Build cognitive solutions for industries, Part 4: Enhancing the simplicity and quality of human-computer interactions

Mathews Thomas
Janki Vora
Utpal Mangla
Sharath Prasad
Matthew Rocco
Amandeep Singh

Cognitive computing is becoming increasingly important within the enterprise. In this fourth tutorial in a series, we discuss the primary methodologies and patterns used to build cognitive solutions for the telecommunications and media and entertainment industries.

Watson Conversation is now Watson Assistant. Although some illustrations in this tutorial may show the service as Watson Conversation, the steps and processes will still work.

Cognitive computing is more than just artificial intelligence. In the third article in our series—"Design patterns for making cognitive data searchable and understandable"—we showed how cognitive computing can be used to process data sets larger than previously imaginable, perform more complex analysis than humans alone are capable of, and provide unprecedented frontiers for data discovery and acquisition.

An often overlooked yet defining quality of cognitive computing is the "cognitive user experience." By using Watson processes and tools, we can reduce the need for specialized user training to interact with advanced machines by allowing for communication using natural language. In doing so, we can enhance the overall user experience.

That brings us to the current article. Here we discuss the primary methodologies and patterns used to build cognitive solutions for the telecommunications and media and entertainment industries. Previously, we collected, analyzed, and interpreted our data to make it searchable. Now, building on that work, we introduce the following:

- IBM Design Thinking, a framework to better understand the needs of stakeholders and customers.
• Patterns for enhancing applications’ conversational experiences using cognitive technology. We describe methods of enabling conversations between the user and the cognitive bot.
• Improving the overall user experience by using patterns to enhance the personalization of applications.

IBM Design Thinking

Design Thinking provides a method of problem solving focused on developing practical and creative solutions by directly addressing users' requirements. This four-step process challenges developers to understand end users and their needs, explore creative solutions, prototype the new design, and evaluate the results to determine what additional steps should be taken to improve the solution. By quickly and repeatedly progressing through these four phases, developers can fulfill the three guiding principles of agile development: make iterative enhancements, mitigate risk, and increase transparency throughout the process.

IBM’s enhancements to Design Thinking aim to extend this method to large and rapidly growing organizations by incorporating three tools into the standard Design Thinking process. These tools, also known as the "Three Keys," are:

• The first key, known as "The Hills," is a structure for developing specific user outcomes that characterize the uniqueness of the user ("The Who"), the impact the feature will have on the user ("The What"), and the differentiator the approach will provide ("The Wow"). For example, the Watson for Network Operations tool focuses on supporting the needs of the entry-level network engineer ("The Who") in a way that enables skill development and increases productivity. A ticket-difficulty ranking system matches entry-level engineers to appropriate challenges and provides easily searchable support documents ("The What"). The progressive, consistent improvement in both confidence and knowledge these features provide has distinct benefits over traditional training and ticketing systems ("The Wow"). By using this tool, you are free to innovate, knowing that there are clearly defined outcomes to work toward.

• The second key, known as "Playbacks," is a series of selectively scheduled events that bring the development team into contact with key stakeholders. By seeking input from critical stakeholders, and concentrating on storytelling rather than discussing specifications, our team has been able to spend more time developing the most important aspects of a solution and has avoided time spent on unnecessary or unsatisfactory components.

• The third key is the use of "Sponsor Users," individuals who represent the target audience and have the time and motivation to provide feedback to the development team, ideally throughout the product lifecycle. Our team has consistently gathered input from both novice and expert-level network engineers, which has helped maximize the impact of key features and avoid redundant work.

By including these three tools in our development process, our team has been able to identify opportunities to apply cognitive user experience patterns and ensure that they meet changing user needs.
Enhancing applications' conversations through cognitive capabilities

In previous articles, we discussed use cases involving Watson for Network Operations and Device Doctor. In each example, we used part or all of the pattern flow illustrated in Figure 2.

Through Design Thinking, we learned that Level 1 technicians often need assistance from senior technicians, resulting in the escalation of a ticket. The system first ingests data from various channels and formats including social media, mobile applications, web, and voice-enabled systems. All user input is converted to text. Intents, entities, sentiment, and tone are extracted. An orchestrator then saves each query and its context. Depending on the nature of the query (long- or short-tail, a decision is made to invoke appropriate work streams. Short-tail questions are defined as pre-trained questions and responses. Long-tail questions are unexpected queries, but a top set of potential responses is available using ranking algorithms (in a pattern similar to making data searchable as discussed in Part 3). If frustration is detected within the intent or during the conversation, the conversation channel is then handed off to a human agent.
The diagram in Figure 2 shows how we designed Watson applications to use conversations between a user and agent. In the discussion below, we describe our process in detail.

**Speech to Text**
We use the Watson Speech to Text service to enhance our interaction with the cognitive bot. Speech to Text can transcribe audio voice into written text. It uses machine intelligence that combines information about grammar and language structure with knowledge of composition of the audio signal to generate accurate transcriptions.

We also use Speech to Text in our cognitive bot to enhance the user's interaction, enabling the user to speak directly to the bot rather than typing the interaction. Combined with Watson Assistant, Watson Natural Language Understanding and other Watson services can simulate a conversation in natural language.

We use the HTTP Rest interface and the WebSocket interface to connect to the Speech to Text service in two different use cases. In Cognitive Field Service Advisor, we use the HTTP Rest Interface. This method lets you send audio through the body of the request or as multipart form data that consists of one or more audio files. We use a session-less connection that includes HTTP calls, which enables us to provide a simple means of transcribing audio without the overhead associated with establishing and maintaining a traditional session.

```
curl -X POST -u '{username}':'{password}'
   --header "Content-Type: audio/flac"
   --data-binary @audio-file1.flac
   --data-binary @audio-file2.flac
  "https://stream.watsonplatform.net/speech-to-text/api/v1/recognize?
timestamps=true&word_alternatives_threshold=0.9&continuous=true"
```

In Device Doctor, we use the WebSocket interface. Requests and responses are enabled over a single TCP connection that abstracts most of the complexity of the request to offer efficient implementation, low latency, high throughput, and an asynchronous response.

```
var token = '{authentication-token}';
var wsURI = 'wss://stream.watsonplatform.net/speech-to-text/api/v1/recognize?watson-token=' + token;

var websocket = new WebSocket(wsURI);
websocket.onopen = function(evt) { onOpen(evt) };
websocket.onclose = function(evt) { onClose(evt) };
websocket.onmessage = function(evt) { onMessage(evt) };
websocket.onerror = function(evt) { onError(evt) };

function onOpen(evt) {
    var message = {
        action: 'start',
        'content-type': 'audio/wav',
        continuous: true,
        interim_results: true
    };
    websocket.send(JSON.stringify(message));
    // Prepare and send the audio file.
    websocket.send(blob);
    websocket.send(JSON.stringify({action: 'stop'}));
}

function onMessage(evt) {
    console.log(evt.data);
}
```
Watson Assistant

Watson Assistant combines a number of cognitive techniques to help you build and train a bot. The system can be further refined with supplementary technologies to make the system more human-like or to give it a higher chance of returning the right answer. Watson Assistant lets you deploy a range of bots through many channels, from simple, narrowly focused bots to much more sophisticated, full-blown virtual agents across mobile devices, messaging platforms such as Slack, or even through a physical robot.

Creating Conversation

Natural-language processing happens inside a workspace, which is a container for all of the artifacts that define the conversation flow for a particular application. You can create multiple workspaces inside a Watson Assistant service. A workspace has three elements:

- **Intents** identify the purpose of a user's input, such as a question about a specific user issue or a bill payment. We define intents for each type of user request that you want your application to support. To train the workspace to recognize your intents, you supply lots of examples of user input and indicate which intents they map to. The name of an intent is always prefixed with the `#` character.

  ![](image)

  **Figure 3. Example of Watson Assistant Service Intent user interface**

- **Entities** are objects relevant to intents and which provide a specific context. For example, an entity might represent a source that the user wants to use for a backup process. To train the workspace to recognize your entities, you list the possible values for each entity and synonyms that users might enter. The name of an entity is always prefixed with the `@` character.
• **Dialog** defines the main conversation flow. It uses a branch pattern that defines how your application responds when it recognizes the defined intents and entities. We use the dialog builder to create conversations with users, providing responses based on the intents and entities that we recognize in their input.

**Figure 4. Example of Watson Assistant Service Entities user interface**

**Figure 5. Example of Watson Assistant Service Dialog user interface**

**Tone Analyzer**

Another service that we use to enhance our cognitive bot is Watson Tone Analyzer. With Tone Analyzer we are able to detect the different tones that can impact the effectiveness of communication. Tone Analyzer can detect tones from the document level but can also go deeper to the sentence level, detecting from text three types of tones:

- Emotion (anger, disgust, fear, joy, and sadness)
- Social propensities (openness, conscientiousness, extroversion, agreeableness, and emotional range)
- Language styles (analytical, confident, and tentative)

We can use Tone Analyzer to classify the tone of a customer, identifying, for example, whether it reflects frustration, sadness, satisfaction, excitement, and so on. With this analysis, we can properly determine whether the customer needs to talk to a live agent. For example, if a user...
becomes frustrated, we can direct him, along with the context of the conversation, to a live agent. We can then follow up with the user to confirm that this transfer occurred.

**Figure 6. Example of Watson Tone Analyzer output**

```
{  "utterance_id":2,  "utterance_text":"Nothing is working",  "tones": [   {      "score":0.801019,      "tone_id":"sad",      "tone_name":"sad"   },   {      "score":0.66593,      "tone_id":"frustrated",      "tone_name":"frustrated"   }  ]}
```

**Transferring to another chat system**

With our bot we also have the capability to transfer the chat and its content over to another chat. For example, if the client primarily leverages the Slack chat platform, we could have our bot transfer the chat logs to an agent. While this sounds simple enough, it is also capable of transferring the context and the keywords of the chat as shown in the example above.

**Watson Virtual Agent**

One can develop everything using the core cognitive components that are described above, but a tool like Watson Virtual Agent, which comes pre-configured with conversations, intents, and analytics, can help jump-start the process.

Figure 7 gives a complete view of how Watson Virtual Agent interacts with different actors, including the customer, life agents, subject matter experts (SMEs), and developers:
The customer can interface directly with Watson Virtual Agent through different channels. A complete functional widget is available on Watson Virtual Agent GitHub. This widget can be easily integrated on any web interface just by configuring several parameters. Other channels such as Facebook, Slack, and SMS require customization. However, there is a client SDK available to allow multiple customizations.

SMEs identify trends and understand how customers should be engaged. Watson Virtual Agent allows the SME to configure the chat bots without the need of deep technical skills by simply enabling or disabling intents. In some cases, this will require linking customized conversations from the Watson Assistant service, but that is a straight-forward process. In addition, the developer will decide which of the active intents should be handled through the conversational bot and which active intents should be handled through escalation to a live human agent. All of this is done through a friendly, graphic-based user interface.
The manager needs to monitor how interactions have been made with the customers. Therefore, Watson Virtual Agent comes with a set of engagement metrics reports. These reports enable the manager to see in real time the most common intents or topics by breaking them down based on time and geographical regions. It also allows exporting interactions as raw data to be handled within other analytical tools.

Live human agents can also be reached directly from within Watson Virtual Agent. Depending on the implementation, Watson Virtual Agent can be configured to automatically escalate to a human agent whenever a specific intent is detected from the customer.

Currently, tools exist for developers that streamline the development process for custom-made bots, greatly accelerating deployment times. Furthermore, these tools allow the developer not only to create conversational robots but also to connect to external services and tools to pull in and interpret the data.

**Watson Virtual Agent and Watson Assistant**

Watson Virtual Agent is IBM's SaaS offering that enables conversational self-service agents to provide responses and take actions. This product packages the power of Watson Assistant bundled with tooling for SMEs, developers, and managers. Out of the box, Watson Virtual Agent brings about one hundred pre-trained intents. In addition, it includes some predefined content that triggers actions such as updating an email address or paying a bill with a credit card. It also enables developers to train their own intents or create their own dialogs by integrating with Watson Assistant.

Although Watson Virtual Agent and Watson Assistant share many features and functions, they do have some significant differences. For example, Watson Assistant is a service available in IBM Cloud with web tooling to train and create dialog. When instantiated, there are no predefined content or integration settings. Watson Virtual Agent, on the other hand, is powered by Watson Assistant but comes bundled in a SaaS offering with lots of predefined intents and content, such as a dashboard for configuration and integration, a fully functional widget to integrate with any website, and a complete SDK for creating custom-made chat widgets.

**Patterns for enhancing applications' personalization**

We applied two major patterns for personalization using cognitive APIs: the Cognitive Data Enrichment process and the Cognitive Profiling process. These patterns can be leveraged together to enhance content and application personalization. Figure 8 illustrates the flow. First, we enrich cognitive attributes for video content by using various Watson APIs. Next, we enrich cognitive attributes of customers leveraging various Watson APIs. We then apply a variety of advanced analytics techniques to personalize the recommendation. In addition, we use the pattern for cognitive conversation to enhance our personalized TV applications user experience as discussed above.
To uncover the maximum number of insights from video and other related media, a pattern we had followed for several of our solutions involved splitting video content into an audio stream and an image stream. Next, we analyzed each stream individually using a combination of cognitive components. Time-coded metadata had been generated for both images and audio associated with each video in our collection. This gave us some new ways to further personalize our applications for each user. Figure 8 represents one possible pattern for creating such a system through the use of cognitive components.

With video metadata generated and ingested into a search service such as Discovery, there are several potential applications and use cases that could leverage the insights provided by cognitive video analysis. In Personalized Television, we could match user interests gathered from social media to entities identified using Natural Language Understanding to recommend personalized video playlists to users. In Watson for Network operations, we could leverage annotated video and audio to make relevant tutorials and product guides available to NOC engineers while they troubleshoot tickets. In the sections that follow, we describe how Watson Visual Recognition and Natural Language Understanding are used.

Watson Visual Recognition

Visual Recognition is the key component used to find insights on frames extracted from the videos. It is a set of REST APIs that let you classify objects, identify faces, and read text on the images. There are many pre-trained classifiers (constantly increasing in number). This feature can be trained by creating custom classifiers where positive and negatives examples should be provided.
Faces will identify some features of the people on the pictures such as gender and approximate age. Visual Recognition is a service available on IBM Cloud. It has a RESTful architecture and is supported on the SDKs from the Watson Developer Cloud for many different languages.

**Figure 9. Example of Watson Visual Recognition output**

Watson sees...

![Image of Watson Visual Recognition output]

**Natural Language Understanding**

Natural Language Understanding is another primary component used in such solutions as Personalized TV, Watson for Network Operations, and Device Doctor. In addition to identifying sentiment as discussed above, Natural Language Understanding enables you to analyze and extract several types of metadata from content including concepts, entities, keywords, categories, emotion, relations, and semantic roles. It represents the next-generation service for processing text. NLU can even be trained to identify entities, concepts, and keywords in a particular knowledge domain or industry.

In Figure 8, the Speech to Text-converted content is passed through Natural Language Understanding with a time code. The video content is then enriched with time code information and the following attributes:
Patterns for cognitive profiling
Humans have the ability to understand personality characteristics such as agreeableness, aggressiveness, and openness, as well as sentiments like joy and anger. We are leveraging Watson APIs in our cognitive profiling pattern to understand these and other human characteristics. As shown in Figure 8, we use Natural Language Understanding and Personality Insights to enrich individual profiles.

Natural Language Understanding
In our Personalized Television solution for Media and Entertainment, Natural Language Understanding was used to analyze audience tweets so that a major broadcasting company could generate more relevant, personalized recommendations for shows a user might be most interested in watching. Additionally, the same solution could assist marketing teams in identifying which advertisements should be targeted to which audiences. In Figure 10, we see that Malcolm Middleton has mentioned movies, including *Star Wars*, suggesting that he might enjoy other works of science fiction. We also see that Malcolm has particular brand interests including "Apple," so perhaps the broadcasting company should show Malcolm (and other *Star Wars* fans like Malcolm) more Apple advertisements.

Figure 10. Cognitive profiling for Media & Entertainment dashboard
As shown in Figure 8, cognitive profiling also leverages natural language understanding. *Sentiment* is very useful in many situations. With sentiment we can determine whether a user has a positive vibe, a negative vibe, or a neutral vibe toward a particular topic. While the range seems very small, this is enough information to determine how a person feels about a topic. With our Natural Language Understanding service, we can pull out the topic and sentiment. This is very powerful in that we can not only figure out what the user is talking about but his or her attitude toward the topic as well.

**Personality Insights**

Personality Insights is a Watson service that can be used to trace the personality of a written extract. The results will show how the text relates to the *"Big 5 Personality Traits"*, as well as values and needs, along with a percentage of how closely that text fits those categories. An excerpt of at least 1200 words is recommended for accurate results. By extracting a user's personality, we can begin to dive deeper into the actions that he might take. For example, we can determine how likely a user is to take an offer based on how open and agreeable he is. We can also determine the best time to display an offer to the user based on his or her emotional range.

**Figure 11. Example of Watson Personality output**

By understanding a user's personality, you can determine what he is like and how likely he is to act. Figure 10 shows an example of our Personalized TV application. As you can see, we have chosen the user Malcolm. From pulling in his tweets and other social media data (see Part 3) and then passing it through Natural Language Understanding, Personality Insights, and other Watson Services, we can learn his interests and something about his personality. *Personality* is an important facet of this process, as it enables us to find a pattern between it and the types of
shows he likes. Then, we can create a generalization of what other shows a user with a similar personality is likely to watch. As shown, we can see that Malcolm is a big fan of *House of Cards*. On the right, in the recommendations section, we see that *Lord of the Rings* might be a movie he would be interested in.

**Conclusion**

Cognitive computing is redefining the relationship between human and machine. By utilizing IBM Design Thinking, the development community can play an integral role in creating innovative, efficient, and meaningful interactions. We can expand our interactions with machines to include natural language, human speech, and visual recognition. With conversation services, virtual agent, and tone analyzer, we can ask contextual questions interpret tone, and include other human characteristics allowing for even more adaptive interaction. This can be extended further through video analysis, personality insights, and sentiment analysis where the human-machine interaction is enriched even further.

These cognitive tools will enable us more than ever before to improve human-machine interaction without the need for extensive machine training, thus empowering professionals to devise previously unattainable solutions to the most pressing challenges within their industry.
Related topics

- Design Thinking
- Watson Assistant
- Natural Language Understanding
- Visual Recognition
- Personality Insight
- Tone Analyzer

© Copyright IBM Corporation 2018
Trademarks
(www.ibm.com/developerworks.ibm/trademarks/)