INTWIXT

FAQ BLOG USER GUIDE

Five Strategies for Building Compelling Chatbots

May 23, 2017

by Luke Birdeau

Background

Messaging app potential is real. They're a viable deployment platform for solutions like *chatbots*, *instant apps*, or *whatever-you-want-to-call-them*. But deploying a solution atop a messaging app has its own unique opportunities and pitfalls when compared to other platforms like iOS, Android, and the Web. Whether you're building a customer-facing chatbot or an internal productivity tool for an enterprise, there are some principles we've found that aid in getting the most out of your chosen messaging app.

We decided to share these principles using Facebook Messenger as the messaging platform, but they apply to other platforms as well (like Slack or iMessage/SMS). To better illustrate each principle, we've created a Messenger bot named *Lists* (http://m.me/listsbot) that helps users create real-time, shared checklists. Lists permeate nearly every app we use (lists of ledgers in a banking app, lists of geo-markers on a map, lists of messages in a messaging app, etc). Try to think of your own use case as you consider the points below. Hopefully, you'll find the examples relevant. We've included a brief video to highlight some of these features as well.

1. Play To Your Platform's Strengths

Messaging apps have surpassed social networks in the time users spend online and in active user count. They're not just any old app on your phone. In terms of familiarity and reach, they have the characteristics of a platform--a viable choice for deploying solutions. But to leverage them successfully, remember to focus on their strengths. Messaging apps, for instance, may not have a mature *Bot Store*, but they do have something potentially more valuable, namely, an interconnected community of users.

As you consider your next deployment, focus on what makes bots shine, instead of delivering a substandard replacement for apps. Here are just a few opportunities we've found.

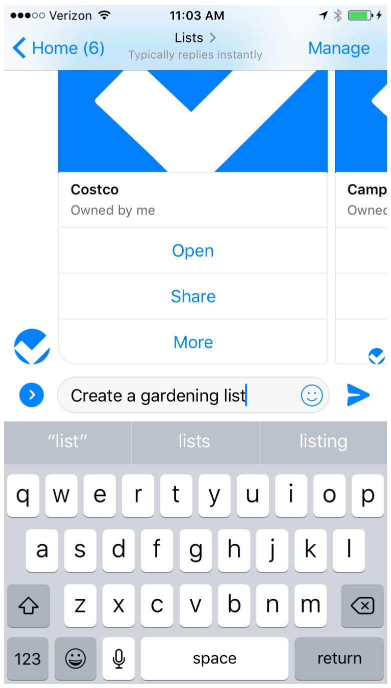
1. **Push-based deployment** | Avoid the friction of discovery, pre-installation, and updates.
2. **Process-oriented security** | Process-driven access provides clearly demarcated access to data that is time- and context-dependent.
3. **Robust notification engine and lightweight workflow** | Messaging apps already integrate well with device notifications and alerts, allowing them to be used for approvals and multi-role apps.
4. **Natural sharing and virality** | Share the app with family, friends, colleagues and relevant users already on your contact list.
5. **Baked-in identity management** | Avoid the friction of account creation, login and credential management (password reset).
6. **Market saturation and product familiarity** | Users are generally familiar with how to use at least one messaging app on their mobile

device. SMS predates even smart phones and is a viable bot channel.

2. Prioritize The Use Case

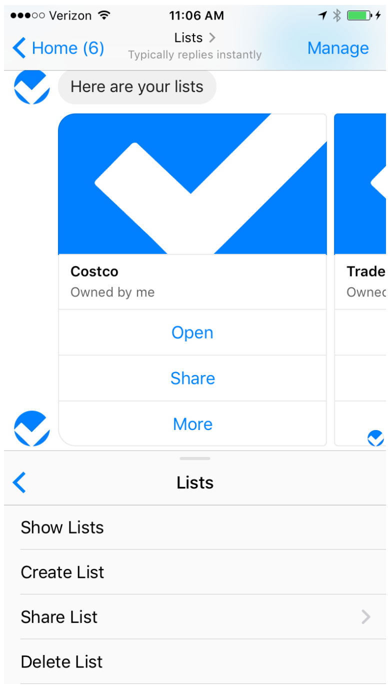
It’s a crowded market and your solution must be compelling. Don’t limit your reach by imagining that bots are conversational while apps are visual. Both sides would benefit to meet in the middle. It’s possible to deploy solutions atop Facebook Messenger with no facility for conversation, just as it’s possible to deploy a mobile app using nothing more than simple message bubbles. It’s really about the use case.

Our *Lists* bot surfaces its capabilities interchangeably, using both conversational and visual interfaces. A user can create a new list by engaging the bot conversationally (1). But they can also navigate a menu (2), or submit a form while using the visual embodiment (3). What matters is that all three interfaces present a common ontology in the user’s mind. The user just wants to create a list.



Create a gardening list

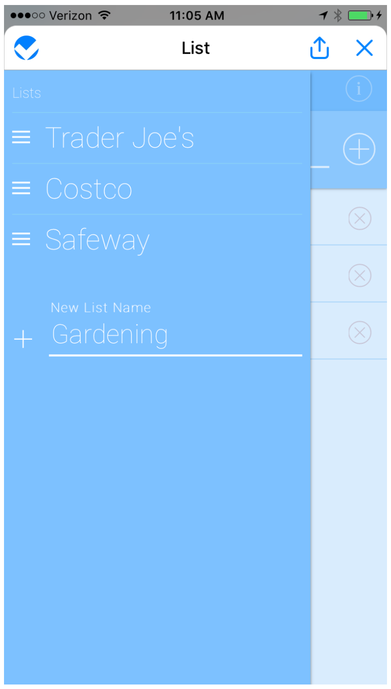
1. Conversational



Here are your lists

Show Lists
Create List
Share List
Delete List

2. Menu with link

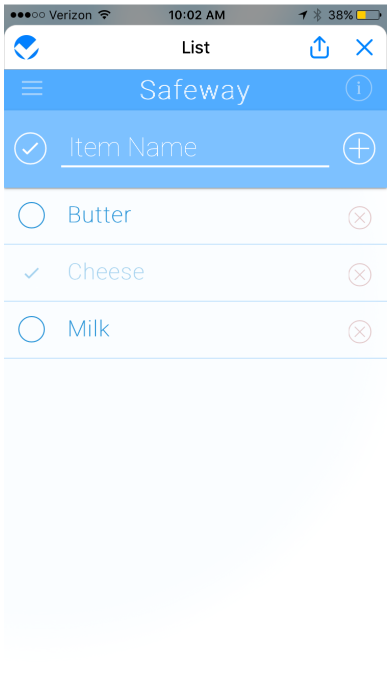


Trader Joe's
Costco
Safeway
New List Name
Gardening

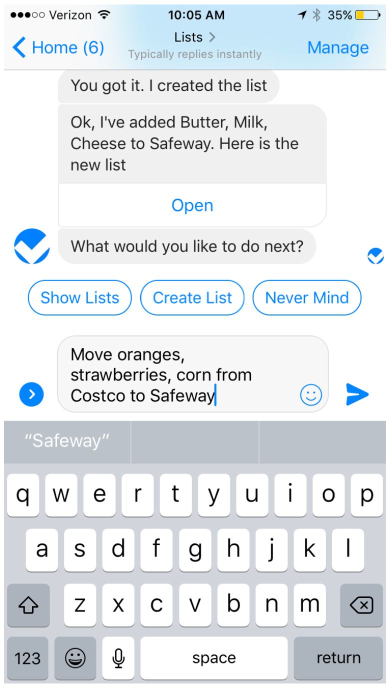
3. Form field with button

As a rule of thumb, we’ve found visual interfaces are useful when the bot is performing *iterative, refining tasks* such as searching for flights or changing a hotel reservation. It's helpful for activities like these to leave the confines of the message bubbles and present a richer, more-interactive UI.

On the other hand, we've found that *complex, one-time tasks* are better supported conversationally. In practice this means that users prefer to manage iterative tasks (such as checking/unchecking items in a list) using visual tools (1). And they prefer the conversational approach for complex, one-time activities such as moving items from one list to another (2).



1. Visual Interface



You got it. I created the list

Ok, I've added Butter, Milk, Cheese to Safeway. Here is the new list

What would you like to do next?

Move oranges, strawberries, corn from Costco to Safeway

2. Conversational Interface

3. Converse With Context

Human conversation can appear seductively simple. But even short phrases can become unwieldy due to the compounding effect of context. NLP systems are sufficiently advanced to parse a single line of text, even handling colloquialisms and misspellings. But they quickly lose the conversational intent when wrapped in competing layers of historical context. The bot, just like humans, gains true meaning through context.

The following command is a perfectly valid instruction for our *Lists* bot if it has enough context.

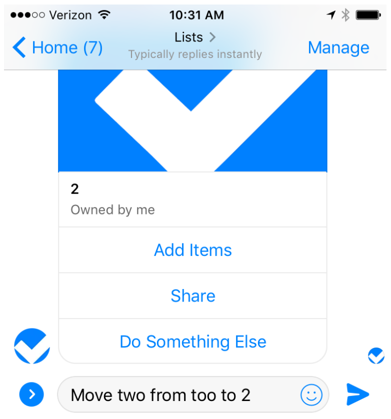
> *Move foo and moo from boo to loo*

But without context, our bot (and likely you) have no idea what's going on. Consider the same sentence, now preceded with some context.

> *Create a list named loo*
>
> *Create a list named boo*
>
> *Add items foo and moo to boo*
>
> *Move foo and moo from boo to loo*

In order to converse accurately over time, your bot may need to evolve its context individually, by user, and learn new meanings for words. In the following example, I've created a pair of lists named "too" and "2" (these are terrible names, but bear with me). Our bot starts life thinking that "2" is an adjective and "too" is an adverb. But because it can learn new meaning, it eventually understands that these are list names. Other users are unaffected by my choices. The words "2" and "too" retain their original meaning for those users as the context is personalized. But when I interact with the bot, what seems like apparent nonsense (I'm trying to move an item with the unfortunate name, "two") now makes sense.

Because the bot and I share an evolving context, when I say "*Move two from too to 2*" the bot knows exactly what I mean.



2
Owned by me

Move two from too to 2

4. Respect The Craft

Depending upon the messaging platform, many of the early solutions were created by casual users leveraging do-it-yourself toolkits. Unfortunately, there are no shortcuts in software. Easy-to-use software isn't easy to build. Suppose for a moment when Apple opened the App Store that most early apps were created by casual users leveraging do-it-yourself toolkits. It's safe to say that the iPhone image and brand would have suffered, with naysayers chiming, "The iPhone just isn't ready."

In time, the market will be established, and we will understand the solutions that work best when run atop a messaging app. Scale and quality will improve. Anyone familiar with the App Store's evolution will remember that it also had growing pains, with its share of juvenile apps. There will always be room for entertainment bots that insult you and tell you the time, but until second-generation solutions appear that move beyond novelty, messaging apps and the solutions that run atop them will continue to be dinged by naysayers.

5. Work The Network

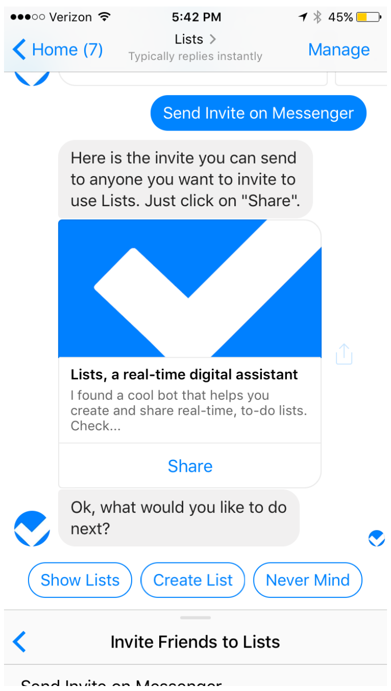
In a saturated marketplace where both apps and bots are struggling to make their mark, a distribution model that focuses on network virality can be advantageous. This was a significant driver for deploying our Lists bot on Facebook Messenger, especially given the robustness of their sharing APIs. What better way to extend the reach of the bot than to leverage the Facebook graph?

Sharing like selling must be done thoughtfully. I once had the good fortune to work alongside a very talented sales executive who told me his quick rule of thumb for closing a deal:

1. Why do they want it (relevance)
2. Why do they want it now (readiness)
3. Why do they want it from us (preference)

It's a great guide for optimizing a sales funnel, because it views the process from the customer's perspective. You're not there to cajole and convince, but rather to listen. If there's a fit, the product sells itself.

Not surprisingly, this approach works equally well when designing a personalized sharing strategy, as the approach helps to acknowledge the recipient's vantage point and the reasons they'll accept the invitation. We felt it was important to help users share our *Lists* bot generally (e.g., "Hey you should check this out!"), so we included a preformatted invitation to simplify the process.

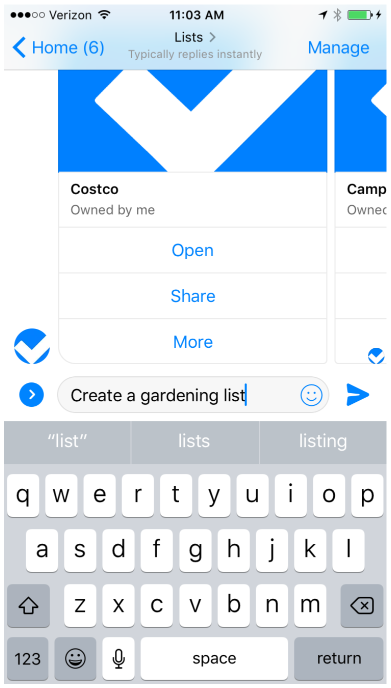


Here is the invite you can send to anyone you want to invite to use Lists. Just click on "Share".

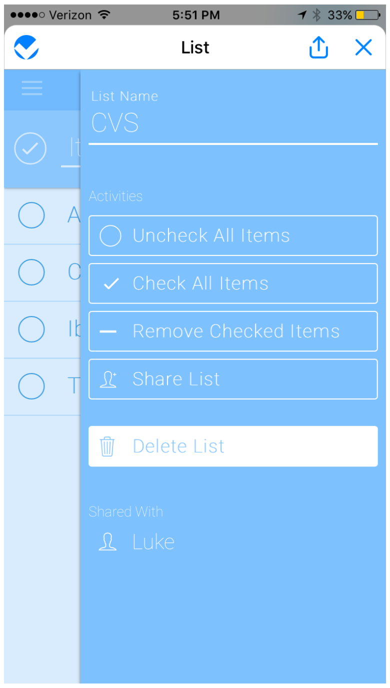
Lists, a real-time digital assistant
I found a cool bot that helps you create and share real-time, to-do lists. Check...

Ok, what would you like to do next?

We also added support for users to share a specific list while they worked on it. This is obviously much more effective as there is a timeliness to the invitation and a specific relevance to the recipient. We surface this capability throughout our bot given the likelihood that the invitation will be accepted. Here are just a few surfaces where we prompt users to share a list, including: in the Messenger message stream (1), and within the visual interface (2).



1. Including the Share link as a carrousel button in the message stream



2. Including the Share link as a button in the visual UI

Conclusion

Even with Messenger's large install base, a substantial number of people are still tied to SMS, email and other traditional media. (Or maybe they just prefer Slack.) The optimal long-term solution should bridge these various media, allowing users to choose their preferred platform and thereby reduce impediments to virality.

As an early experiment we included the ability to share portions of our *Lists* bot over SMS and Email. The functionality is limited for these deployments, but the approach provides a useful window into the ways users share their lists. Only time will tell, and we're excited to discover what works along with everyone else.

❤ 0 Likes ❤ Share

Posted in concepts *and tagged with* messenger, bot, web view, sharing, lists

Newer / Older

TERMS OF SERVICE PRIVACY POLICY VIDEOS

email info@intwixt.com
phone 415.570.8502