Wharton University of Pennsylvania | William and Phyllis MACK INSTITUTE for INNOVATION MANAGEMENT

Reimagining Customer Service Journeys with LLMs: A Framework for Chatbot Design and Workflow Integration

Lennart Meincke and Christian Terwiesch[1]

January 23, 2025
Mack Institute for Innovation Management, The Wharton School, University of Pennsylvania

Introduction

Whether assisting travelers with inquiries about their upcoming flight or destination, providing mental health support, helping students with homework assignments, or aiding in financial planning, chatbots powered by generative artificial intelligence (GenAI) in the form of large language models (LLMs) can help deliver efficient, scalable, and personalized customer support. Building an effective chatbot involves navigating a series of design decisions that influence user experience, technical implementations, and various legal and ethical issues. This white paper outlines a general framework for the functionality of chatbots and their operational use. Our aim is to help executives navigate the complex decisions that need to be made when leveraging the power of LLMs to support their customers, be they travelers, patients, students, or investors.

In the first part of this paper, we articulate five design choices that need to be made when building a chatbot. These choices provide the answers to five important managerial questions:

1. Focused vs. Broad Knowledge Base ("How much does the chatbot know?")
2. Isolated Interactions vs. Long-time Relationships ("Does the chatbot remember users over multiple episodes?")
3. Proactive vs. Responsive ("Does the user reach out to the chatbot or the chatbot to the user?")
4. Static vs. Dynamic ("Does the chatbot learn about a user or a user population over time?")
5. Quality Assurance ("How is the quality of the chatbot's output assured?")

These five dimensions create a taxonomy of chatbots.

In the second part of the paper, we discuss different modes of interactions between the chatbot and the human operators. Indeed, a successful use of chatbots is more than just a matter of designing and building a new technology. Rather, the chatbot must be integrated into the workflow of the organization. At the

[1] Meincke, Terwiesch: The Wharton School, 500 Huntsman Hall, 3730 Walnut Street, Philadelphia, PA 19104, lennart@sas.upenn.edu, terwiesch@wharton.upenn.edu

30,000-foot level, such modes of interactions are broadly referred to as “humans in the loop” processes. But, as we will explain, there are many ways to design a workflow in which a human operator collaborates with a chatbot to support a customer.

Specifically, we distinguish between the following six workflow configurations, all of which have a human in the loop: configurations 1 and 2 are often referred to as AI-enabled auditing, configurations 3 and 4 are also known as AI-assisted workflow, and configurations 5 and 6 correspond to fully automated workflows.

1. Human operator response with offline auditing by chatbot
2. Human operator response with real time auditing by chatbot
3. Chatbot recommendations with human operator deciding and responding
4. Chatbot preparation with human operator responding
5. Chatbot response with real time auditing by human operator
6. Chatbot response with offline auditing by human operator

The decisions along the five chatbot design dimensions and the six workflow configurations provide executives with a blueprint to plan their efforts for reimagining customer support. Many other decisions, including privacy and security considerations as well as system integration, will also need to be made but are beyond the scope of this paper

Part I: The Five Dimensions of Chatbot Design

Dimension 1: Focused vs. Broad Knowledge Base
How much does the bot know?

A chatbot with a broad knowledge base can be created using frontier models such as Open AI’s GPT-4 (OpenAI et al. 2024) or Google’s Gemini (Gemini Team et al. 2024). Thanks to extensive training data, these chatbots can answer a wide range of questions without any form of special training, and more recently, can ingest text, documents, audio and video.

As powerful as these frontier models might be, they do lack depth in specialized areas, especially when it comes to topics that highly depend on specific terms and operating policies. A high-profile example is Air Canada’s chatbot, which was reported to have offered discounted airfares to a customer requiring an urgent flight due to a death in the family. Though many airlines offer such bereavement tickets, Air Canada at that time did not (Melnick 2024). The fact that the Air Canada chatbot was trained on the broad body of knowledge typical for a frontier model (including the policies of many other airlines) as opposed to being focused exclusively on the Air Canada policies led to a frustrated customer and bad publicity.

Technical considerations. From a technical perspective, using an existing model like GPT-4, which has been trained on extensive public data, is the simplest approach to start using LLMs. Depending on the nature of the task, some hyperparameter tuning, such as picking a lower (more deterministic) or higher (more creative) temperature value might be helpful. For instance, when computing the dosage of a specific

medication, a more deterministic approach is likely to be preferred. To guide chatbot behavior, the most useful tool is the "system prompt," which specifies the rules and persona of the bot and what it can and cannot do. A teaching assistant bot, for example, should perhaps only guide a student through a problem, but not reveal solutions too quickly. Customizing a bot's abilities via the system prompt is straightforward and can be done entirely using free-text instructions. While results are often impressive, extensive testing is necessary to evaluate instruction and adherence and whether the model performs well enough without any special knowledge.

If more specialized knowledge is needed, implementing a retrieval augmented generation (RAG) approach can help integrate specific data sources, like product information or operating policies, into the bot's knowledge base.

RAG works by taking a set of documents, breaking them down into smaller pieces (chunks), and storing them in a database. When processing a user question, the query is compared to all the knowledge pieces to determine the most relevant ones to help answer the question. When searching for related documents, ideally the bot only considers highly relevant information, reducing the amount of text it has to process which improves response time and can help with accuracy as information is less likely to be overlooked.

For example, a customer support chatbot for a retailer should probably be aware of return policies ("Is there a longer return window for purchases during the holiday season?"). A "plain vanilla" GPT model is of little value in such cases just as in the Air Canada example. However, overloading the system prompt with this type of information can degrade response performance and is often impractical due to context window limitations. As such, allowing the bot to refer to external information can vastly improve its knowledge base. RAG is especially useful when external information changes rapidly, as the data storage can be quickly updated independently of the LLM. Depending on the data sources needed for RAG, e.g. customer records in Salesforce, additional engineering work is necessary to translate LLM requests to meaningful lookups in the respective databases.

Many intricate details can greatly impact RAG performance. The original documents need to be split into parts that are not too big (and hence too general) and not too small to lack sufficient detail.[i]

Dimension 2: Isolated Interactions vs. Long-time Relationship

Does the bot remember users over multiple episodes?

Most bots start an interaction with a user with a "clean slate" ("Hi, I am a virtual assistant; what can I do for you today?"). Each interaction stands alone and there is no memory from one interaction to the other. This works well for a myriad of inquiries, such as asking about historic events or gift ideas.

In contrast, a relationship-focused bot can offer personalized assistance based on past interactions ("Great to see you today! I know you were wondering about opening an account last week. We just increased our interest rate for new customers. Do you want to learn more?").

The potential for a chatbot to form a long-term relationship with a customer is enormous. Imagine, for example, a user with a learning disability or someone who simply struggles with specific mathematical content (e.g., the concept of compound interest rates). Not only might a relationship-focused bot customize its answers to a form that proves effective over time with the user, such as the terminology used, but it could also help diagnose that the user is experiencing difficulties in the first place. The bot will "remember" what the user struggles with and thereby can adapt its interactions.

A long-term relationship might also make the user more comfortable interacting with the bot. Though such increased comfort comes at the risk of "humanizing" the chatbot and entering an unhealthy relationship (a phenomenon that has been reported especially among teenagers, Roose 2024), there exists enormous potential for customization in utilizing the history of prior interactions.

Technical considerations. It is simple not to store any user history ("stateless"), as only the current question needs to be processed by the bot. No additional resources for storage or authentication are needed. For long-term relationships, the challenge lies with storing past interactions and providing relevant pieces to the chatbot, so it remembers key facts. Due to context window limits (the amount of text an LLM can remember), it is often not feasible to provide the entire conversation history in the prompt to the LLM. Then, a second step is necessary that extracts meaningful insights from previous conversations and summarizes them to provide context for future conversations. In addition, certain customer actions or information from other sources might also be provided as part of the context to the LLM to improve the long-time relationship. Extensive testing is necessary to establish how much information can be provided before performance deteriorates, such as the LLM starting to forget previous facts or responses becoming slower and more expensive since more text needs to be processed with each query.[ii] This is especially important for cases where the customer might expect that all previous conversation history is considered for future requests; clear communication is necessary in cases where this is not possible.

Dimension 3: Proactive vs. Responsive

Does the user reach out to the chatbot or the chatbot to the user?

Most chatbots are responsive, i.e., they wait until the user takes the initiative and approaches the chatbot with a request. However, there is no reason why the chatbot shouldn't be proactively taking the initiative ("Hi Joe, your upcoming flight to Paris is in 3 days. It looks like it will be rainy for the first few days. Do you want to learn more about must-visit indoor spots in the city?"). Such proactive bots might increase engagement by offering timely information or reminders, something that is well-studied in the medical domain (Volpp et al. 2017, Lekwijit et al. 2024).

Currently, most chatbots do not reach out proactively by themselves. There are several reasons for this. In general, chatbots mimic the workflow of a customer support center or a help desk, which is by nature responsive. In addition, deploying a responsive chatbot is very simple on a technical level, as that is precisely what the chatbots are designed to do.

Some recent examples of companies moving toward proactive chatbots include platforms such as Character.ai that have experimented with chatbots sending messages to users who have not engaged with

them for a while. Such proactive chatbots require a more complex technical setup where bots periodically "wake up" to reach out to the customer. In the travel context, the outreach could be triggered by specific events, such as upcoming flights, changes in travel advisories, or be customer-specific behaviors, such as checking in with a customer who usually books specific upgrades.

Technical considerations. Building a proactive chatbot requires integrating calendar systems, monitoring tools, and customer relationship management (CRM) systems. The challenge lies in designing a system that balances proactive engagement without becoming intrusive, potentially requiring advanced user preference management and context-awareness capabilities. A proactive chatbot is generally not possible without knowing the user (dimension 2) and might often require additional data from an external system.

In most cases, this requires system integration and thus additional development resources. Lastly, the medium of outreach can play a big role — nudging via email might be less effective than via a text message or app notification, which would require further development resources to design a mobile app. Data and privacy concerns arise in these cases and should be carefully considered and addressed to ensure user trust and compliance.

Dimension 4: Static vs. Dynamic

Does the bot learn about a user or a user population over time?

A static bot operates with a fixed set of rules or knowledge. For a given question, a static bot always provides the same answer. This might, however, miss important opportunities. For example, an investor who is requesting her tax documents likely has very different preferences in April (when most Americans file their taxes) compared to November. A dynamic chatbot adapts its behavior based on external factors or evolving data.

The choice between static and dynamic bots influences how the bot adapts over time and depending on the user. A static bot uses a system prompt and potentially auxiliary systems, such as RAG, to provide information. While it can refer to previous conversations if those are preserved (see dimension 2), it has no representation of time and its influence on customer needs and changes in the surrounding world.

Technical considerations. A simple way to make the chatbot dynamic is by updating the system prompt periodically to include new knowledge, such as the time to the next Tax Day. An even more dynamic approach could involve retrieving the investor's personal tax schedule. Just as discussed previously, this would require access to additional databases, such as the organization's CRM system.

In addition to learning more about an individual customer over the course of a longer relationship, we can also imagine chatbots learning from other customers (population-level learning, see Siggelkow and Terwiesch 2019) and using this knowledge to improve the quality of the support. In this case, rather than using an existing frontier model, the organization would train its own model based on the incoming support requests and their resolutions ("we noticed that customers like you who faced similar problems benefited from doing xyz").

Dimension 5: Quality Assurance

How is the quality of the bot's output assured?

The final dimension in our chatbot design framework focuses on quality assurance. LLMs can sometimes produce erratic output and are prone to hallucinations, which can be more or less problematic depending on the use case. A chatbot supporting travelers that wrongly attributes the origin of a local dish is not as bad as a chatbot that does not take the user's allergies into account when recommending a medication.

Generally, we need to consider two types of defects. First, random hallucinations are instances where the bot generates false or nonsensical information without apparent reason; it just happened not to know or chose a bad token. Second, a user might intentionally try to manipulate (jailbreak) the bot into performing actions against its programming or the programmer's intent. For example, a student might try to convince a teaching bot that their life depends on passing the course and therefore the bot should provide the full answer to all problems in the class immediately.

The complexity of quality assurance increases with the chatbot's generality (dimension 1). A more focused chatbot is easier to validate, while a general-purpose assistant requires more comprehensive measures. We propose three main approaches to quality assurance.

First, the organization can decide to proceed with no formal quality assurance. This approach works well for low-stakes situations, such as providing recommendations for a holiday trip or ideas for an entertainment event. Here, the focus is on setting clear expectations with users about the bot's limitations and potential for errors. The simplest strategy to mitigate most risks is a robust system prompt that clearly defines the bot's boundaries and ethical guidelines. For instance, a bot can be clearly told never to agree to any price discussions. Even in these cases, however, designers should implement extensive testing scenarios to understand the range of responses and test with "edge cases." The system prompt can also include predefined templates that the AI can fill in to reduce variability, such as asking the LLM to always output its final answer in specific tags (<answer>) after it has reasoned about what to do.

Second, the organization can rely on chatbot-based quality assurance by using another LLM (from the same or from a different frontier model) as an auditor. This method employs an additional chatbot to assess the primary chatbot's output relative to the user's prompt. The secondary chatbot acts as a validator, checking for inconsistencies, hallucinations, or inappropriate content. Complex instructions can lead to the first chatbot not always following all best practices, so a secondary bot checking for the most glaring issues can be a helpful and simple-to-implement defense strategy. The auditor chatbot requires a clear set of criteria and possible actions, such as rewriting parts of the response or asking the first bot to regenerate its answer, to provide an effective safeguard. For instance, a user that managed to trick a complex first chatbot into revealing the answer to a homework problem might find it much harder to also trick the second chatbot that is merely told to "ensure the following response never reveals the full solution" instead of having to also adhere to many other instructions. In theory, one can chain many chatbots to improve overall answer quality.

However, each additional chatbot in the chain impacts response times since the previous chatbot needs to finish generating its response, leading to a sequential dependency. The user request is first sent to the first chatbot in the chain, alongside the system prompt and previous conversation. Then, the full response from the first chatbot is awaited. Once completed, the second (next) chat bot can review the answer based on its instructions. If it is the last chatbot in the chain, the revised answer can be streamed (sending smaller paragraph chunks) instead of awaiting full completion to improve responsiveness. For simpler rules, such as stripping out specific words, a buffer for the response of the first chatbot could be used so that words can be validated before they are returned as part of a streaming response.

The third strategy for quality assurance is to "put a human in the loop," something that we will discuss at length in the second part of this paper.

Part II: Integrating chatbots into workflows

How much autonomy is granted to the chatbot and how much human labor should remain in the workflow depends on the business goals behind the chatbot implementation. In this second part, we will first discuss what organizations might aspire to get out of a chatbot implementation followed by different workflows that determine how human operators interact with the chatbot.

Business Objectives

Broadly speaking, a chatbot can exist for two reasons. First, automation can lead to efficiency gains while maintaining the same level of quality / providing a similar customer experience. When resetting a password or updating a mailing address, for example, a chatbot can achieve the same outcome for a fraction of the cost of a human to manually make such changes].

Second, automation can also be used to enhance the customer experience and the perceived quality of the service provided. For example, a chatbot can be used to engage patients and increase their compliance with their medication regiment in a way that would just not be possible with a non-automated process. Or a student studying geometry can be tutored by a chatbot and obtain a learning experience that otherwise might only be available for those students who can afford private tutoring. Another advantage of the chatbot is that the support of the customer can typically be provided immediately, saving the customer from spending endless time in a queue waiting to be served by a human.

When deciding to implement a chatbot, it is critical to know that they can affect the accuracy of a service and its ability to consistently adhere to a quality standard. For example, hallucinations are a common quality concern for chatbots. Also, customers might have an inherent preference for human operators and human operators might be more knowledgeable and better able to resolve a customer problem. Humans might also be averse to receiving advice from a chatbot (Dietvorst et al. 2015) even though they might be unable to tell the difference between a human and an AI-powered response (Meincke et al. 2024).

On the other hand, chatbots never get tired, are always equally friendly, and can be very knowledgeable if designed accordingly (dimension 1). They also have been shown to be rather charming and persuasive. A recent study comparing chatbots with doctors, for example, found that the chatbots were not only more competent in their diagnosis, but also were perceived as being more empathetic by the patients (Ayers et al. 2023).

As the organization chooses between the following workflows, it should evaluate the most promising process designs alongside their business objectives.

Workflow Configurations: Designing Processes with "Humans in the Loop"

Be it for quality reasons or other business objectives outlined above, fully automated chatbots with absolutely no human oversight are rare and probably are nothing to be desired. For this reason, we now turn to our six workflow configurations that specify how human operators collaborate with GenAI technology. Our six configurations might remind the reader of the levels of autonomous driving. However, while autonomous driving levels are increasing in the responsibility the AI has in driving the car (from level 0: fully human to level 5: fully AI-based) all of our six configurations rely on an interplay between humans and AI. What changes is the division of labor between the two: in configuration 1, the human operator performs the work and the AI focuses on quality assurance and feedback while in configuration 6 it is exactly the opposite. Figure 1 shows the responsibility distribution for response creation and response evaluation performed by the human and agent for each configuration.

We illustrate the six configurations for the hypothetical scenario of a patient seeking help in preparing for an upcoming surgery. In such a scenario, the patient may want to know what and until when she is allowed to eat, when she should arrive at the hospital, and when she can expect to go home and resume her normal activities.

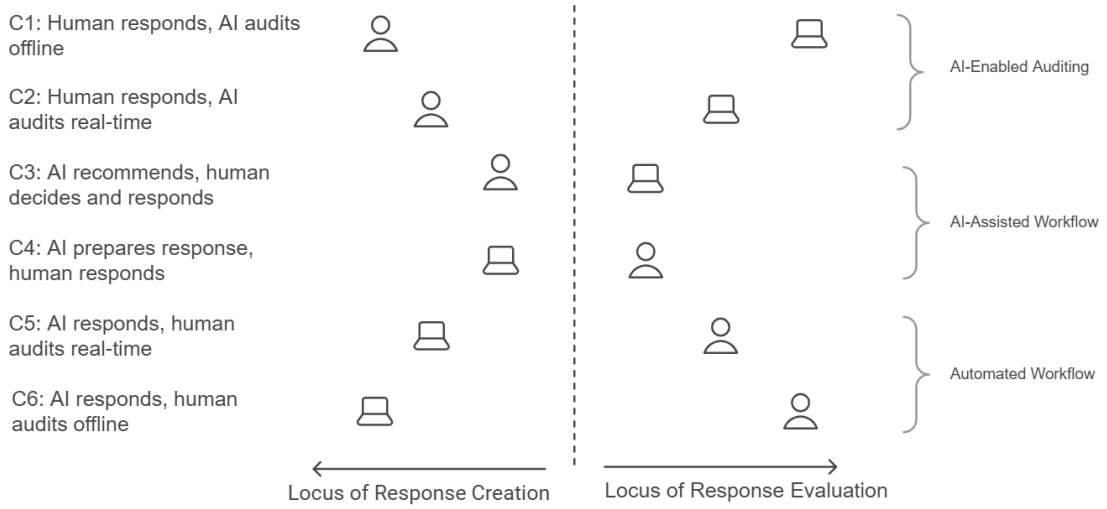

Figure 1: Responsibility distribution for response creation and response evaluation between human and AI

Configuration 1: Human operator response with offline auditing by chatbot

In this first configuration, all the work related to responding to the customer is carried out by the human operator. In the case of our example, the hospital operates a call center or allows patients to contact their care team so that they can ask them questions such as “When is the last meal I am allowed to take?”, “How long will I have to rest after surgery?”, or “Can I have coffee the morning of my surgery?”. The human operator directly provides an answer.

Note that this configuration still allows for some use of AI. For example, if the organization wants to evaluate the accuracy of the medical advice provided by their human operators, it could periodically audit the chats. In such an offline audit, the hospital could use an LLM to identify possible mistakes performed by the human operators and bring them to the attention of management. Such offline auditing by an LLM is not unique to configuration 1 and can be applied to any configuration.

In short, Generative AI is used to improve the accuracy of the customer support and thereby reduces defects and enhances learning about common questions and mistakes. This defect reduction and additional learning should translate to higher efficiency and better customer satisfaction in the long run.

Configuration 2: Human operator response with real-time auditing by chatbot

The second configuration is the same as the first but features real-time AI oversight, instead of offline auditing, of the human operator during the customer support session by AI. Returning to our example, if a human operator provides incorrect or incomplete information to the patient (e.g., the operator gives the wrong co-pay information to the patient or fails to alert the patient that she will not be able to operate a vehicle right after the surgery), an LLM listening to the call or following the chat could instantly pick up the mistake and alert the human operator to it while the conversation with the patient is still ongoing. While this is a technically more complex implementation than many other configurations, it still provides the most autonomy to the human agent outside of configuration 1

Just like the first configuration, the immediate efficiency gains are relatively low. After all, the customer support is still carried out by human operators. However, the immediacy of the feedback that is now happening in real time leads to fewer defects and faster learning.

These first two configurations are most relevant for transactional tasks that require relatively little cognitive work or active problem solving by the customer support person. However, they may be preferred when dealing with high-risk or high-compliance tasks, as they leave most of the autonomy with the human operator. Next, we will look at the role LLMs can play in helping with more challenging support requests.

Configuration 3: Chatbot recommendations with human operator deciding and responding

Returning to our scenario, consider a patient calling the hospital with a medical problem. The hospital or the healthcare network might have hundreds or even thousands of providers. Each provider differs in their

specific expertise, their geographic location, their availability of appointments, and so on. Which of these should be proposed to the patient?

In the case of the third configuration, a GenAI tool can leverage the available information about the patient and the providers to recommend a few options to the human operator. The operator in turn has the responsibility to choose from these options. In other words, the “heavy lifting” is done by generative AI (narrowing down the choice set from thousands to a handful), but the final decision and responsibility rests with the human. Moreover, the human operators might have to add “a finishing touch” to the solution they recommend, such as explaining the choice to the patient.

This configuration plays to the key strengths of an LLM: it is good at generating options, but, due to hallucinations, it benefits from having a human be the last point in the customer support journey.

Configuration 4: Chatbot preparation with human operator responding

This configuration is similar to configuration 3, except that the LLM only recommends one solution. Its focus therefore is not on generating alternatives for the human operator to choose from, but rather to prepare the final answer to the customer as much as possible.

In our use case, imagine the LLM preparing a customized instruction message to the patient preparing for surgery. The message might include the arrival time, when to take (or not take) specific medications the patient is on, and post-surgical instructions. The role of the human operator in this case is to simply read and approve the message and potentially make minor edits.

This configuration is similar to the use case of medical providers using an LLM to summarize an encounter with a patient and then only reviewing the documentation for accuracy before storing it in the patient’s electronic health record rather than typing up the report from scratch. Or, think of a radiologist who uses GenAI to read an image and prepare a first draft of the report automatically but who makes the final sign-off on the report.

The key value proposition is to significantly lower the touch time of the human operator, thereby improving the efficiency of the customer support organization while leaving the “final word” with the human operator. Moreover, from the customer’s perspective, the experience is one of directly interacting with a human and receiving personalized support.

In configurations 3 and 4, the chatbots are doing a lot of the heavy lifting but ultimately the human operator is responsible for providing the customer support. In the final two configurations, we see the primary responsibility for providing customer support shifting from human operators to the chatbots, with the human role becoming more supervisory.

Configuration 5: Chatbot response with real-time auditing by human operator

Configuration 5 moves even more of the work to the chatbot. While in configuration 4 human operators still applied some “finishing touches” to the support request, in configuration 5 they only oversee the work of the chatbot. Such an oversight might involve a formal approval of a response back to the customer or handling an exception where the chatbot cannot help. This reduces the touch time dramatically, allowing one human operator to oversee multiple chatbots in parallel.

Ideally, the chatbots are capable of explicitly calling for human intervention when they are uncertain about a particular customer request. This could be achieved by having a second chatbot monitoring the interaction with the customer and, in real time, alert the human operator to intervene or provide explicit tools to the first chatbot that make it aware of its capabilities.

Configuration 5 has much larger efficiency potential than configuration 4. The challenge is to determine how many chatbots a single human operator can oversee at the same time. In our example, patients submit requests to the chatbot on the hospital’s website and the chatbots provide a response that, before being posted in the chat, needs to be approved by the human operator.

Configuration 6: Chatbot response with offline auditing by human operator

The highest level of autonomy is realized by providing the chatbot with the authority to handle a customer support request without a human in the loop. Returning to our example, patients approach the chatbots with questions and the chatbot provides the answers. This approach reduces the touch time to zero and thus has the highest efficiency potential.

However, the fact that there is no human in the loop does not imply that there should be no oversight. In configuration 6, management audits the automatically executed chats periodically, potentially with the help of an LLM to refine strategies and improve the customer support experience. During these audits, management can get a sense (though with a delay) about the quality of the support provided and what changes might be necessary.

Configuration 7: Hybrid configurations and implementation

A seventh configuration to consider is a combination of any of the previously discussed configurations. An organization might roll out a GenAI initiative by taking 100% of its calls using configuration 1 (human operators only) and use the accumulating data to fine-tune an LLM.

Then, as a second step, it might conduct a set of experiments that would confront chatbots with the situations encountered by the human operators and estimate a level of confidence with which the chatbots are handling requests correctly.

It is also conceivable that an organization might deploy different configurations based on the nature of the support request. For high-risk or high-compliance tasks, configurations 1 or 2 might be the best option, as they leave most of the autonomy with the human operator. For complex decision-making tasks, configurations 3 or 4 can remove cognitive burden from the human operator and allow them to quickly

arrive at the ideal solution. In scenarios with high volumes but simpler tasks, configurations 5 and 6 might work well.

Conclusion

GenAI, primarily in the form of large language models, has already begun transforming customer support. From travel and tourism to healthcare and education to financial services, chatbots based on LLM's have the potential to make customer support a higher-quality experience for the customer while also improving the efficiency for the organization providing it.

In our discussions and experience in several industries, we were surprised to see that the biggest challenge in the transition of GenAI tends not to be a legal or technical challenge but rather for management to develop a vision of how GenAI could be deployed. Before starting the technical development of a chatbot, executives need to ask themselves what type of bot they would like to get.

The goal of this paper is to help executives develop such a vision by reimagining customer support. We have presented five dimensions of chatbot designs and six configurations specifying workflows how humans and GenAI collaborate to provide customer support.

Our five dimensions of bot design and our six configurations together create a menu of design options. This menu can be used for ideating the needed vision. Should an organization work on a chatbot that has broad knowledge, recognizes a customer over a longer relationship, and is able to learn from past interactions and then deploy it by having a human operator interacting with the customer while getting real-time LLM-powered oversight? Or should the organization pursue a chatbot that is focused on particular support problems, has no memory of past interactions, but is able to autonomously interact with customers with only episodic quality auditing?

In our view, there is not one best design or workflow configuration. Instead, it is management's role to systematically explore the potential designs and configurations and use the business objectives described above to find the most promising customer support vision for their own organization.

Acknowledgements

We thank Vibhanshu Abhishek, Martin Bittner, Lilach Mollick and Hummy Song for their helpful comments.

Endnotes/References

Anthropic (2024) Introducing Contextual Retrieval. Retrieved (October 4, 2024), https://www.anthropic.com/news/contextual-retrieval.

Ayers JW, Poliak A, Dredze M, Leas EC, Zhu Z, Kelley JB, Faix DJ, et al. (2023) Comparing Physician and Artificial Intelligence Chatbot Responses to Patient Questions Posted to a Public Social Media Forum. *JAMA Intern Med* 183(6):589–596.

Bastani H, Bastani O, Sungu A, Ge H, Kabakcı Ö, Mariman R (2024) Generative AI Can Harm Learning. (July 15) https://papers.ssrn.com/abstract=4895486.

Cevasco KE, Morrison Brown RE, Woldeselassie R, Kaplan S (2024) Patient Engagement with Conversational Agents in Health Applications 2016–2022: A Systematic Review and Meta-Analysis. *J Med Syst* 48(1):40.

Chen JT, Huang CM (2023) Forgetful Large Language Models: Lessons Learned from Using LLMs in Robot Programming. (October 10) http://arxiv.org/abs/2310.06646.

Dietvorst BJ, Simmons JP, Massey C (2015) Algorithm aversion: People erroneously avoid algorithms after seeing them err. *Journal of Experimental Psychology: General* 144(1):114–126.

Faysse M, Sibille H, Wu T, Omrani B, Viaud G, Hudelot C, Colombo P (2024) ColPali: Efficient Document Retrieval with Vision Language Models. (July 2) http://arxiv.org/abs/2407.01449.

Gemini Team, Anil R, Borgeaud S, Alayrac JB, Yu J, Soricut R, Schalkwyk J, et al. (2024) Gemini: A Family of Highly Capable Multimodal Models. (June 17) http://arxiv.org/abs/2312.11805.

Lekwijit S, Terwiesch C, Asch DA, Volpp KG (2024) Evaluating the Efficacy of Connected Healthcare: An Empirical Examination of Patient Engagement Approaches and Their Impact on Readmission. *Management Science* 70(6):3417–3446.

Meincke L, Nave G, Terwiesch C (2024) The AI Ethicist: Fact or Fiction? (October 11) https://papers.ssrn.com/abstract=4609825.

Meincke L, Carton A (2024) Beyond Multiple Choice: The Role of Large Language Models in Educational Simulations. (May 26) https://papers.ssrn.com/abstract=4873537.

Melnick K (2024) Air Canada chatbot promised a discount. now the airline has to pay it. *The Washington Post*. Retrieved (December 7, 2024), https://www.washingtonpost.com/travel/2024/02/18/air-canada-airline-chatbot-ruling/.

OpenAI, Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, Aleman FL, et al. (2024) GPT-4 Technical Report. (March 4) http://arxiv.org/abs/2303.08774.

Roose K (2024) Can A.I. be blamed for a teen's suicide? *The New York Times*. Retrieved (December 7, 2024), https://www.nytimes.com/2024/10/23/technology/characterai-lawsuit-teen-suicide.html.

Sadasivan C, Cruz C, Dolgoy N, Hyde A, Campbell S, McNeely M, Stroulia E, Tandon P (2023) Examining Patient Engagement in Chatbot Development Approaches for Healthy Lifestyle and Mental Wellness Interventions: Scoping Review. *J Particip Med* 15:e45772.

Siggelkow N, Terwiesch C (2019) *Connected Strategy: Building Continuous Customer Relationships for Competitive Advantage* (Harvard Business Press).

Volpp KG, Troxel AB, Mehta SJ, Norton L, Zhu J, Lim R, Wang W, et al. (2017) Effect of Electronic Reminders, Financial Incentives, and Social Support on Outcomes After Myocardial Infarction: The HeartStrong Randomized Clinical Trial. *JAMA Internal Medicine* 177(8):1093–1101.

---

[i] There are also many different choices of algorithms for comparing user queries to chunks stored in a database. When multiple pieces of information are relevant, they need to be ordered (ranked) to ensure that the bot focuses on the most pertinent data. To improve chunking performance, chunks can be infused with additional information (Anthropic 2024). For instance, homework questions on a business case could repeat key information from the case in each chunk to improve retrieval performance. Alternatively, screenshots of documents can be used instead of text (Faysse et al. 2024). This approach reduces the reliance on efficient chunking and leverages existing formatting in documents.

[ii] Prompt-caching can be an effective solution to improve responsiveness and lower costs for information that remains static over multiple LLM calls.