbots and humans interact on a graphical basis. As the authors use different terms, we refer to these interactions as *chatbot functionalities*. The first publication lists "Rich Interactions", which replace or enhance text-based conversations which would otherwise be very long for completing complex tasks [6]. In their approach, [8] systematically analyzed chatbots and found that "Sign Classes" (individual interaction elements) and "Strategies" were used to convey possible commands, actions and main functionalities. Both publications have some overlaps in their specified chatbot functionalities, while in combination they still present a broad view on them.

While natural language based interactions in chatbots are already researched on in several aspects, the analysis of graphical interaction elements is thus rather limited. We aim to fill this gap by mapping chatbot functionalities to well known interaction patterns to provide a common basis for the engineering of web chatbots. Our contributions in this short paper are:

1. presenting established interaction design patterns and chatbot functionalities
2. selecting these patterns and functionalities on their applicability for the web
3. mapping them with regard to shared constructed and described categories
4. building a catalogue of 12 chatbot interaction patterns

2 Methodology

For the mapping process, we relied on the presented works from [6,7,8]. First, they were preprocessed on their suitability for web interfaces by building inclusion and exclusion criteria, followed by a comparison of different categories for each interaction design pattern, Rich Interaction, Sign Class and Strategy which resulted in the final mapping presented in section 3. The inclusion and exclusion criteria, aim to rule out any interaction design patterns not related to chatbot or web interfaces, as well as chatbot functionalities restricted to single platforms or not applicable for web interfaces. After applying these criteria, we had a set of 45 interaction patterns and 26 chatbot functionalities which were further processed. The complete list of both criteria and the list of all mapping elements can be found in the appendix which is published in a repository[1]. For a systematic mapping, we needed a common basis of categories with which we could compare interaction design patterns and chatbot functionalities. As [7] used the most elaborated structure, their categories were generalized in the following way and applied on the chatbot functionalities: *Situation* (In which context is it used?), *Intention* (How does it impact the user?), *Implementation* (How is it implemented?) These categories were described by hand for each of the chatbot functionalities. For an easier comparison, ChatGPT was used to summarize the descriptions from [7] which were afterwards verified by one of the authors. In the mapping process, we compared the categories of the chatbot functionalities and the interaction design patterns and noted down a match, if we could find similarities in these categories.

[1] https://github.com/vertr/Chatbot-Interaction-Patterns

For the mapping process, we want to address some limitations which might be threats to its validity. An internal threat could be chatbot functionalities, which are only used on specific platforms. We tried to avoid this bias by using inclusion and exclusion criteria with a focus on web environments. An external threat is possible due to the methodology where the final patterns were only mapped theoretically but not verified with real-world examples. This threat was mitigated by choosing publications which include diverse sets of interaction elements and different chatbot implementations.

3 Mapping Results

Here, we present the results of mapping 45 interaction patterns to 26 chatbot functionalities. Table 1 includes only conclusive matches from chatbot functionalities to interaction design patterns, additional matches with only indirect dependencies can be found in the appendix. For readability in Table 1, the names of all mapping elements were abbreviated in the following way.
Sign Classes (C) [8]: Simple Message (*C1*), Simple Image (*C2*), Suggestions or quick replies (*C3*), Card (*C4*), Carousel (*C5*), Persistent Menu (*C6*).
Strategies (S) [8]: Showing the main feature on the first message (*S1*), Guiding the user through a short tutorial during first messages (*S2*), Suggesting the next possible set of actions to the user (*S3*), Having a persistent menu with main features (*S4*), Sending the main menu with main features as message (*S5*), Having a list of available commands (*S6*), Offering contextual help about a feature (*S7*), Showing the main menu or the most frequent features when the user asks for help (*S8*), Showing the main menu or main features when user says something the chatbot cannot understand (*S9*).
Rich Interactions (R) [6]: Files (*R1*), Audio (*R2*), Video (*R3*), Images (*R4*), Buttons (*R5*), Templates (*R6*), Links (*R7*), Emojis (*R8*), Persistent menus (*R9*), Typing indications (*R10*), Webviews (*R11*).
The abbreviations **SE**, **HS**, **TS**, **Cs**, **Cl**, **BG**, **AP**, **Pw**, **LI** respectively stand for the chatbot functionalities Settings Editor, Help Systems, Titled Sections, Cards, Carousel, Button Groups, Action Panel, Preview, Spinner and Loading Indicators.

From the 45 interaction design patterns which were included, 9 could be mapped conclusively to at least one of the 26 chatbot functionalities. The *Settings Editor (SE)* can be matched to chatbot functionalities C6 and R9. *Help Systems (HS)* matched with 9 functionalities (S1-9). The next pattern of *Titled Sections (TS)* is also mapped to 9 functionalities (C1, C2, C4, R1-R4, R6 and R11). *Cards (Cs)* are mapped to the functionalities C4 and R6, while the *Carousel (Cl)* pattern has one exact match, even on the naming, with C5. The sixth pattern of *Button Groups (BG)* is matched to 2 functionalities C3 and R5. The *Action Panel (AP)* was mapped to 3 patterns, S5-7, while the last two patterns were matched each with one functionality. *Preview (Pw)* to R11 and *Spinners and Loading Indicators (LI)* to R10. Two functionalities were not mapped conclusively to any pattern: *Links (R7)* and *Emojis (R8)*.

Table 1. Results of mapping chatbot functionalities to interaction design patterns.

	SE	**HS**	**TS**	**Cs**	**Cl**	**BG**	**AP**	**Pw**	**LI**
C1	.	.	**X**	.	.	.	.	.	.
C2	.	.	**X**	.	.	.	.	.	.
C3	.	.	.	.	.	**X**	.	.	.
C4	.	.	**X**	**X**	.	.	.	.	.
C5	.	.	.	.	**X**	.	.	.	.
C6	**X**	.	.	.	.	.	.	.	.
S1	.	**X**	.	.	.	.	.	.	.
S2	.	**X**	.	.	.	.	.	.	.
S3	.	**X**	.	.	.	.	.	.	.
S4	.	**X**	.	.	.	.	.	.	.
S5	.	**X**	.	.	.	.	**X**	.	.
S6	.	**X**	.	.	.	.	**X**	.	.
S7	.	**X**	.	.	.	.	**X**	.	.
S8	.	**X**	.	.	.	.	.	.	.
S9	.	**X**	.	.	.	.	.	.	.
R1	.	.	**X**	.	.	.	.	.	.
R2	.	.	**X**	.	.	.	.	.	.
R3	.	.	**X**	.	.	.	.	.	.
R4	.	.	**X**	.	.	.	.	.	.
R5	.	.	.	.	.	**X**	.	.	.
R6	.	.	**X**	**X**	.	.	.	.	.
R7	.	.	.	.	.	.	.	.	.
R8	.	.	.	.	.	.	.	.	.
R9	**X**	.	.	.	.	.	.	.	.
R10	.	.	.	.	.	.	.	.	**X**
R11	.	.	**X**	.	.	.	.	**X**	.

"**X**" indicates a conclusive match, "." indicates no match.

4 Discussion

In this section we debate the findings of our mapping process and which influence they have on formulating the chatbot interaction patterns catalogue in section 5. The abbreviations for chatbot functionalities and interaction design patterns from Table 1 are again used.

Direct matches. Several Patterns could be mapped directly to even similar named chatbot functionalities. While **Cards (Cs)** present information visually, often in combination with a link, the visualisation only slightly differs in a chatbot or can be customized (C4, R6) [6]. A **Carousel (Cl)** describes in both graphical and chatbot interfaces a horizontal menu of items (most often Cards) which can be navigated either by scrolling or by using arrow buttons (C5).

Matches with chatbot specific equivalents. The following patterns have matches where the specific implementation is different in chatbots, but similar in the categories *Situation* and *Intention*. The **Settings Editor (SE)** and persistent menus (C6 and R9) match, even though some specific interactions differ. While the Settings Editor is mostly used for profile and preference settings, the persistent menu usually offers information on the most important functionalities and sometimes includes a FAQ to the chatbot itself. The pattern of **Titled Sections (TS)** relates to dividing the interface by using its layout. In chatbots, this is usually done by separating the conversation into respective speech bubbles from the chatbot and the user. While the conversational part is mostly textual (C1), it can also include different media functionalities (C2, C4, R1-4, R6, R11). The **Action Panel (AP)** is a dynamic part in the graphical interface, dependent on the current context. As chatbots, of course, also react dynamically on requests from the user, we mapped this functionality to main menus which are part of strategies S5-7. The pattern **Preview (Pw)** shows similarities to the webview (R11). While the Preview is used for example in webshops to offer different colour and material options, the webview fetches data from other homepages like routes in online maps or product information. In GUIs, **Spinners and loading indicators (LI)** visualize to users that processes are running in the background, even though the interface itself does not change. While chatbots are able to reply instantly, [6] mention this as possibly unsettling, as human communication partner would need to take time to think and type. As a result, chatbots sometimes insert typing indications (R10), which users recognize from messengers, and delay their answer for a few seconds to let the user feel more comfortable. **Button Groups (BG)** match to chatbot functionalities which differ from their use in GUIs: Either added to a card to represent actions, or as a selection of possible answers for the user beneath a chatbot message [3][2]. These functionalities are either called Buttons (R5) for both possible implementations [6], or only refer to the second one as is seen in names as Suggestions or Quick replies (C3) in [8]. Quick replies can be the only way to communicate for the users, if chatbots do not allow free text input. One criticism by [8] on them is that they are usually disappearing after being clicked and reappear as if the user wrote a message with the same content. This makes it hard in retrospect to see which message came from the users themselves and which one was prompted by the system. The authors also noticed that it can make a difference for follow-up reactions of the chatbot if the Quick Reply is chosen or if the same command is typed in by hand.

Matches to chatbot functionality groups. The pattern **Help Systems (HS)** can be directly mapped to all strategies of [8][3], which led us to make further distinctions for a more precise matching. We grouped similar chatbot functionalities to show where different forms of Help Systems are overlapping:

[2] While the interaction design pattern groups similar actions, different answers in chatbots usually lead to diverging conversations.

[3] The authors analyze how chatbots explain their functionalities and thus focus inherently on Help Systems.

Functionality Introduction where main functionalities or a tutorial are provided in the first chat messages (S1, S2); *Functionality Menus* which are either always present or sent as a reaction on a message from the user (S4-7); *Requested Help* where specific information is provided on an explicit request (S5, S6-9); and *Breakdown avoidance* where a conversation breakdown [2] is avoided by the chatbot offering users possible options or providing functionality menus (S3, S5). Some of the mapped chatbot functionalities relate to more than one of these groups, as the same functionality can be used in different situations.

Further findings. Two chatbot functionalities could not be mapped to interaction patterns: Links (R7) and Emojis (R8). While links are universal in web interfaces, they are not feasible on their own as a pattern an can rather be considered as a part of text messages (C1). Emojis on the other hand are introduced by [6] as either part of a chat message or as a reaction to the content of a message from the chatbot (most often thumbs-up or thumbs-down as is for example implemented as a feedback option in ChatGPT). One general finding is related to chatbots as a combination between conversational and graphical user interfaces. While most patterns can be matched to chatbot functionalities as graphical interface elements, others only appear in combination with a message which was sent by the user. The pattern is then often related directly to the content of the user command or request (for example S7-9). This adds an additional layer of possible interactions to the GUI, which is usually dominated by actions such as tapping, swiping, scrolling or hovering with the mouse or using keyboard commands [7]. For the design of web chatbots this means that web engineers always have to plan for different situations in the conversation when designing interaction elements.

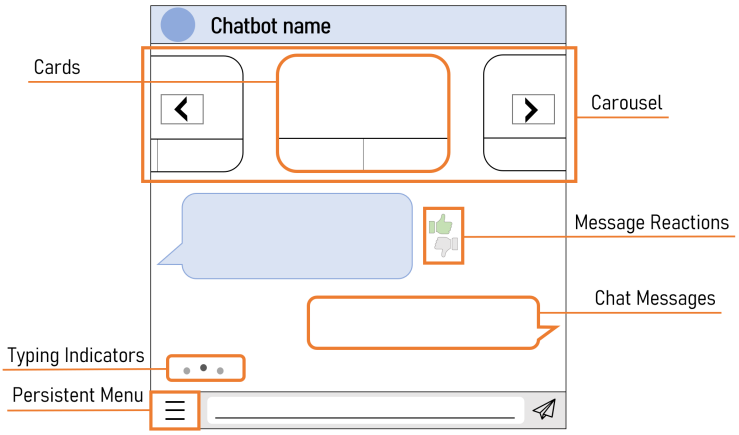

Fig. 1. A representation of some chatbot interaction patterns: *Chat Messages*, *Cards*, *Carousel*, *Message Reactions*, *Typing Indicators*, and *Persistent Menu*.

5 Chatbot Interaction Patterns

This section presents our catalogue of chatbot interaction patterns, which we built according to the findings from our mapping process. Table 2 is providing a short overview on the chatbot interaction patterns. As in pattern languages different patterns relate and build on each other we also show this in the table by relating to the numeration of other patterns. More detailed descriptions of their relations and the patterns analog to our built categories for the mapping are provided in the appendix. Figure 1 illustrates some of the described patterns as they would appear in a chatbot interface.

When possible, the naming of the original interaction design patterns was applied. In cases where web chatbots have a specific implementation, the name was built based on the matched chatbot functionality and was sometimes adopted slightly. The pattern of Help Systems is relating to other pattern types which are all offering help in different forms and situations. While it can be argued that such a pattern is redundant, we base this decision on [7] who propose a similar pattern. From the Help Systems groups of chatbot functionalities listed in Section 4, we list Functionality Introduction and breakdown avoidance (as the pattern Conversation Recovery) as independent patterns, as the others are already related to in the patterns Call-on Menu and Persistent Menu.

Table 2. Chatbot interaction patterns, their description and relation to other patterns.

Nr. Pattern name	Description. (Relation to other patterns)
1. Chat Messages	Displaying outputs from chatbot and user, most often as speech bubbles.
2. Cards	Presenting similar information, often with text, images, and actions. (1., 3.)
3. Carousel	A set of similar items in a horizontal list. (2.)
4. Quick Replies	A set of actions or functionalities under a chat message in the form of buttons. (1.)
5. Typing Indicators	A visual representation of the waiting time for the reply of the chatbot. (1.)
6. Persistent Menu	A constant menu in the chat window for important information or settings.
7. Call-on Menu	A dynamic representation of consistent information which was requested by the user. (1., 4., 6.)
8. Webview	A preview on web content from other providers or the chatbot homepage itself. (1.)
9. Message Reactions	Emojis beside or under a chat message to provide feedback on the answer of the chatbot. (1.)
10. Help Systems	Offering assistance for users, either on request or proactively by the chatbot. (4., 6., 7., 11.)
11. Functionality Introduction	Introducing the main functionalities or providing a tutorial in the first chat messages. (1.)
12. Conversation Recovery	Avoiding conversation breakdowns by providing possible actions or commands. (4., 7.)

6 Conclusion and Future Work

In this paper, we present a catalogue of 12 chatbot interaction patterns which were derived by mapping known chatbot functionalities to already established interaction design patterns. We show that patterns which are based on purely graphical web and mobile interfaces are transferable or adaptable for web chatbots. The graphical and conversational nature from chatbots leads to these patterns being also dependent on the context of the previous conversation.

With our presented catalogue, we provide an orientation for web engineers for building web chatbots which will feel familiar and consistent to the users. In future research we plan to enhance this catalogue with more patterns which were not yet described and to sharpen the descriptions of the ones we introduce here. Due to the limitation in this paper on described functionalities from [6] and [8], we might be able to verify the application of more interaction design patterns from [7] like Accordions, Collapsible Panels or List Inlays in web chatbots by analysing real-world chatbots. Another research direction would be to look into different domains of chatbots in the web and if there are distinctions in the used chatbot design patterns. It is possible that users have preferences on which patterns they are using, which are depending on the tasks the chatbot has to perform and the domain it is implemented in.

References

1. Alexander, C.: A pattern language: towns, buildings, construction. Oxford university press (1977)
2. Ashktorab, Z., Jain, M., Liao, Q.V., Weisz, J.D.: Resilient Chatbots: Repair Strategy Preferences for Conversational Breakdowns. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. p. 1–12. CHI '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3290605.3300484
3. Jain, M., Kumar, P., Kota, R., Patel, S.N.: Evaluating and Informing the Design of Chatbots. In: Proceedings of the 2018 Designing Interactive Systems Conference. p. 895–906. DIS '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3196709.3196735
4. Marcus, A.: Chapter 19 - Graphical User Interfaces, pp. 423–440. North-Holland, Amsterdam, second edition edn. (1997). https://doi.org/10.1016/B978-044481862-1.50085-6
5. Moore, R.J., Arar, R.: Conversational UX design: A Practitioner's Guide to the Natural Conversation Framework. ACM Books, ACM Books, New York, first edition edn. (2019). https://doi.org/10.1145/3304087
6. Shevat, A.: Designing bots: Creating Conversational Experiences. O'Reilly, Beijing [China]; Boston, first edition edn. (2017)
7. Tidwell, J., Brewer, C., Valencia, A.: Designing Interfaces: Patterns for Effective Interaction Design. O'Reilly, Beijing [China]; North Sebastopol, CA, third edition edn. (2020)
8. Valério, F.A.M., Guimarães, T.G., Prates, R.O., Candello, H.: Chatbots Explain Themselves: Designers' Strategies for Conveying Chatbot Features to Users. Journal on Interactive Systems **9**(3) (Dec 2018). https://doi.org/10.5753/jis.2018.710