



Back to the Future: Lessons from Knowledge Engineering Methodologies for Chatbot Design and Development

Cameron, G., Cameron, D., Megaw, G., Bond, RR., Mulvenna, M., O'Neill, S., Armour, C., & McTear, M. (2018). Back to the Future: Lessons from Knowledge Engineering Methodologies for Chatbot Design and Development. In R. Bond, M. Mulvenna, J. Wallace, & M. Black (Eds.), *Proceedings of the 32nd International BCS Human Computer Interaction Conference (HCI-2018)* BCS Learning & Development Ltd. <https://doi.org/10.14236/ewic/HCI2018.153>

[Link to publication record in Ulster University Research Portal](#)

Published in:

Proceedings of the 32nd International BCS Human Computer Interaction Conference (HCI-2018)

Publication Status:

Published (in print/issue): 10/05/2018

DOI:

<http://dx.doi.org/10.14236/ewic/HCI2018.153>

Document Version

Author Accepted version

General rights

Copyright for the publications made accessible via Ulster University's Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The Research Portal is Ulster University's institutional repository that provides access to Ulster's research outputs. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact pure-support@ulster.ac.uk.

Back to the Future: Lessons from Knowledge Engineering Methodologies for Chatbot Design and Development

Gillian Cameron
Ulster University
Belfast, N. Ireland
g.cameron@ulster.ac.uk

David Cameron
Inspire Workplaces
Belfast, N. Ireland
d.cameron@inspirewellbeing.org

Gavin Megaw
Inspire Workplaces
Belfast, N. Ireland
g.megaw@inspirewellbeing.org

Raymond Bond
Ulster University
Belfast, N. Ireland
rb.bond@ulster.ac.uk

Maurice Mulvenna
Ulster University
Belfast, N. Ireland
md.mulvenna@ulster.ac.uk

Siobhan O'Neill
Ulster University
Belfast, N. Ireland
sm.oneill@ulster.ac.uk

Cherie Armour
Ulster University
Belfast, N. Ireland
c.armour1@ulster.ac.uk

Michael McTear
Ulster University
Belfast, N. Ireland
mf.mctear@ulster.ac.uk

This paper explores the area of conversational user interfaces and chatbot development, focusing on the methodological aspects of development. The domain in this paper for chatbot development is healthcare. An increasing issue in chatbot development relates to the difficulty in eliciting specific domain knowledge. As chatbots become more ubiquitous in our daily lives with more complex use cases, the process of eliciting and codifying the domain knowledge has become more complex. This is a problem revisited; in the 1980's, 'expert systems' grew rapidly in popularity and such systems required the same processes of elicitation and codification of human know-how or expertise as we now re-witness in modern chatbot development. A new area of 'knowledge engineering' developed from the expert systems or 'knowledge-based systems' field and from this several knowledge engineering methodologies emerged. The present paper revisits these methodologies and asks if there are lessons to be learned for chatbot design and development from such decades old knowledge engineering methods. The paper presents an amendment to a chatbot methodology, incorporating new stages of 'knowledge gathering' and 'usability testing' into the process.

Chatbot Development, Chatbot Methodology, Knowledge Engineering, Chatbot Lifecycle, Chatbot Usability, Chatbot Scripting, Knowledge Elicitation, Expert Systems, Mental healthcare chatbots.

1. INTRODUCTION

Chatbots or conversational user interfaces are becoming more prevalent in our daily lives, with bots available to check the weather [12], plan holidays [15] and even talk with a virtual therapist [8]. Følstad and Brandtzæg predict that soon users may prefer to engage with such conversational user interfaces to complete activities that are traditionally done through a web page, or a mobile application [9]. Facebook Messenger is an example of a platform where the user can interact with bots, and there is no need to switch between multiple apps, therefore reducing the cognitive load for the user [3].

It is now easier than ever for users to communicate with chatbots through voice activated devices, with numerous channels for accessing the bot, including Slack, Facebook Messenger, and web channels. In mental health, Fitzpatrick, Darcy and Vierhile studied the user of chatbots on college students who

identify as having symptoms of anxiety and depression [8]. Their chatbot, 'Woebot' draws from Cognitive Behavioural Therapy techniques. Rizzo et al. developed SimCoach, a virtual human support agent that promotes access to psychological healthcare information [14].

2. CHATBOT DEVELOPMENT

Amir Shevat describes bots as a new kind of user interface, that allows users to interact with brands and services using their favourite messaging apps [21]. A chatbot can also be seen as a complete application, with back-end logic and a conversational interface being developed through a whole life-cycle. In this paper, we are referring to the development of the conversational interface and the back-end logic components. Chatbots can be developed rapidly using bot platforms such as motion.ai, Botsify, and Chatfuel. These platforms

can be utilised by beginners or non-technical users, as they require no coding [4]. However, these tools can be limited, and using a bot development framework can allow the developer to implement more complex functionality. A bot development framework equips the developer with a set of predefined functions and code to allow them to develop bots easier and faster. An example of a development framework is the Microsoft Bot Framework. The Microsoft Bot Framework incorporates Microsoft's Cognitive Services, an Application Programming Interface (API) that can process natural language, enable a chatbot to recognise speech, and image-processing technology that can recognise images [10].

2.1 Current best practice in chatbot development

The chatbot development lifecycle (Figure 1), contains many stages, such as requirements gathering and architecting, which are also followed when developing traditional software programs [5].

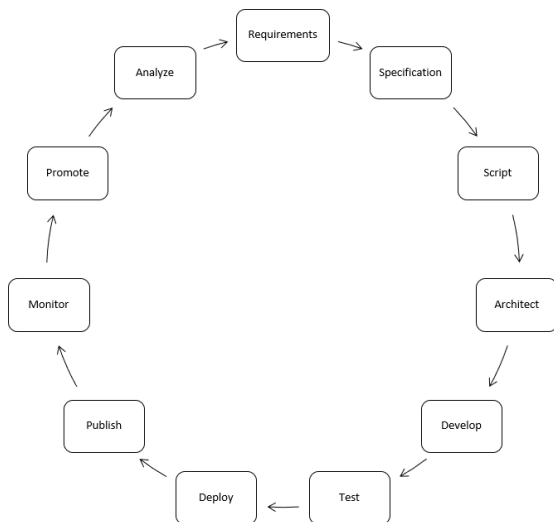


Figure 1 – Chatbot Lifecycle (Chatbots Magazine)

Within the Specification stage, the chatbot scope should be defined. Recast AI put forward that an enterprise chatbot scope should contain 4 elements: a main use case that covers the core functionality of the bot, an extended use case which facilitates broader user requests, default questions the user might ask about the company/maker of the chatbot, and finally the handling of small talk [13].

There are several unique steps that make developing a chatbot different. Scripting is a unique step in the lifecycle compared to traditional software development, as chatbots have a conversational interface. Instead of building wireframes, the focus needs to be placed on building conversational scripts [5]. The Developing and Testing stages, although similar to traditional software development, are highly interlinked in chatbot development. Furthermore, chatbots require extended testing, as

there are numerous messaging applications that users can utilise. Post deployment, conversational logs need to be monitored to improve the usability and conversational flow of the chatbot. In this lifecycle, the usability is tested by examining the conversational logs. However, this does not identify any possible pain points a user may experience when interacting with the chatbot. These can be identified by performing cognitive walkthroughs, and observing the user interacting with the chatbot.

Furthermore, a fundamental step has been left out of the lifecycle, which is knowledge gathering. This is especially relevant when developing chatbots for healthcare disciplines, as accuracy of knowledge is key to the validity, safety and veracity of the chatbot.

3. KNOWLEDGE ENGINEERING

Knowledge engineering is traditionally concerned with the development of expert systems, where knowledge plays a vital part in helping to aid decision making [17]. Expert systems have two main components, a knowledge base which consists of knowledge elicited from an expert in the domain, encoded into facts, rules and relationships, and an interface that elicits this knowledge appropriately [23]. Knowledge engineering is now seen as a modelling activity, and the process is cyclic in nature [22]. In the 1980s, knowledge engineering was limited by the size and power of computers, and a lack of trained knowledge engineers [7].

A knowledge engineer is required to obtain key concepts from the domain expert, to create a schema for the knowledge base. The knowledge engineer then needs to elicit, structure, formalize and add the knowledge into the knowledge base. Experts and sources can differ on knowledge, and “tacit” knowledge, which is knowledge that is difficult to articulate, can be difficult to obtain and map [20]. Different communication styles and use of language can create assumptions and ambiguities in the knowledge base. These issues can be lessened by using a knowledge engineering methodology.

3.1 Knowledge Engineering methodologies

In order to facilitate knowledge engineering, several different methodologies have been developed. In this paper, CommonKADS is described. These types of methodologies for knowledge engineering may be adapted for use in developing the back-end logic component of the chatbot.

CommonKADS and its predecessor KADS (Knowledge Analysis and Documentation System) was developed from a series of research projects [18]. CommonKADS is a collection of structured methods for identifying, capturing and modelling knowledge to be incorporated into knowledge systems. Expert knowledge is modelled, as well as

how to use this knowledge. Before KADS and CommonKADS, knowledge-based systems were developed through trial and error, but there was a clear need for guidelines and techniques to aid the knowledge engineering process [19]. There are 6 models involved in CommonKADS; Organization, Task, Agent, Knowledge, Communication, Design.

The Organizational Model consists of key components, business processes, structural units and business resources, and their relationships. The Task Model consists of the different tasks needed to perform an operation. The Agents Model allows the mapping of key agents involved, human or software, and their capabilities that are required to perform tasks. The Communication Model identifies the transactions that are required between agents during operation. The Knowledge Model is used to identify the expertise required to perform the operation. Finally, the Design Model encompasses the design of the Knowledge Based System.

The main model in the CommonKADS methodology is the Knowledge Model, which can be described using a Conceptual Modelling Language. Different types of knowledge are categorised as domain knowledge, interface knowledge and task knowledge. Domain knowledge includes explicit knowledge, concepts, properties of concepts and their relationships. Interface Knowledge describes the steps required for the reasoning process using the domain knowledge. Task Knowledge refers to what goals an application pursues, and the steps taken to achieve these goals. Parallel development of these different models is encouraged. CommonKADS allows for the creation of a complete knowledge model, independent of implementation.

3.2 Lessons from knowledge engineering methodologies for chatbot development

Knowledge engineering methodologies use an iterative approach, using a knowledge analysis phase to help specify a subsequent knowledge systems development stage. As chatbots are constantly refined based on the conversational logs, and new knowledge added, this practice can be applied. This knowledge analysis stage is equivalent to chatbot scripting while knowledge systems development is analogous to developing the scripts into decision trees used to guide conversations.

Knowledge engineering methodologies facilitate the creation of a knowledge base that can exist independently from the development of the system. This aspect could be beneficial for chatbot development, as there are many platforms to deploy chatbots to, and different types such as text-based, and voice-based chatbots. Ontologies can be used to name and formally define the types, properties, and relationships between entities within a particular domain. They include all relevant constraints between classes, attribute values, instances and

relations. Al-Zubaide and Issa built a chatbot using ontologies to turn knowledge into a relational database, to drive the conversation [1].

4. PROPOSED CHATBOT DEVELOPMENT LIFECYCLE

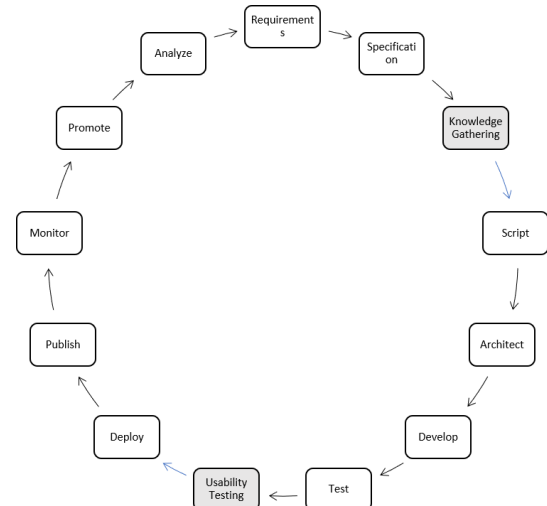


Figure 2 - Proposed chatbot development lifecycle

Figure 2 shows a proposed chatbot development lifecycle, which has two steps added, namely *knowledge gathering* and *usability testing*.

4.1 Requirements and Specification

At the requirements step, the overall goal and benefit of the chatbot is established, and high level requirements are set by the domain expert, such as if it needs to be text-based, or voice-based. Using the requirements, a specification is written that outlines the functionality of the chatbot.

4.2 Knowledge Gathering and Scripting

An essential step in developing chatbots for healthcare is knowledge gathering. Similar to expert systems, chatbots for healthcare need to contain knowledge from domain experts, and representations of the relationships between pieces of knowledge to create a useful and robust chatbot. Holding interviews with experts to retrieve different types of knowledge is necessary in order to map out a knowledge base schema. Knowledge can then be retrieved from the domain expert, structured and formalized before inputted into the knowledge base.

In the context of the work reported in this paper, interviews were held with a domain expert in the area of mental health. These interviews established the structure of knowledge in the form of coping mechanisms. A web based form was co-designed with the domain experts, which allows them to input coping mechanisms directly into the database that is linked to the chatbot. Crowdsourcing can be facilitated using this form, as multiple domain experts can utilise it. The domain expert can add

coping mechanisms for mental health issues directly to the database using this form, which then populates the chatbot with usable knowledge. The domain expert can input the content of their coping mechanism, the level of intensity and what category is applicable, such as stress or anxiety. The knowledge is structured appropriately, to ensure the correct information is given to the user.

Scripts are created to fulfil a goal, such as extracting information from the user, or giving the user information. The chatbot scripts are created using the knowledge base, and then passed back to the domain expert for validation, and refined until the expert thinks the script can meet the specified goal. Knowledge gathering and scripting are two steps that are interlinked, as the scripts are continuously refined, and further knowledge is added to the knowledge base. Aspects of film script development and analysis can be adapted to analyse chatbot scripts. Analytical thinking can be organised into categories: Synopsis, Premise, Structure, Character, Dialogue, Visual Grammar, Pace, Conclusion [16]. A synopsis, to briefly define what the chatbot does, and relay this back to the user. Structure to evaluate the order of the script, where a conversation should end and begin to keep the user engaged. The character is the bot, that is given a personality and can use humour and emotion to improve the user experience. The dialogue should be consistent with this personality.

In the context of the work ongoing in this project, passing the script back and forth to the domain expert has allowed the conversation to be validated, and improved on each iteration.

4.3 Architect, Develop and Test

The knowledge engineer/developer takes the requirements, and chooses the development platform that will be able to fulfil them. The front-end and the back-end components are defined before development. Candello and Pinhanez [2] used the Wizard of Oz technique, where a human simulates the response of a system, to help shape the design of their chatbot. As noted before, development and testing are heavily interlinked.

4.4 Usability testing

The usability of a chatbot can be improved by following best practices [11], such as onboarding users appropriately to ensure the user knows what the bot can do. Multimodal feedback, such as spinning wheels when content is loading, or indicators to give the illusion that the bot is typing, can notify the user that an action is being taken. If the chatbot fails to understand a user's request, reiterating what the chatbot can do can be a helpful way of getting the conversation back on track. Incorporating a consistent personality into the

chatbot, that uses humour and emotion in an appropriate context can improve the user experience. Engaging with conversational experts can improve the conversational flow, as it is important not to force the user down one conversational pathway.

Before deploying the chatbot, it should be tested with representative users. Chatbottest has created a chrome extension, Alma, which tests 7 chatbot design areas, personality, onboarding, understanding, answering, navigation, error management and intelligence [6].

4.5 Deploy and Publish

Once developed and tested, the chatbot can then be deployed and published in the chosen environment.

4.6 Monitor and Analyse

Once the chatbot gains users, monitoring conversational logs can allow the developer to pinpoint any further conversation breaks that are not discovered at the usability testing step. After analysing the conversational logs, the scripts can be refined, and any dropped conversation points addressed, repeating steps in the lifecycle to continuously improve the chatbot.

This lifecycle has been developed from a project on designing and developing a chatbot for mental health care. The database linked to the chatbot is provisioned with coping mechanisms that have been derived from a domain expert on the project team. The conversational flow has been developed using this repository of coping mechanisms and validated by the domain expert. Using this lifecycle has enabled an easier development process, ensuring the chatbot system is validated and robust.

5. CONCLUSIONS

This paper has examined the area of chatbot development, focusing on the methodological aspects of development in the domain of mental healthcare. A key area in chatbot development is the increasing requirement for codification of sometimes complex, domain specific knowledge. This issue has been experienced in the past when expert or knowledge based systems were commonly developed using basic knowledge 'building blocks' or facts, rules, relationships and frames for more structured knowledge. In order to address the maintenance, quality and design issues in the development of these systems, knowledge engineering methodologies evolved to cater for the field. This paper has revisited some of these methodologies and it is proposed that two new stages of 'knowledge gathering' and 'usability testing' are incorporated into the process.

6. REFERENCES

- [1] Hadeel Al-Zubaide, Ayman Issa. 2011. Ontbot: Ontology based chatbot. In Proceedings of Innovation in Information & Communication Technology (ISIICT'11), 7-12.
- [2] Heloisa Candello, Claudio Pinhanez. 2017. Designing the user experience of a multi-bot conversational system. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, ACM.
- [3] Chatbot's Life. 2018. ChatBots—The Rise of Conversational UI. Retrieved March 01, 2018 from <https://chatbotslife.com/chatbots-the-rise-of-conversational-ui-8a59078e2f95>
- [4] Chatbots Magazine. 2018. How To Develop a Chatbot From Scratch. Retrieved January 22, 2018 from <https://chatbotsmagazine.com/how-to-develop-a-chatbot-from-scratch-62bed1adab8c>
- [5] Chatbots Magazine. 2018. The Bot Lifecycle. Retrieved January 24, 2018 from <https://chatbotsmagazine.com/the-bot-lifecycle-1ff357430db7>
- [6] Chatbottest.com. 2018. The free guide for you to understand what is your chatbot doing wrong. Retrieved February 8, 2018 from <http://chatbottest.com/>
- [7] Edward A Feigenbaum. 1980. Knowledge engineering: The applied side of artificial intelligence (No. STAN- CS-80-812). STANFORD UNIV CA DEPT OF COMPUTER SCIENCE.
- [8] Kathleen Kara Fitzpatrick, Alison Darcy, Molly Vierhile. 2017. Delivering cognitive behavior therapy to young adults with symptoms of depression and anxiety using a fully automated conversational agent (Woebot): a randomized controlled trial. *JMIR mental health*, 4, 2: e19.
- [9] Asbjørn Følstad, Petter Bae Brandtzæg. 2017. Chatbots and the new world of HCI. *Interactions* 24, 4: 38-42.
- [10] Microsoft Azure. 2018. Cognitive Services. Retrieved January 22, 2018 from <https://azure.microsoft.com/en-us/services/cognitive-services/>
- [11] Robert J. Moore, Raphael Arar, Guang-Jie Ren, and Margaret H. Szymanski. "Conversational UX Design." In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, ACM, 2017.
- [12] Poncho. 2018. Wake Up Weather. Retrieved January 22, 2018 from <https://poncho.is>.
- [13] Recast.AI. 2018. Building Great Bots: An Enterprise Chatbot Methodology. Retrieved January 24, 2018 from <https://recast.ai/blog/enterprise-chatbot-methodology/>
- [14] Albert A. Rizzo, Belinda Lange, John G. Buckwalter, Eric Forbell, Julia Kim, Kenji Sagae, Josh Williams, Barbara O. Rothbaum, JoAnn Difede, Greg Reger, Thomas Parsons, Patrick Kenny. 2011. An intelligent virtual human system for providing healthcare information and support. *Studies in Health Technology and Informatics*, 163: 503-509.
- [15] Justin Sablich. 2018. How to Plan Your Next Vacation With a Chatbot. Retrieved January 22, 2018 from <https://www.nytimes.com/2017/01/17/travel/vacation-travel-chatbot-booking.html>
- [16] Lucy Scher. 2012. Reading Screenplays: How to Analyse and Evaluate Film Scripts. Oldcastle Books.
- [17] Guus Schreiber. 2000. Knowledge engineering and management: the CommonKADS methodology. MIT press.
- [18] Guus Schreiber, Bob Wielinga, Joost Breuker. 1993. KADS: A Principled Approach to Knowledge Based System Development. Academic Press.
- [19] Guus Schreiber, Bob Wielinga, Robert de Hoog, Hans Akkermans, and Walter Van de Velde. 1994. CommonKADS: A comprehensive methodology for KBS development. *IEEE expert* 9, 6: 28-37.
- [20] Nigel Shadbolt, Nick Milton. 1999. From knowledge engineering to knowledge management. *British journal of management* 10,4: 309-322.
- [21] Amir Shevat. 2017. Designing Bots: Creating Conversational Experiences. O'Reilly Media, Inc.
- [22] Rudi Studer, V Richard Benjamins, Dieter Fensel. 1998. Knowledge engineering: principles and methods. *Data & knowledge engineering*, 25, 1-2: 161-197.
- [23] Chuck Williams. 1986. Expert systems, knowledge engineering, and AI tools-an overview. *IEEE expert* 1, 4: 66-70.