



Design Systems for Conversational UX

Robert J. Moore, Eric Young Liu, Saurabh Mishra, Guang-Jie Ren

IBM Research-Almaden

San Jose, California, 95120 USA

rjmoore@us.ibm.com, Eric.Young.Liu@ibm.com,

Saurabh.Mishra51@ibm.com, gren@us.ibm.com

ABSTRACT

Design systems for creating user experience reduce effort, scaffold learning and increase collaboration [2], yet they are currently available primarily for applications with visual interfaces. Design systems for conversational UX are still few and far between. Any design system should provide a distinctive philosophy, interaction patterns and content format. The Alma Design System, from IBM Research, provides these. Its design philosophy takes human conversation as its metaphor and adapts principles from conversation science, such as mutual understanding, recipient design, minimization and repair. Its pattern language provides 100 interaction patterns and components adapted from the literature of Conversation Analysis. It includes formal patterns for conversational activities and their management. And its content format consists of six slots and can be extended to any number of parts.

CCS CONCEPTS

• **Human-centered computing** → **Natural language interfaces**

KEYWORDS

Conversational UX design, UX design, Conversation Analysis, Human-Computer Interaction

ACM Reference format:

Robert J. Moore, Eric Young Liu, Saurabh Mishra and Guang-Jie Ren. 2020. Design Systems for Conversational UX. In *2nd Conference on Conversational User Interfaces (CUI '20)*, July 22–24, 2020, Bilbao, Spain. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3405755.3406150>

1 INTRODUCTION

Today, interaction and user experience (UX) design for graphical interfaces, including desktop and mobile, is relatively mature. Icons, windows, menus, dragging, swiping, searching, etc. are familiar to millions if not billions of users. Since 2011, design systems have emerged to support UX design at scale [2]. "Design systems lay out the core elements and components of visual and

interactive design.... They offer guidance for layout, shape/form, color palettes, buttons, forms, tables, typography, user task flows, ... content, branding, tone, and personality" [2].

Unfortunately design systems are mostly available only for graphical interfaces. While there are some design systems for the *graphical elements* of text-based chatbots [1], design systems for conversational interfaces, are still few and far between. Since the launch of Apple's Siri in 2011, followed by Amazon's Alexa in 2014, as well as numerous text-based chatbots, natural language interfaces have surged in popularity; however, the design discipline has not yet caught up with them! Conversational interfaces constitute a fundamentally different kind of user experience. It is achieved primarily through the sequencing of utterances rather than through the visual arrangement of graphical components. Conversational UX design requires knowledge of human conversation rather than graphic design and interaction.

2 ADVANTAGES OF DESIGN SYSTEMS

Churchill [2] argues that design systems in general offer three advantages: "Design systems reduce effort.... They help maintain consistency and reduce divergence across products and product suites.... Design systems scaffold learning.... [they] show you a world that you can dismantle, re-create, rearrange, and extend.... Design systems can increase cross-disciplinary and cross-functional collaboration." In short, design systems enable developers to scale a particular style or system of UX.

3 CORE COMPONENTS OF DESIGN SYSTEMS

Design systems should have a few core components: a philosophy, interaction patterns and a content format.

A *design philosophy* establishes the overarching vision and goals of the system. It provides interaction metaphors and principles that designers can rely on when creating something new or seeking to insure consistency across the user experience.

Interaction patterns make up the core of the user experience within a design system. They provide designers with reusable components and users with recognizable interaction types. This frees designers from having to reinvent how users interact with the application and frees users from having to learn new interaction methods so both can focus on the content of the particular application.

Users can easily navigate and operate graphical interfaces, in part because they recognize certain interaction mechanisms, such as icons, scroll bars, drop-down menus, hamburger menus, etc. Similarly, we can engage in natural conversation with

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CUI@CHI Workshop Paper

CUI '20, July 22–24, 2020, Bilbao, Spain

© 2020 Copyright is held by the owner/author(s).

ACM ISBN 978-1-4503-7544-3/20/07.

<https://doi.org/10.1145/3405755.3406150>

people we have never met because we employ recognizable mechanisms when we talk, such as greetings, offers, requests, news deliveries, farewells, etc.

A pattern language for natural-conversation-style agents should provide generic solutions to the following kinds of problems: How different kinds of activities are structured; how both parties manage those activities; how they progress from one to the next; and how they enter into and exit out of the encounter.

A *content format* enables efficient generation of new content for particular use cases. It enables a content designer to create new material for the application without having to get into the guts of the system. For example, in a conversational interface, a content designer can produce user and agent utterances for the particular use case and give them to a dialog designer or engineer. The format enables the content to be added to the dialog tree in an easy and even an automated way. This frees the content designer to focus on content generation, rather than programming intents and dialog nodes.

4 ALMA DESIGN SYSTEM

The Alma Design System (ADS) for conversational UX is based on the Natural Conversation Framework (NCF) [3], developed by conversation scientists at IBM Research. The ADS implements the NCF on the IBM Watson Assistant platform.

4.1 Design philosophy

The design philosophy of the Alma Design System consists of two assumptions: that *natural conversation* is the primary interaction metaphor and that *mutual understanding* between user and agent is an overarching goal.

In addition to a philosophy, a design system should have concrete principles that guide design. Because Alma assumes natural conversation as its goal, it draws design principles from human conversation. Conversation analysts show that humans exhibit three principles in the ways they design their own talk: Recipient Design, Minimization and Repair [4].

Recipient Design is the tendency to tailor an utterance for its particular recipient. *Minimization* is the tendency to use the fewest words necessary to get the particular recipient to understand. And *Repair* is a set of practices for recovering when either party encounters trouble in speaking, hearing or understanding. Alma takes these native design principles of human speakers and adapts them to the UX design of automated agents.

4.2 Interaction patterns

The NCF adapts formal conversation patterns from the literature of Conversation Analysis. It provides a pattern language that is organized into three classes of patterns: A. Conversational Activities, B. Sequence-Level Management and C. Conversation-Level Management. Each class contains five modules containing 100 patterns collectively [3].

Conversational Activities (A) consist of basic interaction patterns, including the user asking the agent a simple question or making a simple request (A1), the user making a complex request (A2), the agent telling a story, giving instructions or doing some other extended telling (A3), the agent asking the user a question to which it already knows the answer and evaluating the user's response (A4) and the agent asking the user a question to which it does not know the answer and capturing new information (A5). (See Figure 1 for two of the 100 patterns.)

A4.0 Quiz (Correct)

Pattern

1 A: QUESTION
2 U: CORRECT ANSWER
3 A: CONFIRMATION

Example

1 A: Why can't you trust atoms?
2 U: umm they make up everything
3 A: That's it!

A4.1 Quiz (Incorrect)

Pattern

1 A: QUESTION
2 U: INCORRECT ANSWER
3 A: DISCONFIRMATION
4 U: REFUSAL/ABORT
5 A: CORRECT ANSWER
6 U: SEQUENCE CLOSER

Example

1 A: Why can't you trust atoms?
2 U: uhh you can't see them?
3 A: Wrong! Try again.
4 U: you tell me
5 A: They make up everything.
6 U: haha good one!

Figure 1: Two Interaction Patterns.

These five software modules enable designers to create all kinds of conversational activities, from answering a healthcare policy question (A1) to booking an airline ticket (A2) to eliciting a restaurant review (A5) to much more. In addition, like building blocks, these modules are combinable [3], e.g., a *troubleshooting* activity can be created by combining the Open Request (A2), in which the user reports a problem and the agent asks a series of diagnostic questions, with the Extended Telling (A3), in which the agent gives a set of instructions for fixing the problem. Or a *tutoring* activity can be created by combining an Extended Telling (A3) about an educational topic, with an Inquiry (A1) through which the student asks the agent questions and followed by a Quiz (A4) in which the agent tests the student on the material presented. A design system should provide the

building blocks out of which *any* kind of conversational activity can be built.

While Conversational Activity (A) patterns make up the primary activities of a conversation space, the Sequence-Level Management (B) patterns of the NCF enable user and agent to *manage* those primary activities. What should the agent do if it does not understand the user's utterance? Or if the user does not understand the agent's utterance? How does the agent know when it has answered the user's question or fulfilled the user's request and should move on? Human conversation provides general solutions to these problems, which automated agents can reuse.

The NCF provides patterns for agents to repair (B1) its own troubles in understanding, for users to repair their own hearing or understanding troubles (B2, B3), for users to signal the completion of a sequence (B4) and for users to abort an unsuccessful sequence (B5). For example, if the user makes a simple inquiry (A1), "what's your favorite movie?," and the agent responds with "I guess I like movies with a strong AI lead," the sequence-level patterns give the user multiple options for responding in "3rd position" [4]: "thanks" (B4.2) suggests a successful sequence, "what did you say?" (B2.1) indicates a trouble in hearing, "what do you mean?" (B2.4; see Figure 2) indicates a trouble in understanding and "never mind" (B5.0) suggests failure and the intent to move on. These patterns thus constitute forms of fine-grained user feedback that are native to natural conversation. They help user and agent achieve the core design assumption of *mutual understanding*.

Finally, while Sequence-Level Management (B) patterns help the parties manage particular sequences in the interaction, the Conversation-Level Management (C) patterns of the NCF enable them to coordinate the initiation and ending of the encounter. They provide familiar ways for the agent or user to open the conversation (C1, C2), talk about what they can do together (C3) and ways to end the conversation by closing it (C4) or by disengaging without closing (C5). Together, these patterns provide ways for the user and agent to introduce topics or activities, transition to next new ones, check for and produce last topics and close the encounter. The agent opening (C1) module provides slots for a *first topic* and a *second topic* that can be configured with actions such as welfare checks, name requests, offers of help, capability giving and more. The closing (C4) module provides patterns for checking for users' last topics, with "Anything else?," and a slot for the agent's *last topic* before farewells. The last topic slot may be configured for appreciations, success checks, name requests, promotions and more.

Taken together the three classes of interaction patterns in the NCF provide for all of the major stages of a natural conversation.

4.3 Content format

The content format of the NCF follows the principles of *minimization* and *repair*. It includes these slots:

- 0.1 User Utterance
- 0.2 Agent Utterance
- 0.3 Agent Repeat
- 0.4 Agent Paraphrase
- 0.5 Agent Example
- 0.6 Agent Definition

The User Utterance (0.1) captures things that users might say. Multiple variations of the same action (at least 20) are labeled 0.1a, 0.1b, 0.1c, etc. and used as training data for an *intent*. (The User Utterance may alternatively contain a text trigger, sent by a backend, which simulates the agent initiating a sequence.) The Agent Utterance (0.2), on the other hand, is the response to the User Utterance. It may contain a single text string or multiple strings to be selected either sequentially or randomly.

```

1 U: what's your favorite movie?
2 A: I guess I like movies with a
    strong AI lead.
3 U: what did you say?
4 A: I guess I like movies with a
    strong AI lead.
5 U: a strong what?
6 A: ai lead.
7 U: what do you mean?
8 A: I like movies in which the
    main character is an
    Artificial Intelligence.
9 U: what do you mean by
    artificial intelligence?
10 A: Oh! AI stands for Artificial
    Intelligence. Basically,
    robots and smart machines
    like me!
11 U: can you give an example?
12 A: Um, 2001: A Space Odyssey...
    The Matrix... Terminator...
    those are a few.
13 U: ok thanks
14 A: You're welcome!

```

Figure 2: Example of NCF-based Interaction.

The remaining slots in the content format (0.3-0.6) consist of responses to multiple *repairs* that users can initiate on the Agent Utterance (0.2). The Agent Repeat (0.3) is what the agent will say if the user requests a repeat of the Agent Utterance (0.2), which typically will be a copy but may contain other things like a partial copy or an abort, "never mind." The Agent Paraphrase (0.4) is an alternative wording of the Agent Utterance (0.2). Following the principle of *minimization*, the latter should typically be a short but understandable utterance, while the former an *elaboration*, which makes fewer assumptions about the user's knowledge but may be inelegant and more awkward (see Figure 2, line 8). However, occasionally, when the Agent Utterance (0.2) must be long and is potentially confusing, the Agent Paraphrase (0.4) should then be shorter, formulating only the *upshot*. The Agent Example (0.5) is the response to, "can you

give an example?" and may be an example of a more general entity or an example of the type of response expected by the agent. Finally, the Agent Definition (0.6) consists of technical or potentially ambiguous terms contained in the Agent Utterance (0.2) and their definitions.

The slots in the content format, 0.1-0.6, then make up a sequence of content. For example:

```
0.1a how do I meditate?
0.1b how do I do that?
0.1c teach me to meditate
0.2 First, sit comfortably and
    breathe slowly.
0.3 Sit comfortably and breathe slowly.
0.4 Just sit and relax.
0.5 Sit cross-legged or in a chair.
0.6 null
```

Alternatively, if the content designer chooses to give a slot a *null* value, a unique default message is given for that slot type, e.g., "I don't know how else to say it."

In addition, to create a *multi-utterance unit*, or Extended Telling (A3), such as a story or set of instructions, the content designer simply repeats the content format for each additional part of the telling (e.g., 0.1-0.6, 1.1-1.6, 2.1-2.6, etc.). Unlike the initial Agent Utterance (0.2), subsequent utterances (1.2, 2.2, 3.2, etc.) are triggered by a pause or by *continuers*, such as "ok," "mm-hmm," "uh-huh," "cool," etc. produced by the user instead of by a unique User Utterance (0.1). A sequence of content can contain any number of parts, e.g., a telling of the story *Goldilocks and the Three Bears* consists of nearly 30 parts.

5 DISCUSSION

As Churchill [2] argues, design systems reduce effort, scaffold learning and increase cross-disciplinary and cross-functional collaboration. The Alma Design System (ADS), which implements the NCF on IBM Watson Assistant, does each of these. It *reduces effort* by providing designers with a template for a whole conversation. They can start by inserting their domain-specific content into a fully functioning conversation space without having to create activity structures or conversation management mechanics. And all conversation spaces based on Alma work in the same way. The interaction method will be familiar to both users and designers who have encountered other Alma-based agents. The ADS also scaffolds learning by providing a sophisticated conversation space that designers or engineers can "dismantle, re-create, rearrange, and extend." It embodies many kinds of conversational actions and sequences. Both building on a dialog platform, such as Watson Assistant, Alexa or DialogFlow, and understanding natural conversation in a formal way are things that are new to many designers. For them, Alma provides a model from which they can learn. And it can also increase cross-disciplinary collaboration by embodying insights from conversation science. The development team and even stakeholders do not need to waste time arguing about the basics of how a conversation works.

6 CONCLUSION

Although there are numerous design systems for graphical interfaces, those for conversational interfaces, especially those which seek to emulate human conversation, are just beginning to emerge around the current generation of natural language and voice technologies. Such emerging systems should at least provide designers with an overarching philosophy, a set of interaction patterns and a content format.

Systems like the Alma Design System further demonstrate that UX design for voice assistants and text-based chatbots need not be divided into voice-only and text-only approaches. When the interaction is designed like a natural conversation, the same utterance sequences can be used with either mode of interaction. If *minimization* is applied to both, long agent utterances, which can be displayed but not spoken, are intentionally avoided. Paragraph-length utterances are broken into parts and are made interactive through the Extended Telling (A3) module. We call this "conversation-first design" [3].

REFERENCES

- [1] IBM Carbon Design System, Retrieved January 20, 2020 from <https://www.carbondesignsystem.com/>
- [2] Churchill, Elizabeth. 2019. Scaling UX with Design Systems. Interactions. XXVI.5 September-October, ACM, New York, 22.
- [3] Moore, Robert J. and Raphael Arar. 2019. Conversational UX Design: A Practitioner's Guide to the Natural Conversation Framework, ACM, New York.
- [4] Schegloff, Emanuel A. 2007. Sequence Organization in Interaction: A Primer in Conversation Analysis, vol 1. Cambridge: Cambridge University Press.