

# Conversational UX Design

A Practitioner's Guide to the Natural  
Conversation Framework



**Robert J. Moore  
Raphael Arar**



ASSOCIATION FOR COMPUTING MACHINERY

# **Conversational UX Design**

***A Practitioner's Guide to the Natural  
Conversation Framework***

**Robert J. Moore**

*IBM Research-Almaden*

**Raphael Arar**

*IBM Research-Almaden*

*ACM Books #27*



Copyright © 2019 by Association for Computing Machinery

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopy, recording, or any other except for brief quotations in printed reviews—without the prior permission of the publisher.

Designations used by companies to distinguish their products are often claimed as trademarks or registered trademarks. In all instances in which the Association for Computing Machinery is aware of a claim, the product names appear in initial capital or all capital letters. Readers, however, should contact the appropriate companies for more complete information regarding trademarks and registration.

*Conversational UX Design: A Practitioner's Guide to the Natural Conversation Framework*

Robert J. Moore

Raphael Arar

books.acm.org

<http://books.acm.org>

ISBN: 978-1-4503-6301-3 hardcover

ISBN: 978-1-4503-6302-0 paperback

ISBN: 978-1-4503-6303-7 ePub

ISBN: 978-1-4503-6304-4 eBook

Series ISSN: 2374-6769 print 2374-6777 electronic

DOIs:

[10.1145/3304087](https://doi.org/10.1145/3304087) Book

[10.1145/3304087.3304095](https://doi.org/10.1145/3304087.3304095) Chapter 7

[10.1145/3304087.3304088](https://doi.org/10.1145/3304087.3304088) Preface

[10.1145/3304087.3304096](https://doi.org/10.1145/3304087.3304096) Chapter 8

[10.1145/3304087.3304089](https://doi.org/10.1145/3304087.3304089) Chapter 1

[10.1145/3304087.3304097](https://doi.org/10.1145/3304087.3304097) Chapter 9

[10.1145/3304087.3304090](https://doi.org/10.1145/3304087.3304090) Chapter 2

[10.1145/3304087.3304098](https://doi.org/10.1145/3304087.3304098) Appendix A

[10.1145/3304087.3304091](https://doi.org/10.1145/3304087.3304091) Chapter 3

[10.1145/3304087.3304099](https://doi.org/10.1145/3304087.3304099) Appendix B

[10.1145/3304087.3304092](https://doi.org/10.1145/3304087.3304092) Chapter 4

[10.1145/3304087.3304100](https://doi.org/10.1145/3304087.3304100) Appendix C

[10.1145/3304087.3304093](https://doi.org/10.1145/3304087.3304093) Chapter 5

[10.1145/3304087.3304101](https://doi.org/10.1145/3304087.3304101) Appendix D

[10.1145/3304087.3304094](https://doi.org/10.1145/3304087.3304094) Chapter 6

[10.1145/3304087.3304102](https://doi.org/10.1145/3304087.3304102) References/Index/Bios

A publication in the ACM Books series, #27

Editor in Chief: M. Tamer Özsu, *University of Waterloo*

Area Editor: Michel Beaudouin-Lafon, *Université Paris-Sud and Institut Universitaire de France*

This book was typeset in Arnhem Pro 10/14 and Flama using ZzTeX.

Cover art: Raphael Arar.

Cover image: The Euphonia, (The Talking Machine); Source: *The London Journal: Weekly Record of Literature, Science and Art*. NO. 1340, VOL. LII, London: 1870, November 1, 1870, pg. 245.

First Edition

10 9 8 7 6 5 4 3 2 1

# Contents

Preface **xiii**

## **Chapter 1** Introduction **1**

- 1.1 Natural Language Interfaces **3**
- 1.2 Conversational UX Design **5**
- 1.3 Conversation First **7**
- 1.4 Mutual Understanding **17**
- 1.5 About This Book **23**

## **Chapter 2** Conversation Analysis **25**

- 2.1 What Is a Conversation? **26**
- 2.2 Topic vs. Structure **27**
- 2.3 Anatomy of a Conversation **29**
- 2.4 Conversation Types **39**
- 2.5 Conclusion **43**

## **Chapter 3** Conversation Authoring **45**

- 3.1 A Simple Finite-State Machine **47**
- 3.2 Conditions (Intents, Entities, Context) **49**
- 3.3 Responses **56**
- 3.4 Branching **59**
- 3.5 Conclusion **62**

## **Chapter 4** Natural Conversation Framework **63**

- 4.1 Interaction Model **65**
- 4.2 Content Format **69**
- 4.3 Conversational Activity Patterns **76**
- 4.4 Conversation Navigation **79**
- 4.5 Conclusion **85**

**Chapter 5** Conversational Activity UX Patterns **87**

- 5.1 Pattern Language for Conversational UX **88**
- 5.2 A1 Inquiry (User) **90**
- 5.3 A2 Open Request **93**
- 5.4 A3 Extended Telling **103**
- 5.5 A4 Quiz **106**
- 5.6 A5 Inquiry (Agent) **109**
- 5.7 Combining Patterns **114**
- 5.8 Conclusion **119**

**Chapter 6** Sequence Management UX Patterns **121**

- 6.1 B1 Repair (Agent) **122**
- 6.2 B2 Repair (User) **128**
- 6.3 B3 Extended Repair **138**
- 6.4 B4 Sequence Closers **142**
- 6.5 B5 Sequence Abort **145**
- 6.6 Conclusion **147**

**Chapter 7** Conversation Management UX Patterns **149**

- 7.1 C1 Opening (Agent) **149**
- 7.2 C2 Opening (User) **155**
- 7.3 C3 Capabilities **161**
- 7.4 C4 Closing **165**
- 7.5 C5 Disengaging **174**
- 7.6 Conclusion **180**

**Chapter 8** Conversational UX Design Process **183**

- 8.1 Design Thinking **183**
- 8.2 Stage 1: Empathize **187**
- 8.3 Stage 2: Define **190**
- 8.4 Stage 3: Ideate **192**
- 8.5 Stage 4: Prototype **203**
- 8.6 Stage 5: Test **208**
- 8.7 Conclusion **209**

**Chapter 9** Conclusion **211**

- 9.1 Conversation Metrics **212**
- 9.2 Final Thoughts **217**

**Appendix A** Conversation Analysis Transcription Conventions **219**

**Appendix B** Dialog Design Pseudocode **221**

**Appendix C** NCF Pattern Language Summary **223**

    NCF Pattern Types **223**

    NCF Patterns and Examples **226**

**Appendix D** Sample Conversation with Alma **265**

    References **275**

    Index **283**

    Author Biographies **297**

# Introduction

*All conversations are interactions, but not all interactions are conversations.*

—Moore and Arar [2018]

While we all know how to have a conversation, we do not all know exactly how they work. We typically ignore the mechanics of human conversation in order to focus on the message being communicated. But if we are to *design* such mechanics, we must understand them more formally. Fortunately, there are fields of study within the social sciences that do just this. Social sciences, like Conversation Analysis (CA), can help user experience (UX) designers understand the principles and patterns of natural human conversation for the purpose of creating user interfaces that attempt to simulate them.

The best way to understand the mechanics of natural human conversation, and other conversation-like interactions, is to capture their details in a mechanical recording and closely analyze a text-based transcription of that recording. Transcripts enable you to focus on the talk itself, rather than the medium, although when done face-to-face, embodied actions, as well as the physical setting and artifacts, may be required to understand it fully. Consider the following excerpt from a transcript of a conversation-like interaction (Example 1.1).

**Example 1.1** Rogerian Therapy.

- 1 A: Men are all alike.
- 2 B: In what way?
- 3 A: They're always bugging us about something or other.
- 4 B: Can you think of a specific example?
- 5 A: Well, my boyfriend made me come here.
- 6 B: Your boyfriend made you come here?
- 7 A: He says I'm depressed much of the time.
- 8 B: I am sorry to hear that you are depressed.
- 9 A: It's true. I am unhappy.
- 10 B: Do you think coming here will help you not to be unhappy?

What can we say about this interaction? We might say that it is about “how men are annoying.” Or we might say that A is complaining about men and B is listening. Or we might say that A is making statements and B is asking follow-up questions. Or we might even say that B is practicing Rogerian therapy, a person-centered method of psychotherapy pioneered by Carl Rogers in the 1940s, by getting A to talk about herself.

In fact, A is a user and B is the computer program ELIZA [[Weizenbaum 1966](#), pp. 36–37], one of the first chatbots. The user typed her utterances, and the automated agent returned textual responses. ELIZA appears to be a good listener, but it does not have much to say. It does not have any real-world knowledge to offer. And apart from Rogerian-style psychotherapy, it provides no service. [Weizenbaum \[1966\]](#), p. 42], the creator of ELIZA, explains how the chatbot works psychologically.

. . . it has a crucial psychological utility in that it serves the speaker to maintain his sense of being heard and understood. The speaker further defends his impression (which even in real life may be illusory) by attributing to his conversational partner all sorts of background knowledge, insights and reasoning ability. But again, these are the *speaker’s* contribution to the conversation. They manifest themselves inferentially in the *interpretations* he makes of the offered responses.

[Weizenbaum \[1966\]](#) points out that the behavior of the system leads users to attribute to it the ability to hear and understand. In other words, because ELIZA can do “listening” and even “probing,” users attribute more cognitive capabilities to it than it actually has. He suggests further that we may even form such illusory impressions, to some degree, with other humans in “real life.”

*From a conversational UX design perspective, the question is: how does ELIZA do it? How does the automated agent do “listening”?* An analysis of the transcript (see Example 1.1), as well as Weizenbaum’s (1966) paper, reveals how. First, we can see that ELIZA does listening by producing a generic *paraphrase request* (line 2) of the user’s initial statement (line 1). Such an utterance can be employed appropriately after *anything* the user says without recognizing its intent. The agent also produces a generic *example request* (line 4), which can also be used after about any statement made by a speaker (line 3). By requesting paraphrases, ELIZA gets the user to say what she already said again but in different words, which humans routinely do in order to manage troubles in understanding [[Schegloff et al. 1977](#)].

Next, ELIZA employs a range of generic repeat techniques. It repeats the user’s entire utterance (line 6) but with the appropriate transformations of the pronouns. This demonstrates that the agent “heard” what the user said (line 5) and understands it to the degree that it could change “my” to “your” and “me” to “you.”

Weizenbaum [1966, p. 42] suggests that this grammatical understanding leads the user to attribute deeper understanding. ELIZA further repeats a part of the user's prior utterance, with transformation, "I'm depressed" (line 7), and incorporates it into an utterance that recognizes it as bad news by offering an apology, "I am sorry to hear that you are depressed" (line 8). Finally, ELIZA produces another *partial repeat* of the user's utterance (line 9) and combines it with a reference back to an even earlier utterance (line 5): "Do you think coming here will help you not to be unhappy?" (line 10). Through the production of such generic repeat and paraphrase techniques, tailored to the user's prior utterance(s), ELIZA helps foster the impression that the user is being "heard and understood."

Weizenbaum (1966) explains that ELIZA consists of a set of rules and templates for transforming input utterances. ELIZA can do this without understanding the meaning of what is said but simply by manipulating natural language. Like a foreigner with limited knowledge of English [Weizenbaum 1966, p. 37], ELIZA applies a rule-based template to English utterances; for example, "one that specifies that any sentence of the form 'I am *blah*' can be transformed to 'How long have you been *blah*?,' independently of the meaning of *blah*." ELIZA identifies the "most important keyword," some minimal grammatical context, and transforms the user's utterance into a conversation-like response [Weizenbaum 1966, p. 37]. Through this mechanism, it appears to do "listening."

## 1.1

### Natural Language Interfaces

While chatbots and virtual agents have been around since ELIZA [Weizenbaum 1966], there has been a recent resurgence of interest in them as major computer companies have released their own. Apple's Siri, Amazon's Alexa, Google's Assistant, Microsoft's Cortana and IBM's Watson are just a few examples, not counting natural-language-based assistants by startups. With persistent Internet connections and machine-learning algorithms, virtual agents are much smarter today than they were 20 years ago. While most of these systems accept voice input from users, especially using far-field microphone arrays, many accept text input, sometimes from standard applications like SMS and Instant Messaging. But although virtual agents are becoming ubiquitous, interactions with them are still awkward, confusing, limited and fraught with troubles in mutual understanding.

Conversational interfaces are very different from graphical user interfaces (GUI). *In conversational interfaces, the graphical elements are generally minimal, for example, a text-entry box and a message history or a microphone button or nothing at all. User interaction is conducted primarily through the words: typed or spoken.* The

interaction metaphor for these interfaces is the natural, human conversation, rather than the direct manipulation [Shneiderman 1982] of graphical interfaces.

Although natural language processing (NLP) has given us powerful automated tools for analyzing the spoken and the written word alike, it does not provide a model of how bits of language are sequenced into an interaction that is recognizable as a “conversation” [Sacks et al. 1974]. Some [Button et al. 1995, p. 209] have pointed out that a purely linguistic approach to building conversational interfaces, in which natural language is isolated from social activities, underestimates the “extent to which the capacity to talk and to carry on a conversation involves the possession of something other than purely ‘linguistic’ skills, and, indeed, of other things than skills.” There is something more than language that must be modeled in a conversational interface and that is the organization of *natural conversation*. Natural human conversation is a complex system in its own right [Sacks et al. 1974, Schegloff 2007], which Sacks characterized as a “machinery” [Sacks 1984].

*Natural language processing (NLP) techniques enable the automatic analysis of bits of language, for example, English or Spanish or Mandarin, but how to string those bits of language together into the kinds of sequences that occur in natural conversation is left to the UX designer or software developer.* Authoring “questions” and their “answers” is simple enough. But how should the agent respond if the user says “okay” or “oh” or “what do you mean?” Building the range of natural conversational actions is still a hard problem. Luckily, Conversation Analysis (CA) focuses precisely on this domain. It documents the systematics, or “mechanics,” of how people naturally talk in a wide variety of settings and languages [Sacks et al. 1974, Schegloff 2007, Sidnell and Stivers 2013]. Understanding these systematics formally is necessary for designing conversational UX, in addition to understanding the systematics of language.

Today’s chatbots and voice assistants are something in between web search and a human. On the one hand, they are more conversational than interacting with a web search engine. You never say “thank you” or “what do you mean?” to a search engine, although it may say, “Did you mean X?” to you [Moore and Churchill 2011]. Search engines do not understand those kinds of conversational actions, only “queries.” On the other hand, chatbots and voice assistants are not as conversational as a native human speaker, such as your family or friends. While you can talk to your mother about a wide range of things that are going on in your life, virtual assistants, limited by current technology, do not possess that range of knowledge or ability to handle such conversational complexity. Instead, they are more like customer service interactions, which tend to be limited in scope, or conversations with non-native speakers, which tend to require simpler utterances.

Even the most “conversational” of natural-language agents do not truly engage in conversation. Human conversation is merely the metaphor for this type of user interface, just as *direct manipulation* is the metaphor for graphical user interfaces [Shneiderman 1982]. This kind of human-computer interaction is at best a “simulacrum” of human conversation [Button 1990, p. 68] and at worst bears no resemblance to it at all. We might think of conversational interfaces as “conversation games.” Like video games, they offer simplified simulations of the interactions with the real world, whether navigating a humanoid avatar through a realistic 3D environment or interacting with other players through virtual face-to-face [Moore et al. 2007].<sup>1</sup> Conversational interfaces are game-like in that they are interactive but consist of a limited set of rules and legal “moves” compared the real phenomenon they attempt to evoke. Just as users must learn how video-game interfaces work, or any other user interfaces for that matter, they too will need to learn how to “play” conversation games and how to navigate conversation spaces. Conversational interfaces constitute a distinctive form of interaction, which borrows interaction patterns from natural human conversation but also exhibits its own mechanics.

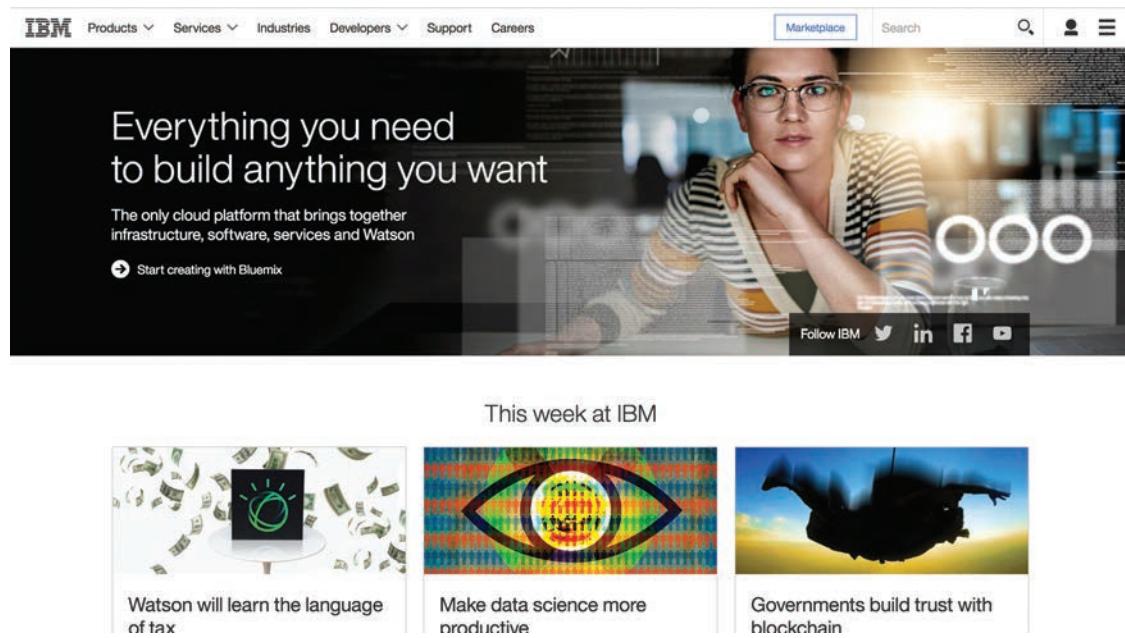
## 1.2 Conversational UX Design

While it is easy to create natural-language interfaces with today’s platforms, creating an effective and engaging user experience is still a major challenge. Creating an interaction that works like a human conversation in particular is an even harder problem. To appreciate the need for a new kind of UX design, consider *web design* for a moment. In 2016, the IBM homepage looked like this (Figure 1.1). Whether you like it or not, it is clearly designed by a professional. It has a clean look and feel. Attractive fonts and color palette. Colorful hi-res images. It uses standard, familiar elements, such as drop-down menus, a search box, social media icons, a “hamburger” menu and more. Furthermore, you can hire someone to build a user interface like this. *Web designer* is an established role that is part of a relatively mature discipline.

But this was not always the case. Consider the IBM homepage back in 1996 (Figure 1.2). By today’s standards, it looks amateurish at best. (No doubt it was created by an amateur web designer.) There are no consistent color palette or font guidelines. The main interaction element is the hypertext link. The layout is simplistic. The images, low-res. And the bullets and shadows look like they were created in

---

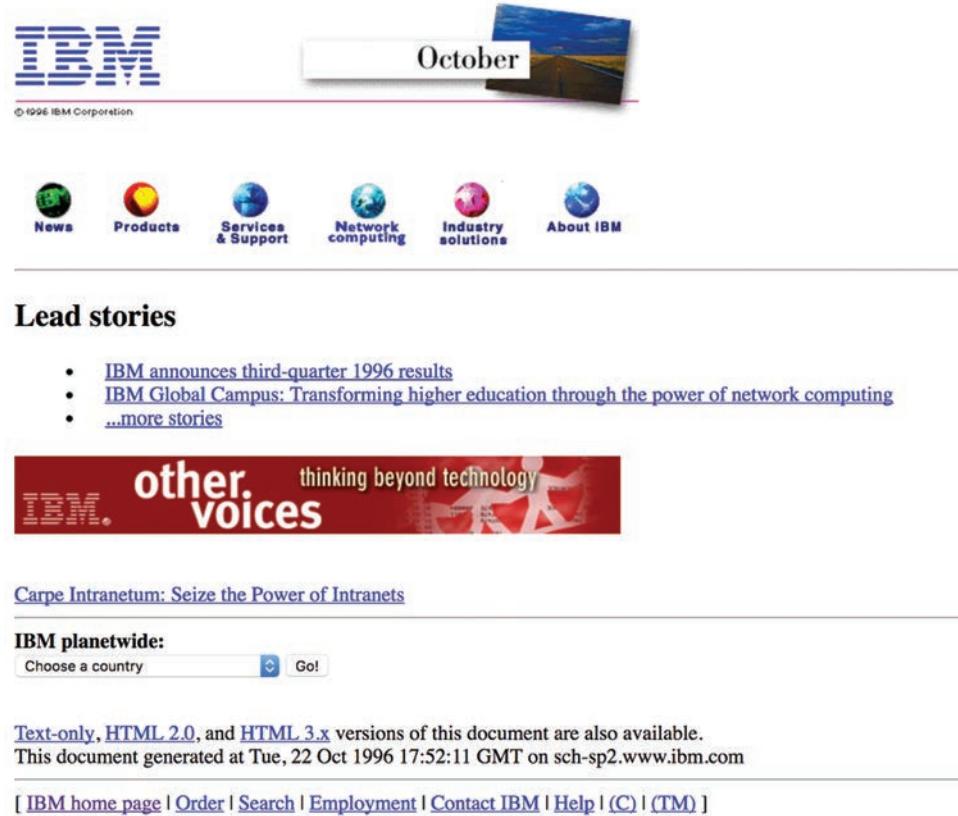
1. We do *not* mean to suggest that conversational agents are the same as the simplistic “dialogue trees” common in video games. Conversational agents are much more sophisticated and nonlinear.



**Figure 1.1** IBM Homepage 2016. (Reprint Courtesy of International Business Machines Corporation, © International Business Machines Corporation)

PowerPoint. So what is the difference? Over the past 20 years, the discipline of web design emerged. As the technical capabilities of the world wide web evolved, for example the <IMG> tag and high-speed networks, it became possible to apply formal knowledge of *graphic design* to web pages to make them “look more like glossy magazines” [Gillies and Cailliau 2000]. With the combination of user-interface design and graphic design, originally developed for the print industry, a new design discipline was born.

*Today, conversational interfaces are at the stage that web interfaces were in 1996: the technologies are in the hands of the masses, but mature design standards have not yet emerged around them.* And the visual solutions that evolved for graphical interfaces do not help us much with conversational interfaces, especially in the case of voice. With conversational interfaces, the user experience consists primarily of the design and sequencing of utterances. Instead of looking to the graphic arts for help, this time UX designers should look to the social sciences. Just as formal knowledge of the graphic arts revolutionized graphical user interface design, formal knowledge of human conversation can revolutionize conversational user interface design.



**Figure 1.2** IBM Homepage 1996. (Reprint Courtesy of International Business Machines Corporation, © International Business Machines Corporation)

## 1.3

### Conversation First

Current chatbot and voice assistant platforms enable you to create a variety of different styles of user interaction. Because they use natural language classification (NLC), these platforms are a technical advance over earlier voice control systems that only recognized a set of pre-specified command phrases. NLC enables recognition of a much wider range of phrases. However, how NLC is used together with the agent's responses and other UI elements can produce very different user experiences. Below are *invented* examples of different interaction styles that we see in today's chatbots and virtual agents: System-Centric, Content-Centric, Visual-Centric and Conversation-Centric [Moore and Arar 2018]. The styles are not mutually exclusive. A given system may contain features of all of them.

### **System-Centric Style**

Although system-centric interaction styles recognize natural language as input, they still require the user to conform to certain unnatural input requirements. That is, they require users to formulate their utterances as valid queries, which pack all of the details relevant to their request into a single turn at speaking. This input requirement is due to the fact that the automated agent will not remember the sequential context, or what has been said so far, from turn to turn. Technically speaking, each turn by the user initiates a new interaction, which always consists of only two turns. *Context* is used within an utterance but not between utterances. In addition, such systems tend to be user-driven: interactions with the system are initiated by the user. Finally, system-centric styles tend not to do much conversation management, things like repairing the prior turn, “what do you mean?”, or closing the prior sequence, “okay”, no doubt because they do not remember the prior utterance to which the user is referring.

### **Voice Control**

The system-centric interaction style is the most prevalent today. It includes voice control interfaces and information-retrieval interfaces, like that of web search. Voice control interfaces are perhaps the oldest type of natural language interface. They are based on an interaction model of simple input-output, or two-turn sequences, for the purpose of controlling a system. The system can only understand a short set of commands, not other social actions, and perform the appropriate response. Voice input is often provided as an alternative to button input so that users can operate the device hands free. Such voice control interfaces have been widely used since the early 2000s in cars. In addition to pressing a button on the dashboard or steering wheel, users can give a command, for example, “air conditioner on” or “radio 106.7 FM” or “find nearest gas station.” In most cases, the response to the command by the car is a nonverbal action.

Today, virtual assistants often use a system-centric, voice control style. For example, leading voice assistants, or smart speakers, mostly recognize and perform commands issued by the user, but they tend to respond with a wider range of utterances, as well as nonverbal actions. However, these assistants still largely fail to remember the sequential context across turns, treating most user utterances as a new exchange.<sup>2</sup> This prevents these agents from performing a wide variety of

---

2. An exception might be support for so-called “slotted intents.” These enable the agent to remember an intent from a previous turn until dependent details are elicited from the user. This is a limited form of multi-turn conversation and will be discussed in Chapter 5.

conversational activities. For example, although virtually all assistants claim to “tell jokes,” very few actually can do *joke telling* correctly. If you request a joke from an agent (Example 1.2, line 1), it will say a question joke (line 2) and then give away the punchline (line 3) (U stands for user and A stands for agent).

**Example 1.2** Question-Answer Joke Telling.

- 1 U: Alexa, tell me a joke.
- 2 A: How does NASA organize a party?
- 3 They planet!

When telling a question-answer joke, the teller waits to see if the recipient can answer the joke and gives the punchline only after the recipient shows that he or she cannot. Most virtual agents cannot do this, but instead only deliver one-liners. A more extreme example can be found from another leading agent. If you request a knock-knock joke from it, you get an even more unnatural response (Example 1.3).

**Example 1.3** Knock-Knock Joke.

- 1 U: Tell me a knock knock joke
- 2 A: Knock knock. Who's there? Bob. Bob who?
- 3 Bob, I don't do knock-knock jokes.

This agent will pack the whole joke-telling sequence into a single response. Neither of these are a correct way of telling a joke. Although the natural language content is that of a question-answer joke and a knock-knock joke, the sequential structure of the jokes is anything but natural. The problem of course, is not access to joke content, but the underlying interaction model of independent, two-turn exchanges. Jokes cannot be performed in a single turn, although witty remarks can. And likewise many other conversational activities require more than two turns to perform.

### Information Retrieval

In addition to voice control interfaces, another kind of system-centric style is that of natural language search interfaces. Like voice control interfaces, these interfaces recognize natural language, are user-driven, and are limited to two-turn exchanges. However, unlike voice control, natural language search interfaces return information as their responses, rather than nonverbal actions. Such interfaces tend to be text-based rather than voice-based and are built on a simple query-response model found in web search and other forms of information retrieval. For example, the following exchange is an example of this style (Example 1.4).

**Example 1.4** Information Search.

```
1 A: Hello. How can I help you!?
2 U: Original Medicare coverage diabetes screenings
3 A: Medicare covers tests to screen for diabetes as well
4     as services and supplies to help treat the disease.
5 U: Original Medicare eligibility diabetes screenings
6 A: Part B covers these screenings if your doctor determines
7     you're at risk for diabetes or you're diagnosed with
8     pre-diabetes. These lab tests are covered if you have
9     any of these risk factors:
10    ((list of factors))
11 U: what about glaucoma tests?
12 A: There are no results for your query. Please enter a
13     valid query.
```

The healthcare assistant recognizes only “valid queries” and then looks up medical information in a database. As long as the user types a valid query (lines 2 and 5), the agent will return results, if available (lines 3–4 and 6–10). In each query, all of the relevant keywords must be given and therefore repeated, for example, “Original Medicare” and “diabetes screenings” (line 2 and 5). However, if the user speaks in increments (line 11), building off the context of prior turns, as is common in natural human conversation, the system fails (lines 12–13). Information search interfaces are therefore very similar to web search engines, like Google or Bing. They treat everything as a new query. We know how to use search engines to find all kinds of useful information, but we never say “thank you” to them because we know they will search the web for that too.

In practice, system-centric interaction styles tend to be what results when the user experience is created by an engineer or developer rather than by a designer. They are modeled after familiar device control or information retrieval interactions, and they require the least work to build the dialog because they force the user to adapt to the valid-query format. However, system-centric styles are not very conversational. They lack a key characteristic of natural conversation, *persistent context*, and the affordances that are enabled by that.

### **Content-Centric Style**

Another interaction style found in chatbots and virtual assistants is what we might call content-centric. In terms of their underlying interaction model, they are identical to information search interfaces. They too recognize natural language, are user-driven, and are limited to two-turn exchanges. However, they differ primarily in the form of their content. Instead of retrieving data, they retrieve handcrafted

answers to questions like those commonly found in web-based Frequently Asked Question (FAQ) pages. In fact, in many cases, they are no doubt adaptations of existing FAQ content. For example, the following exchange is inspired by such systems (Example 1.5).

**Example 1.5** Q & A.

1 A: Hello, I am an automated Medicare Assistant. I can  
2 answer questions about Medicare insurance. What is your  
3 question?  
4 U: am I covered for diabetes screenings?  
5 A: Medicare covers tests to screen for diabetes as well  
6 as services and supplies to help treat the disease.  
7 Medicare will pay for 100% of its approved amount for  
8 test even before you have met the Part B deductible.  
9 You will pay no copay or deductible for these tests  
10 if you see doctors who take assignment. Doctors and  
11 other health care providers who take assignment cannot  
12 charge you more than the Medicare approved amount.  
13 Medicare Advantage Plans cover all preventive services  
14 the same as Original Medicare. This means Medicare  
15 Advantage Plans will not be allowed to charge cost-  
16 sharing fees (coinsurances, copays, or deductibles)  
17 for preventive services that Original Medicare does  
18 not charge for as long as you see in-network providers.  
19 U: thanks  
20 A: I'm sorry, I don't understand your question.

When the user asks a yes/no question about health insurance coverage (line 4), he receives a thorough and lengthy response (lines 5–18). Instead of answering the question with a yes or a no, the agent answers with a statement, “Medicare covers tests to screen for diabetes” (line 5). From this kind of response, the user can usually infer a yes or a no. However, the agent continues on to explain multiple details and contingencies: coverage includes related “services and supplies” (line 6), it does not count toward your deductible (lines 7–8), there is no copay (line 9), approved doctors cannot charge more than the approved amount (lines 10–12), this applies to both Medicare Advantage Plans and Original Medicare (lines 13–18), and more. In other words, instead of saying yes or no, the agent implies that “it depends,” and then details the set of circumstances on which it depends. Notice that some of the details will not be relevant to the user; for example, users with “Original Medicare” only do not need to be told about “Medicare Advantage Plans.”

Overall, the authored answers tend to be relatively verbose and detailed since accuracy and thoroughness are of primary importance. In fact, content-centric interfaces tend to be co-created with subject matter experts or content designers who author the responses. But the disadvantages of long and thorough responses is that they preclude use in voice-based interfaces, and their length tends to limit opportunities for back-and-forth over a single topic since all of the details are presented in one extended monologue. In content-centric systems, recognized utterances, or “intents” (Chapter 3), tend to be topic-based rather than action-based, as they direct all inquiries or requests pertaining to a particular topic to a single response.

### **Visual-Centric Style**

Visual-centric interaction styles borrow heavily from GUI, web and mobile interfaces, all of which rely on the principle of *direct manipulation* [Shneiderman 1982] instead of text commands or natural language. Graphical elements, such as buttons, images, emojis and other visual elements, are mixed into the interaction alongside natural language inputs. The most common type of visual-centric interface is the “chatbot” interface. Chatbots combine chat, Instant Messaging or texting-style interfaces with Internet bots that perform simple tasks. The following example shows what the chatbot or visual-centric style interaction looks like (Example 1.6).

**Example 1.6** Chatbot.

```

1 A: Hi, I'm Melanie, your virtual nurse! I can answer your
2 Medicare questions. To start, which plan do you have?
3 [Original Medicare] [Medicare Advantage] [ Other ]
4 ((U clicks/taps Medicare Advantage button))
5 A: Okay, Medicare Advantage. What is your question?
6 U: am I covered for diabetes screenings?
7 A: Medicare covers tests to screen for diabetes as well
8 as services and supplies to help treat the disease.
9 Medicare will pay for 100% of its approved amount for
10 the test even before you have met the Part B
11 deductible. You will pay no copay or deductible for
12 these tests if you see doctors who take assignment.
13 A: Was this answer useful?
14 [ Yes ] [ No ]

```

Like the older Interactive Voice Response (IVR), chatbots tend to constrain user to a limited set of response options. However, instead of surfacing those options

through speech, chatbot interfaces surface them through sets of buttons (lines 3 and 14). By offering buttons instead of requiring free text, chatbot systems improve the discoverability of intents recognized by the system and, for those particular intents, improves the speed of user input. Also, because they are text-based, chatbot interfaces are often verbose or text-heavy, slowing the rate of interaction with longer responses.

Chatbots often incorporate web and mobile techniques for eliciting user feedback after returning a response. For example, after providing an answer (lines 7–12) to the Medicare question (line 6), the agent asks, “Was this answer helpful?” (line 13), along with buttons for yes and no (line 14). Or they will elicit feedback on the answer with thumbs-up and thumbs-down buttons. Since yes/no questions already project the type of answer they prefer and because both responses are very short, there is really no advantage to displaying buttons for yes and no, unless the designer intends to constrain the user to only those options. In fact, in keyboard and mouse situations, clicking yes/no buttons requires moving one’s hand off of the keyboard and onto the mouse or trackpad and back, thereby eliminating their convenience. But more importantly, chatbot interfaces tend to be agent-driven and somewhat linear, minimizing points at which users are not told what to do next. For example, the Medicare chatbot does not wait for the user’s response to its answer about diabetes screenings. Instead it takes the next turn, hijacking “third position” with the request for feedback (line 13). In human conversation, third position, which comes after the first and second parts of an action pair, or “adjacency pair” (e.g., question-answer, Schegloff [2007]), is an important slot in which the initiator of the pair can give natural feedback, such as closing the sequence with “thank you” or repairing the second part with “what do you mean by doctors who take assignment”? Asking “was this helpful?” after every agent answer precludes the user from doing a range of third-position actions that provide a wider range of conversational feedback (Chapter 6).

Because chatbots solve some conversational UX design problems through visual elements, rather than through verbal actions, they cannot be fully ported to voice-only platforms. As a result, we see a split in the design of natural language interfaces along the lines of text- and voice-based systems. As chatbot interfaces introduce more buttons and visual elements, their designs drift farther from voice and conversational interfaces and closer to GUI and mobile interfaces. The more sophisticated chatbot interfaces are created by UX designers who apply their expertise in graphic design to the problem of natural language interaction.

### Conversation-Centric Style

Finally, a conversation-centric style, the topic of this book, can be distinguished from the other three. In addition to recognizing natural language, a conversation-centric style also recognizes and exhibits the kinds of actions that are distinctive of natural human conversation. In other words, it can do *conversation management*: it can recognize social actions for managing the conversation itself. “*What did you say?*,” “*none of your business*,” “*okay*,” “*never mind*” are all operations on prior utterances in the interaction. The social actions they perform constitute a major part of what makes a conversation a conversation. Likewise, you would not say a computer could “play chess” if it could not perform checking, castling and all the other actions that constitute *chess*. Furthermore, you will not find such conversational actions in other forms of language use, such as news stories, administrative forms, poems, etc. Designing the Medicare agent with a more conversational interaction style would entail conversation management (Example 1.7).

**Example 1.7** Conversational.

```

1 A: Hi, I'm Alma. I'm here to answer any questions you
2      have about your Medicare insurance. How can I help
3      you?
4 U: I have a question about diabetes screenings
5 A: Okay, what's your question?
6 U: am I covered for them?
7 A: Diabetes screenings are covered at 100% of the
8      approved amount so there is no copay or deductible
9      if you see doctors who take assignment.
10 U: what do you mean by take assignment?
11 A: Oh. Assignment means that your doctor, provider or
12      supplier agrees to accept the Medicare-approved
13      amount as full payment.
14 U: ok thanks
15 A: You're welcome! Is there anything else I can do
16      for you?

```

The conversational agent can handle *preliminaries* to inquiries and requests (line 4), which check the conditions upon which those inquiries and requests depend. The conversational agent can handle a range of *repeat* and *paraphrase requests* on its own or the users' responses (line 10). And the conversational agent recognizes common methods of *closing a sequence* (line 14) or *closing a conversation*. All of these require that the agent remember what it is talking about across multiple turns.

In addition, the responses in a conversation-centric interaction are relatively short, or “bite-sized,” as utterances tend to be in natural human conversation. This enables efficiency and speed as conversations happen in real time, either through

voice or through text. Instead of giving the user a long and thorough response, as with the content-centric style, the conversation-centric style breaks the same content into smaller chunks, returns the most relevant chunk first and allows the user to unpackage the other chunks as needed. The result is more back-and-forth utterance exchange instead of longer monologues.

Conversation-centric styles are fully mixed-initiative and nonlinear. Either the user or the agent can lead, and the user can always redirect the conversation. Consequently, they frequently rely on the user to decide next actions, rather than on the agent leading at every turn transition. Such a conversation-centric style of interaction is the goal of a conversational UX designer, who brings a formal knowledge of how human conversation works to the design of talking agents. And because it is accomplished primarily through the words, without relying on visual interaction methods, a conversational style will work with either voice or text input. The conversation-centric interaction style therefore is characterized by (a) persistent sequential context; (b) fewer words per response, more back-and-forth; (c) nonlinear interactions with many trajectories; and (d) recognition of common conversational activities.

While each of the natural-language interaction styles detailed above, system-centric, content-centric, visual-centric, conversation-centric, can be useful to users for particular applications, we believe that the future of AI interfaces is the conversation-centric style. People have always wanted to talk to machines the way they talk to each other. This book is about designing conversation-centric interfaces, or *conversational agents*, not simple chatbots or voice control systems.

We advocate a *conversation-first* approach to the design of natural language interfaces, which is analogous to the mobile-first strategy in web design [Moore and Arar 2018]. While mobile-first design begins with the small screen of the mobile device and scales up to larger displays [Wroblewski 2011], conversation first begins with just verbal input and output, whether voice or text. The UX designer must enable the user to converse with the agent through the words alone, without buttons or visual aids. Voice interfaces, or platforms like the Short Message Service (SMS), force one to design for short utterances rather than for buttons, long lists, or document-like responses. Once the conversation is fully functional, it can be enhanced, as needed, through coordination with visual aids, just as a human speaker may use menus or charts to supplement his or her talk. As a result, users can talk to an agent through multiple communication channels, although the experiences will vary in their affordances, similar to human conversation when the speakers are talking face-to-face versus talking over the telephone. A conversation-first design strategy involves a focus on (1) conversation-centric interaction styles, including support for conversation management; (2) core functionality through the words

**Table 1.1** Types of natural-language interaction styles**System-Centric Style**

like voice control or web search  
 within-turn context  
 two-turn interaction model  
 recognizes commands or queries  
 user-driven  
 voice or visual

**Content-Centric Style**

like frequently asked questions (FAQ)  
 within-turn context  
 recognizes questions  
 document-like, topic-based responses  
 user-driven  
 visual only

**Visual-Centric Style**

like desktop or mobile interfaces  
 within-turn context  
 requires direct manipulation  
 buttons, menus, lists, images, etc.  
 agent-driven  
 visual only

**Conversation-Centric Style**

like natural conversation  
 cross-turn, persistent context  
 recognizes conversational actions  
 fine-grained back-and-forth  
 mixed initiative  
 voice or visual

alone, whether voice, text, or both; and (3) compatibility with multiple platforms, from voice-only and simple chat to desktop and large displays. In this manner, starting with conversation enables a designer to consider progressive enhancement [[Gustafson 2015](#)], so that greater functionality can be increasingly implemented as the modalities of conversation evolve.

To summarize, the four natural-language interaction styles exhibit multiple distinctive features (Table 1.1). Each style is an ideal type, and multiple styles may be mixed in the same application.

## 1.4

### Mutual Understanding

Although we can build user interfaces that in some ways work like natural conversation, can these machines really understand us? As we saw above with ELIZA, a chatbot may sometimes continue a conversation without understanding the substance of what the user is saying. Today's chatbots and voice assistants are often characterized as "AI" (artificial intelligence) or "cognitive" and as capable of "understanding natural language" so that the user can speak to them as he or she would to another person. The user does not need to learn a specialized set of commands, as in the days of command-line interfaces. These metaphors clearly suggest that machines can think and understand. But what does this mean? Does it suggest that computers have consciousness, private thoughts, or empathy? What does it mean to say that a computer "understands" the user? To explore this question, we briefly draw on philosophy and social science to specify what we mean when we say that humans "understand" each other.

### Perspectives on Human Understanding

From a commonsense view, we often conceive of "understanding" as a mental state, a psychological phenomenon. For example, if I picture an elephant in my mind and tell you to "picture an elephant," then an image of an elephant appears in your mind too. If we both imagine the same animal, then we would say that we "understood" each other. But if I had an image of a rhinoceros in my mind and incorrectly said "elephant," then we have not understood. According to this notion, understanding is first and foremost an overlap of mental states or contents.

While this is a commonsense view of understanding, similar, more sophisticated versions have been proposed by philosophers and scientists. For example, according to the picture theory of meaning [Wittgenstein 1922], language consists of complex, or "molecular," propositions that can be broken down into "atomic" propositions, which can be broken down no further. These atomic propositions then correspond to "atomic facts" in the world. If I observed an elephant swaying to live music, my mind would decompose that reality into its atomic facts, find the corresponding atomic propositions, and transmit them to you through the complex proposition, "I saw an elephant swaying to live music!" Your mind then reverse engineers the complex proposition into its atomic propositions, and from the corresponding atomic facts, your mind forms a complex image similar to what I saw.

While the commonsense notion of understanding or even the picture theory of meaning seem plausible at first glance, many scholars have questioned them. Even the philosopher of language who formulated the picture theory later argued that

he was wrong. In his later work, Ludwig [Wittgenstein \[1953\]](#) argued that shared understanding cannot be achieved from private mental states, like the image of a swaying elephant. I have no way of knowing what you pictured in your mind, but I make judgments about whether you understood nonetheless. How can I do this?

[Wittgenstein \[1953\]](#) argues that shared understanding is in the first place the result of social practices, or “language games.” Rather than consisting of atomic propositions, the meaning of a word or phrase is actually its *use* in some social practice or situation [[Wittgenstein 1953](#), #43]. According to this view, understanding becomes a social event or sociological phenomenon. To demonstrate how meaning and understanding are based on social practice, Wittgenstein (1953), in part, used numerous examples of imagined interactions between math teachers and students. We attempt to summarize his argument with just two imagined examples of our own (Example 1.8).

**Example 1.8** Non-Pedagogical Inquiries.

- 1 Teacher: Do you know what the frequency of human speech is?
- 2 Student: Yes
- 3 Teacher: Do you know how it differs from pitch?
- 4 Student: Uh-huh

Teachers never do this, at least not when they are testing a student’s understanding of a concept. They do not because to answer these questions adequately is not to *demonstrate* understanding of the concepts of “frequency” and “pitch” but merely to *claim* it. As [Weizenbaum \[1966\]](#), p. 43] points out, “The crucial test of understanding, as every teacher should know, is not the subject’s ability to continue a conversation, but to draw valid conclusions from what he is being told.” Instead, testing a student looks more like this (Example 1.9).

**Example 1.9** Pedagogical Inquiries.

- 1 Teacher: What is frequency in human speech?
- 2 Student: The number of complete cycles of variations in air pressure occurring per second.
- 3 Teacher: How is it different from pitch?
- 5 Student: Pitch is the human perception of frequency.
- 6 Teacher: What’s a similar example of the relationship between frequency and pitch?
- 8 Student: That between wavelength and color.

This method of testing, sometimes called *viva voce* (or “oral examination”), invites the student to give definitions and explanations of the concepts, which in turn reveal the student’s interpretations or misinterpretations. Wittgenstein would say

that in such “language games,” when the student produces a correct or adequate explanation, the teacher says he or she “understands” the concept. Conversely, the teacher says he or she “misunderstands” or “partially understands” when incorrect or inadequate explanations are demonstrated. That is how we *use* the term “to understand” in practice. That is how it functions in social interactions. Understanding and misunderstanding are not attributed when an image appears privately in the student’s mind. The teacher has no access to that, only to the student’s actions, demonstrations, performances. In other words, this is what *understanding* is, from a practical or functional perspective. It is a kind of outcome or achievement of an interaction, of a social practice.

When it comes to understanding in the context ordinary conversation, sociologist Harold [Garfinkel \[1967\]](#) points out another dimension. While the social practice of *viva voce* involves spending time probing the limits of a student’s understanding of topics, conversation in everyday life mostly does not. In the midst of everyday activities, we have places to go and things to do; there is “no time out” to probe and test the other person’s underlying interpretations. So for practical considerations, we instead *assume* that the other person understands until further notice, that is, until there is some evidence to the contrary.

[Garfinkel \[1967\], p. 42](#)] demonstrated this through a simple exercise: “Students were instructed to engage an acquaintance or a friend in an ordinary conversation and, without indicating that what the experimenter was asking was in any way unusual, to insist that the person clarify the sense of his commonplace remarks.” The student experimenters overwhelmingly met with resistance ([Example 1.10](#)).

**Example 1.10** Norm Breaching.

- 1 Subject: How are you?
- 2 Student: How am I with regard to what? My health, my finances, my school work, my peace of mind, my...?
- 3
- 4 Subject: Look I was just trying to be polite. Frankly, I don't give a damn how you are.
- 5

Here, as in accounts of other students, probing understanding of ordinary expressions and utterances quickly drew reprimands from the subjects. What is normal in the classroom with technical expressions is abnormal outside the classroom with ordinary expressions. “The anticipation that persons *will* understand, the occasionality of expressions, the specific vagueness of references, the retrospective-prospective sense of a present occurrence, waiting for something later in order to see what is meant before, are sanctioned properties of common discourse”

[[Garfinkel 1967](#), p. 41]. A level of uncertainty and vagueness in ordinary conversation is expected and trying too hard to remedy it will be seen as breaching a basic social trust.

But one implication of [Garfinkel's 1967](#) exercise is that on any particular occasion, understanding may be assumed incorrectly. [Weizenbaum \[1966\]](#), p. 42] likewise suggests that speakers in “real life” may falsely attribute “background knowledge, insights and reasoning ability” to their recipients. Understanding in conversation is rarely definite. It may be demonstrated, faked or assumed. And it can only be probed or tested through further interaction, and even that will be abandoned once the tester is satisfied for all practical purposes. This *functional* conception of understanding is much messier than the picture theory.

Starting from this functional notion of shared understanding, sociologist Emanuel Schegloff went a step further and asked, how do people do it? By analyzing detailed transcripts of naturally occurring human conversations, Schegloff and his colleagues [[Schegloff et al. 1977](#), [Schegloff 1992b](#)] identified and formalized the methods that people use to achieve shared understanding, or “intersubjectivity.” They specify the “machinery” through which people design their talk for their particular recipient(s) and then deploy a set of methods for “repairing” troubles in understanding only if they emerge. Consider the following invented exchange (Example 1.11).

**Example 1.11** Understanding Repair.

- 1 Speaker A: How are you?
- 2 Speaker B: Fine.
- 3 Speaker A: No, I mean, do you still have symptoms?
- 4 Speaker B: Oh, yeah my knee is hurting me.
- 5 Speaker A: Okay.

In this case, Speaker B displays an interpretation of A’s prior inquiry, “How are you?,” as the kind of *welfare check* typical in the openings of conversations by responding with “Fine.” However, in the next turn, or “third position,” Speaker A initiates repair on his initial utterance by marking it as such, “No, I mean,” and paraphrasing it, “do you still have symptoms?” Speaker B then provides a different kind of response. Thus Speaker B initially displayed misunderstanding of A’s inquiry, but then understanding following A’s repair. We will talk more about this infrastructure for achieving shared understanding in conversation in the chapters that follow.

## Machine Understanding

Given this discussion of understanding, we now return to the question, “Can machines understand?” According to the commonsense theory of mind, machines cannot understand people because they cannot conjure an image as it appears in a human speaker’s mind. Computer algorithms work differently from human brains so whatever the machine does internally is not understanding. “Computers cannot think because they are machines” [Button et al. 1995, p. 149]. According to this view, “understanding” is a uniquely human ability.

However, we come to a very different conclusion if we conceive understanding as the outcome of social practice. According to a functional conception of understanding, a machine like a human can *understand* if it can *do*. “The grammar of the word ‘knows’ is evidently closely related to that of ‘can,’ ‘is able to.’ But also closely related to that of ‘understands.’” [Wittgenstein 1953, #150]. If a machine could successfully play the role of the student in a *viva voce*, then we could use the term “understanding” to characterize that interactional outcome or performance.

Computer scientist Alan Turing argued a similar point in his discussion of whether machines could think and in his formulation of an “imitation game,” today known as the “Turing test.”<sup>3</sup> Turing [1950, p. 433] replaces the question of whether machines can “think” with the question of whether they can “imitate” humans. Where the former question leads to endless philosophical arguments, the latter question can inspire the development of computer technologies. No doubt it has, in part, inspired scientists at IBM to develop Deep Blue, a computer that can imitate the playing of chess, and Watson, a computer that can imitate the playing of the trivia game Jeopardy. And of course it inspired the development of ELIZA, a computer that can engage in Rogerian-style conversation.

---

3. Button et al. [1995, p. 145] acknowledge that Wittgenstein could “certainly” be “aligned with the functionalists” and that the “Turing test might be seen as instantiating” his “understanding-as-performance” view, but they then dismiss it as “the most superficial reading of Wittgenstein.” However, their supporting argument depends on a straw man. They replace the “understanding-as-performance” view with a superficial version of it in which single performances of apparent understanding are taken at face value. As a result, they claim, inauthentic performances cannot be distinguished from authentic performances, for example, a student being fed answers by someone else from one who actually knows the material. But of course these *are* routinely distinguished *through further performances*. If the teacher suspects such cheating, he or she arranges a new test, under different conditions. *Viva voce* has the advantage that the questions are not predetermined and the questioning can be adapted on the fly to the student’s responses, making understanding harder to fake. And at some point, the teacher will stop eliciting new performances and make a practical judgment of the student’s understanding.

Turing [1950], p. 446] points out that, from a functional perspective, we employ *viva voce*, with its unpredictability and probing, to “discover whether someone really understands something or has ‘learnt it parrot fashion.’” If a computer can engage successfully in *viva voce* with a clever and thorough teacher, we should say that it “understands” the topic in question, even though there is the possibility that we are being fooled, just as with human students. A machine’s level of understanding is judged by its performances in multiple situations, under varying circumstances. Similarly, if a computer can engage in conversation with a user, we can judge its ability to understand conversational topics, as well as to understand conversational actions themselves. The value of the Turing test is not to fool subjects into thinking that a machine is a human but to enable them to compare the performance of the computer with the performance of a human. It redirects our attention from the philosophical question to the technical one.

So in principle, we argue that computers can potentially understand without thinking, just as humans can understand without thinking. However, today’s computers are far from demonstrating general or deep understanding. Creating a computer that can engage in the kind of *viva voce* of educational settings is still a hard problem. Returning definitions and explanations, like “What is frequency in human speech?” is easy enough, but performing real-time conceptual analysis, like, “What’s a similar example of the relationship between frequency and pitch?” is much harder. Today’s chatbots may demonstrate limited understanding in narrow tasks but fall apart when probed in different contexts or even with different phrasing. Even with the possibility of functional machine understanding, achieving it with real systems, at levels comparable to humans, may not ever be feasible. Such is the challenge of *general AI*.

But if we acknowledge Garfinkel’s (1967) observation that shared understanding in ordinary conversation is usually only deep enough for all practical purposes, then machines may be able to understand humans well enough in many kinds of narrow conversational settings, such as service encounters. In service encounters, the conversations tend to be highly repetitive and the usual goals relatively narrow. Customer service agents typically answer inquiries or fulfill recurrent requests or troubleshoot predictable problems within a limited domain. Even though today’s chatbots and voice assistants cannot handle domain-independent *viva voce*, they may be able to understand what the user says and does well enough to answer inquiries, fulfill requests, or troubleshoot problems for all practical purposes.

Furthermore, in order to do this, machines must be able to engage in the repair practices that Schegloff and colleagues demonstrate. Natural language understand-

ing (NLU) techniques, while necessary, are not enough. They provide only for the machine to interpret the user's utterances. But such interpretations must be tested in interaction before understanding can be determined and must be repaired if misunderstanding or partial understanding is displayed. Thus conversational systems also need natural conversation understanding (NCU), or the ability to engage in repair practices, as specified by Schegloff [[Schegloff et al. 1977](#), [Schegloff 1992b](#)]. We will return to this topic in Chapters [4](#) and [6](#).

To summarize, we offer a definition of *understanding* that we will assume throughout the remainder of this book. Understanding is not the same thing as interpretation. Interpretation is the analysis of the language and the action of an utterance, but *understanding is the demonstration of correct or adequate interpretation of social action within interaction*. Otherwise *thinking* one understands would be the same thing as understanding.

Weizenbaum [[1966](#), p. 43] admits that ELIZA does not understand what the user is saying, but rather "maintains the illusion of understanding." However, while speakers may on occasion conceal misunderstanding from each other, in general, achieving mutual understanding is a fundamental goal of natural conversation. Weizenbaum also acknowledges that the display of misunderstanding or partial understanding is critical for natural conversation: "A switch of objectives from the concealment to the revelation of misunderstanding is seen as a precondition to making an ELIZA-like program the basis for an effective natural language man-machine communication system." Conversational agents must display their interpretation of what the user has said or did, as well as of the topic at hand, so that users can see if the agent understands them and if not, can attempt to repair it.

## 1.5

### About This Book

The goal of this book is to inspire a new form of user experience (UX) design for the development of conversational agents. Our intended audience is the UX designer working on applications with natural-language interfaces, such as chatbots, virtual agents or voice assistants. We are speaking to the UX designer who gets his or her hands dirty authoring dialog logic and agents' utterances, rather than relying on a system developer or content designer to determine the user experience. We trust that those who have attempted to design the user experience for conversational agents have found, as we have, that such applications demand something different from what is needed for other kinds of desktop or mobile applications. *The user experience consists not primarily of the manipulation of buttons, menus and windows,*

*but in the sequencing of utterances in ways that seem “natural” or “conversational.”*

As a result, a more formal knowledge of how human conversation is structured, turn by turn, seems invaluable.

To this end, we aim to provide a guide and a reference for conversational UX designers that includes an overview of conversation science, assets for UX design, and a process to use in development projects. First, we review principles and models from the field of Conversation Analysis (CA) to provide a theoretical foundation for understanding *human conversation* (Chapter 2); this material will be unfamiliar to most UX designers. Second, we provide a brief overview of *conversation authoring* using the Intent-Entity-Context-Response (IECR) paradigm (Chapter 3), which will be familiar to those who have created dialog flows on platforms like Amazon Alexa, Google Assistant, and IBM Watson Assistant. Third, we outline a new approach to conversational UX design, the Natural Conversation Framework (Chapter 4), which adapts principles, concepts and models from Conversation Analysis to the design of natural-language interfaces. Fourth, we provide a “pattern language” [Alexander 1977, Erickson 2000] of conversational UX design patterns for common conversational activities, as well as conversation management (Chapters 5–7), which we intend readers to use as a reference when designing conversational UX. Finally, we help the reader put all that they have learned from previous chapters into practice by outlining a general *design-thinking* process and adapting it for the unique challenges of conversational UX design (Chapter 8). By the end of this book, the reader should be prepared to design conversational user experiences that draw from a scientific foundation in Conversation Analysis and that use a systematic design framework.

# Conversational UX Design

*A Practitioner's Guide to the Natural Conversation Framework*

Robert J. Moore, Raphael Arar

With recent advances in natural language understanding techniques and far-field microphone arrays, natural language interfaces, such as voice assistants and chatbots, are emerging as a popular new way to interact with computers. They have made their way out of the industry research labs and into the pockets, desktops, cars and living rooms of the general public. But although such interfaces recognize bits of natural language, and even voice input, they generally lack conversational competence, or the ability to engage in natural conversation. Today's platforms provide sophisticated tools for analyzing language and retrieving knowledge, but they fail to provide adequate support for modeling interaction. The user experience (UX) designer or software developer must figure out how a human conversation is organized, usually relying on commonsense rather than on formal knowledge. Fortunately, practitioners can rely on conversation science.

This book adapts formal knowledge from the field of Conversation Analysis (CA) to the design of natural language interfaces. It outlines the Natural Conversation Framework (NCF), developed at IBM Research, a systematic framework for designing interfaces that work like natural conversation. The NCF consists of four main components: 1) an interaction model of "expandable sequences," 2) a corresponding content format, 3) a pattern language with 100 generic UX patterns and 4) a navigation method of six basic user actions. The authors introduce UX designers to a new way of thinking about user experience design in the context of conversational interfaces, including a new vocabulary, new principles and new interaction patterns. User experience designers and graduate students in the HCI field as well as developers and conversation analysis students should find this book of interest.

## ABOUT ACM BOOKS



ACM Books is a series of high-quality books published by ACM for the computer science community. ACM Books publications are widely distributed in print and digital formats by major booksellers and are available to libraries and library consortia. Individual ACM members may access ACM Books publications via separate annual subscription.

ISBN 978-1-4503-6302-0

9 0000



9 781450 363020