Microsoft Bot Framework Best Practices

Darren Jefford
Artificial Intelligence Lead Architect
Microsoft Services Technology Office
darrenj@microsoft.com
We learn and use technology.

Technology learns us and discovers ways to be helpful.

- Coordinating
- Answering
- Listening
- Observing
- Anticipating
1980s: **PC**
- Desktop

1990s: **Internet**
- Search
- User “visits” websites

2000s: **Mobile**
- Social
- User download apps from App Stores

The future: **Conversations**
- Natural language between people and technology
- Conversational canvas
- Bots and agents
Conversations as a platform

- Developers: The new canvas
- Businesses: The new opportunity
- Individuals: More personal way to discover, access, and interact
Conversations as a platform
Over 1 billion created account with 700 million active users

806 million MAU of Weixin & WeChat, increase of 34% YoY

Users in China spend 70 minutes per day on WeChat

Average revenue per UU more than $7 per year
Transforming engagement

- **Productivity task completion**
  Enhance productivity through task automation and through automated workflows

- Reach customers anywhere, on any platform or device

- Engage with customers and users in a natural way, conversationally and in context

- Expose your products and services via messaging platforms, where users are already highly engaged

- Improve customer experience and reduce the need for human assistance
Microsoft AI Portfolio

Agent
- Cortana

Applications
- Office 365
- Dynamics 365
- SwiftKey
- Pix
- Customer Service and Support
- Skype
- Calendar.help

Services
- Cortana Intelligence
- Cognitive Services
- Bot Framework
- Cortana Devices SDK
- Cognitive Toolkit

Infrastructure
- Azure Machine Learning
- Azure N Series
- FPGA
Build applications that understand people

- Faces, images, emotion recognition and video intelligence
- Spoken language processing, speaker recognition, custom speech recognition
- Natural language processing, sentiment and topics analysis, spelling errors
- Complex tasks processing, knowledge exploration, intelligent recommendations
- Bing engine capabilities for Web, Autosuggest, Image, Video and News
Cognitive Services

23 purpose-built ML and AI services to add predictable, configurable intelligence into any software

- Parse natural language (LUIS)
- Real-time transcription and translation services for voice
- Vision and face detection algorithms
- Sentence diagramming
- Web language (hashtag) parsing
- Many more

Developer Portal

Connect your bots to text/sms, Skype, Slack, Facebook Messenger, Office 365 mail and other channels.

- Register, connect, publish and manage your bot through your bot’s dashboard
- Automatic card normalization across channels
- Skype channel auto-configured
- Embeddable Web chat control
- Host your bot in your app via the Direct Line API
- Fast, scalable message routing
- Bot Directory for discovery and trial

Bot Builder

Tools and services to build great bots that converse wherever your users are.

- Open source SDK on Github for Node.js, .NET and REST
- From simple built-in prompts and command dialogs to simple to use yet sophisticated ‘FormFlow’ dialogs
- Support for rich attachments (image, card, video, doc, etc.); support for calling (Skype)
- Online/offline chat Emulator
- Add bot smarts with Cognitive Services for language understanding and more
Bot Builder SDK

Node.js, .NET and REST

- Dialogs to model conversation
  - Dialogs are reusable
  - Types of Dialogs include:
    - Built-in prompts
    - Yes/No, String, Number, Choices
    - FormFlow and form slot filling (branching, disambiguation, multi-turn)
  - Conversations are scalable to multiple machines

- Rich interactions
  - Support for rich attachments (image, card, video, doc, etc.); support for calling (Skype)
  - Service extensions for language understanding (LUIS) and translation

- Online/offline Chat Emulator

- Samples

Getting Started

What is Bot Builder for Node.js and why should I use it?
Bot Builder for Node.js is a powerful framework for constructing bots that can handle both freeform interactions and more guided ones where the possibilities are explicitly shown to the user. It is easy to use and includes features like Express & Restify to provide developers with a familiar way to write bots.

High-Level Features:
- Powerful dialog system with dialogs that are isolated and composable.
- Built-in prompts for simple things like Yes/No, strings, numbers, enumerations.
- Built-in dialogs that utilize powerful AI frameworks like LUIS.
- Bots are stateless which helps them scale.
- Bots can run on all major bot platforms like the Microsoft Bot Framework, Skype, and Slack.

Build a bot
Create a folder for your bot, cd into it, and run npm init.

```bash
npm init
```

Get the Botbuilder and Restify rest cannot be used with open.

```bash
gson install --save botbuilder
gson install --save restify
```

Make a file named app.js and say hello is a few lines of code.

```javascript
// Create bot and add dialogs
var bot = new builder.BotConversationBot();
app = new express.Application();
app.use('/bot', function(req, res, next) { res.send(404, 'Not found!'); });
app.post('/sendmessage', function(req, res) {
  res.send('Post received');
});
```

Bot.Builder();
make sure it's not running on the same port as express or restify.

```javascript
// Setup Restify Server
var server = new restify.Server();
server.listen(9999, function() { console.log('Bot Framework listening on port 9999'); });
```
• Register your bot
• Connect to channels
• Test
• Publish
• Manage
• Measure
Bot Directory

Public Directory of Bot Framework Bots

• Users can discover, try, and add bots to the conversation experiences on which the bot is configured (no app required)

• Bots are public at developer discretion; bots must be submitted for review in order to appear in the directory

• Searchable
Language Understanding Intelligent Service (LUIS)

Examples of intent detection
- Who is John?
- Where does Sue work?
- What is Erika's phone number?

Examples of entity extraction
Building a Bot
Hi, I’m [Bot name]

I love to...add the service description of the bot here

<table>
<thead>
<tr>
<th>PERSONALITY TRAITS</th>
<th>GOALS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The dev is the easy bit...

<table>
<thead>
<tr>
<th>MY VOICE &amp; TONE CHANGES WHEN I'M...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting started</td>
</tr>
<tr>
<td>Being proactive</td>
</tr>
<tr>
<td>Troubleshooting</td>
</tr>
<tr>
<td>Sensing a problem</td>
</tr>
<tr>
<td>Wrapping up</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MY RELATIONSHIP WITH MY USERS</th>
<th>THINGS I MUST DO</th>
<th>THINGS I NEVER DO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Introduce yourself
Knowledge and Tasks
UX / Language sophistication
Translation
General Knowledge
Personality
Search Powered
Exception Handling
Attachments
private async Task<Activity> HandleSystemMessage(Activity message)
{
    ...

    else if (message.Type == ActivityTypes.ConversationUpdate)
    {
        // Your introduction here
    }
}
UX and language sophistication
I think it's a close up of a bird
Next steps

Experiment! Start with understanding the end users needs...

Sign up for a free Azure Trial
https://azure.microsoft.com/en-us/free/

Microsoft Bot Framework
https://dev.botframework.com/

Cognitive Services