Using AI and chatbots to simplify finance tools at Microsoft

At Microsoft, many of our core business processes (such as procurement and payment) were built around a system-centric approach. Each of these apps was in its own silo, often disconnected from each other. Without a unifying, overarching strategy, each of these apps evolved independently without considering user experience in what we call the Procure to Pay process.

These complex and unwieldy apps required significant resources to maintain. The employee experience suffered because there wasn’t a single end-to-end process for Procure to Pay. Employees had to work within several apps—each with its own interface—to complete a task. That forced users to learn to navigate and complete manual steps between certain apps, making the process even more cumbersome. This in turn slowed completion of every Procure to Pay instance, making the feature expensive to maintain.

Ongoing efforts by Core Services Engineering and Operations (CSEO) to move toward a user-centric approach gave the Microsoft Finance Engineering team within CSEO an opportunity to rethink the Procure to Pay process by designing an AI-first, services-oriented architecture for it. The Procure to Pay process would focus on the end-to-end user experience and would replace the system-centric view of services.

Strategic approach

A connected user experience is what our employees are looking for. Our goal with bots is to simplify the user experience and provide important information with fewer clicks. It’s a challenge to create a truly conversational and interactive bot that’s able to seamlessly shift from one context to another and perform actions based on searching and on data. We started with a few fundamental goals and objectives:

- Determine the intent of the user, especially when multiple contexts or a wide range of user personas can be expected.
- Provide a seamless experience for users seeking information from a separate but related context.

We had to think about transformation. Bots have the potential to change the way our business engages with customers; we don’t think of a bot as something that simply replaces a human agent and works 24/7. Conversational AI enables a new model for engaging customers through fluid and frictionless conversations. We had to consider how the power of conversation can eliminate or automate some process tasks, remove friction, and transform how we address customer needs and give them a memorable experience.

Technical goals and implementation

As part of this initiative, we are consolidating all 16 discrete services that make up our Procure to Pay platform. Each service has a unique set of functionalities, presentation layer, APIs, and master data (Figure 1). In addition, we defined an overarching End-to-End User Experience layer that created a singular, unified experience.
Using AI and chatbots to simplify finance tools at Microsoft

Figure 1. Finance Digital Assistant architecture model is designed with scalability in mind and integration with many other vertical services.

The following technologies were used for implementation:

- **Azure Bot Service**
- **Microsoft LUIS (aka Language Understanding)**
- **QnA Maker**
- **Microsoft Azure Cognitive Services**
- **Microsoft Azure Stream Analytics**
- **Microsoft Azure Event Hubs**
- **Azure Service Fabric**
- **Microsoft Azure Text analytics**
- **Azure Databricks**
- **Azure Data Lake Storage**

A critical aspect of providing an end-to-end experience is automating the front- and back-office operations (such as support) as much as possible. To create this automation, our architecture incorporates a Procure to Pay support layer underneath all services. The Finance Digital Assistant (Figure 2) integrates into Procure to Pay services to monitor user activity and proactively offer guidance when deemed appropriate. Moreover, if the support bot can’t quickly resolve the issue, it will silently shift to a human supervisor who can interact with the user within the same support window. Our objective is to make the support experience so seamless that users won’t be able to tell when they’re interacting with a bot or a support engineer.

By using cognitive-sentiment analysis to detect the user sentiment, we built a seamless experience to transfer the user to a live agent. The live agent has the complete context of the user’s inquiry and can start the conversation when the bot fails to answer. A corresponding support ticket is auto-created for the user-bot conversation, allowing the agent to perform offline follow-up or post-call quality analysis.
Design considerations for bot implementation:

**Self-healing:** The bot has the ability to detect a failing service/API and escalate it to the respective DRI team, while providing an exceptional user experience to let them know of failing or failed services. Our services watch for failed dependencies and auto-heal once the failed services are back online.

**Telemetry and analytics:** Chatbots should log telemetry to collect metrics automatically and unobtrusively. Enhancement decisions can be made by detecting changes in telemetry trends and comparing trends. Telemetry will help answer some of the important questions around how our users interact, as well as frequency of use, duration, and what options are commonly selected.

**Proactive messaging:** Conversational AI enables a new model for engaging with your customers through fluid and frictionless conversations. With proactive messaging, we wow the customer and give answers without users having to ask for it. Using learning models, we trained our bot to understand the most common interactions and present the answers in proactive mode.

**Detailed reporting:** Business and engineering teams have various KPIs to measure. The KPIs should be identified and scoped in detail before creating a bot solution. We try to avoid siloed reporting solutions to make sure that the focus stays on solution goals rather than on KPI reporting. We used Azure Data Lake to bring the telemetry data from various systems to build the KPI reporting solution for each stakeholder.

**Multilanguage support:** Using Azure Cognitive Services for language translation, we enabled the bots to converse in more than 66 languages. Our bot solution auto-detects the language of the user and provides answers in that language. When a live agent (who is usually English-speaking) is needed, we use the same solution to present the question in English to the live agent and convert the answer to the user’s preferred language. This runtime dynamic translation breaks the barriers of language while working on support.

**Well-defined APIs to support transactions:** A bot is successful only when it’s built on well-defined API standards and supports the performance needs of conversational UI. APIs play a crucial role for the success of bots and, without that solution, the bots will fail.

**Multichannel support (web, Cortana, etc.):** The availability of bots on various channels make solutions discoverable for the customers and provide native experience where our customers are most comfortable. With Cortana, we were able to enable voice-to-question and answer-to-voice.
Self-learning and self-awareness: We added self-learning by integrating our telemetry and creating alerts on knowledge-base changes to frame questions to the bot. This helped users to not only articulate their question correctly but with fewer keystrokes. Self-awareness brought an ability to learn what other questions have been brought to the bot, and also enables the ability to highlight when new features/knowledge are added to address the evolving needs of our customers.

Lessons learned and best practices

Set clear objectives: Before implementing any strategy, it’s important to clearly define the business goals that the strategy sets out to achieve. Bots will likely play an important role in company strategy, and it’s also a long-term investment that needs to align with your organization’s vision.

Define ROI: In the long run, bots have the potential to improve customer experience, but it’s hard to quantify that benefit. So, make an economic argument for cost savings or revenue improvements, including any indirect savings because of automating the manual work.

Prioritize the use cases: Since bots are not created equal, it can be helpful to think of them in terms of different categories that can broadly identify their role and level of complexity. From simple-question/simple-answer interactions, such as the kind done with FAQ bots, to complex customer journeys like customer onboarding or data processing, the use cases for bots can enhance your customer experience and lower support-ticket cost.

Understand your users: It’s important to understand the needs of the users and possible scenarios. A chatbot’s ability to detect user intent and provide the answer in fewer words will result in higher customer satisfaction. Fine-tuning KB articles based on feedback keeps the bot content fresh and more relevant.

Fallback strategy: Have a backup plan—a contingency strategy that can be used if something goes wrong when bots answer complex questions. Having a strategy in place can elevate the user experience. For our part, we enabled the ability to chat with a live agent as well as the ability to auto-create a support ticket on the user’s behalf.

Factors that influence a bot’s success

Successful apps or websites have at least one thing in common: a great user experience. Bots are no different in that regard. Therefore, ensuring a great user experience should be your priority when designing a bot. Some key requirements include:

- The bot easily solves the user’s problem with the minimum number of steps
- The bot runs on the devices and platforms the user commonly uses.
- The bot is discoverable, and users intuitively know how to use it.

With our experience, we’ve found several application scenarios where bots are most effective:

- Large enterprise applications that utilize many user roles.
- Applications that lack the ability to show status or transaction reports for ad-hoc questions.
- Applications dependent on other applications where verifying the status of transactions is not intuitive.
- Applications not available on multiple channels (e.g., Skype, Microsoft Teams, email, Cortana)

Benefits

In the six months since we’ve implemented our AI initiative, we’ve received several business benefits:

- 20 percent reduction in support tickets during the three-month launch period, trending to reduce support tickets by 50 percent during the first full year
- Bot use is increasing rapidly, reducing the need for human intervention in requests
• Increased business-partner satisfaction

“In modern finance, the essence of improving and driving impact to our day-to-day roles is digital transformation. Our bot leverages AI and machine learning to drive out the waste of repetitive tasks and in return allows us to focus on strategic outcomes. Through collaboration and partnerships with our engineering partners to develop and implement intelligent bots, we can drive quick-turnaround solutions that are focused on the customer and, more importantly, our customers’ success!”

—Bonnie Cowan, Finance Director, Global Finance Operations

Conclusion

At Microsoft, rethinking our services using an AI-first approach to modernize architecture is helping us become a data-driven organization. By consolidating the user experience, we’ve created a self-serve platform on which anyone can consume the certified data and present it in a seamless, end-to-end manner to our customers. Following this successful implementation, our finance business partners have requested to bring this experience to all Microsoft Finance applications. We’re also offering new features, such as assisted-learning capabilities, pattern-matching to detect user input to application status (similar to courier tracking ID in Bing search), layered-questions capabilities, and much more.

“As we reimagine how we offer services to our customers, we need to put our customer lenses on and provide AI-driven simple solutions that stitches the seamless experience across the services.”

—Ashish Sayal, Principal Software Engineering Manager, Finance Procure to Pay, Microsoft

For more information

Microsoft IT Showcase
microsoft.com/itshowcase

Designing a modern service architecture for the cloud

Driving Microsoft’s transformation with AI

How automation is transforming revenue processing at Microsoft