DevDash

Building a Conversable agent with Dialogflow CX

By DevDash Labs . January 21, 2025

✉ innovate@devdashlabs.com

1. Executive Summary

In today's fast-paced digital landscape, customers demand immediate and efficient support. Many companies struggle to meet these expectations, often hindered by long response times and limited agent availability, leading to frustrated customers and lost revenue. This white paper introduces Dialogflow CX as a solution for building conversational agents that provide 24/7 instant support, automate resolutions for common issues, and seamlessly escalate complex cases to human agents. By combining empathetic, conversational design with advanced AI capabilities, Dialogflow CX enables businesses to create scalable, adaptable, and context-aware virtual agents. This next-generation approach not only transforms customer service into a faster, more reliable, and satisfying experience but also enhances operational efficiency and reduces costs. This paper delves into the key features of Dialogflow CX, exploring its architecture, development process, integration capabilities, and real-world application to demonstrate its transformative potential in customer interaction.

2. Introduction

Imagine a customer facing an issue with a company's services and seeking immediate resolution, only to encounter challenges like long response times and limited agent availability. To address these problems, we introduce a chatbot built using Dialogflow CX. It provides 24/7 instant support, automates resolutions for common issues, and seamlessly escalates complex cases to human agents when needed. With its empathetic and conversational design, the chatbot ensures customers feel valued while streamlining the support process. By eliminating delays and enhancing efficiency, it transforms customer service into a faster, more reliable, and satisfying experience.

3. Background

The evolution of conversational AI has been remarkable, progressing from simple rule-based chatbots to sophisticated systems capable of understanding and generating natural language. Early chatbots were hampered by limitations in context retention and flexibility. However, advancements in machine learning and natural language processing (NLP) have enabled dynamic, human-like interactions. Early chatbots were primitive, with predefined steps. In this day and age, with the revolution in AI, Dialogflow CX represents a next-generation approach, offering robust features like multi-turn conversations, advanced NLU models, and stateful flows. This enables businesses to create scalable, adaptable, and context-aware virtual agents, marking a major leap in delivering seamless user experiences across multiple channels.

3.1 Problem Statement

Many companies face significant challenges in providing timely and effective customer support. Traditional support systems, heavily reliant on human agents, often struggle to meet the demand for 24/7 instant assistance, resulting in:

1. **Long Response Times:** Customers often experience frustrating delays, waiting for extended periods to receive help.
2. **Limited Agent Availability:** Human agents cannot be available at all times, leading to gaps in service coverage, especially outside of standard business hours
3. **Inconsistent Service Quality:** The quality of support can vary depending on the agent, leading to an inconsistent customer experience.
4. **Scalability Issues:** Scaling up support operations to meet increased demand can be expensive and complex with traditional human-centric models.
5. **Missed Opportunities:** Many companies face missed opportunities as they cannot scale their business because of limited resources.

These challenges highlights the need for an advanced and scalable solution to manage customer interactions efficiently and effectively. This is where Dialogflow CX presents itself as a transformative solution.

4. Why Dialogflow CX?

When choosing a platform for building a chatbot, Dialogflow CX stands out due to its advanced capabilities, such as state management, robust NLP capabilities, and a focus on scalability, making it ideal for customer support use cases. Dialogflow CX delivers better intent recognition and contextual understanding. It also surpasses contemporary chatbot platforms like Amazon Lex AI in terms of visual design and minimal coding. Moreover, Dialogflow CX excels in supporting complex conversational requirements compared to other platforms. A key advantage is its broad multichannel integrations, including **Messenger**, **Slack**, **Twilio**, **Telegram**, **Viber**, **Workplace**, **LINE**, **Discord**, and **Google Chat**. There are many features we can exploit in Dialogflow CX, which we will explore in this article.

5. Core Components: A Deep Dive

This section will delve into the core components that make up the architecture of Dialogflow CX.

5.1 State Architecture

The backbone of Dialogflow CX lies in its state machine architecture, defined by the following:

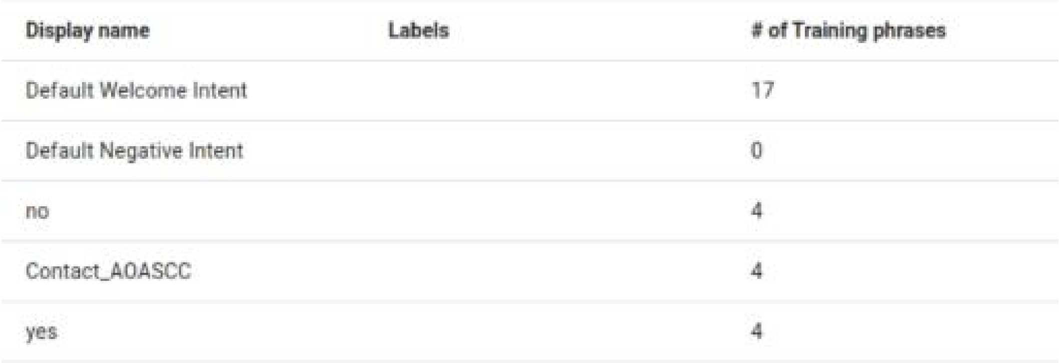
a. States and Transitions

In Dialogflow CX, states are defined by pages or flows.
A transition represents the path a conversation takes when moving between states (pages or flows) based on specific conditions. It dictates where the conversation moves next, whether it be another page within the same flow or a different flow altogether.

A transition occurs when:
* The user input matches an Intent.
* A specific event (like no-match or no-input) is triggered.
* The required parameters are successfully collected.

i. How do Intents Lead to Transition?

Intents are the basic building blocks of the Dialgoflow CX that makes it possible to interact with the chatbot. Each Intent is destined for some use case. Basically, Intent is what the user intends to do. Here are some of the intents:

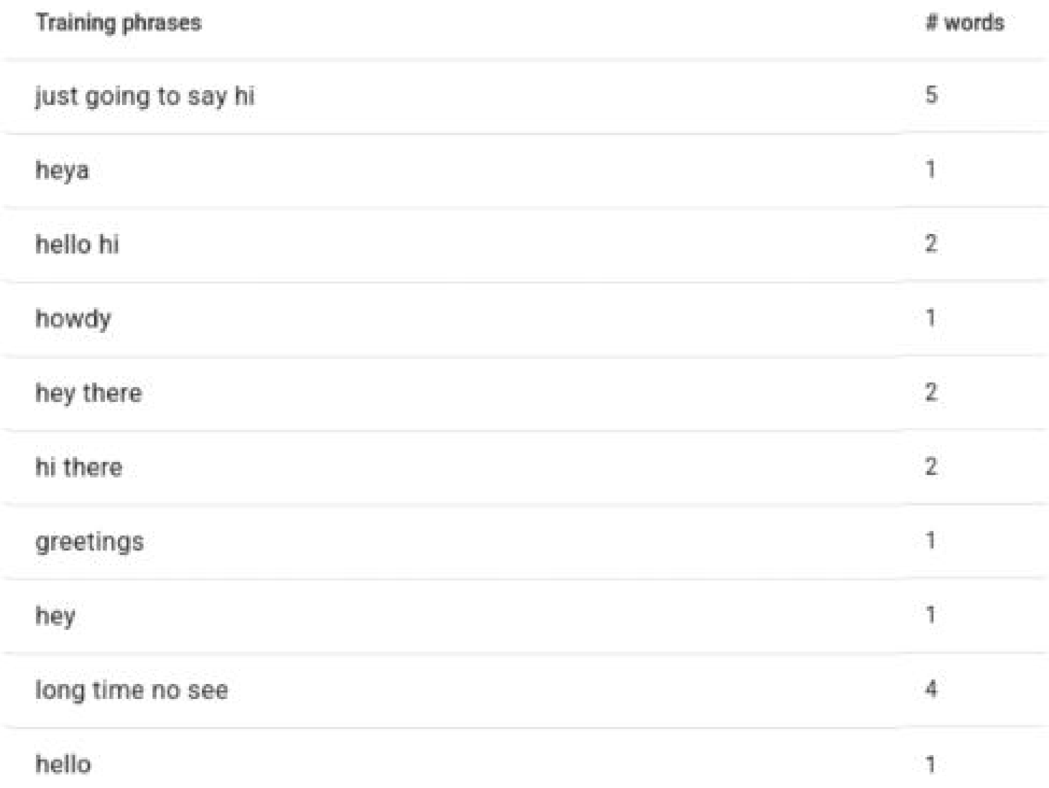


Display name	Labels	# of Training phrases
Default Welcome Intent		17
Default Negative Intent		0
no		4
Contact_AOASCC		4
yes		4

Fig i. Multiple intents

The above Default Welcome Intent gets triggered when a user says hi, or hello. We have to set the training phrases allowing Dialogflow CX to recognize such intentions of the user.

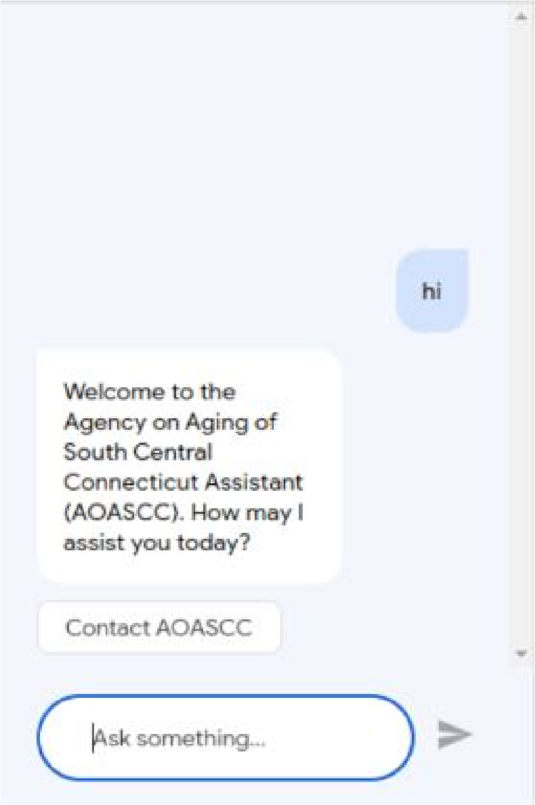
Here's the training phrase of "Default Welcome Intent"



Training phrases	# words
just going to say hi	5
heya	1
hello hi	2
howdy	1
hey there	2
hi there	2
greetings	1
hey	1
long time no see	4
hello	1

Fig ii. Training Phrases for Default Welcome Intent

We can provide any number of training phrases, the more the better. Now, when the user input matches that intent, we make the agent respond to it through a message (text or payload) and finally transition to the next state. For example, the agent says “Welcome to the Agency on Aging of South Central Conneticut Assistant. How may I assist you today?”



hi

Welcome to the Agency on Aging of South Central Connecticut Assistant (AOASCC). How may I assist you today?

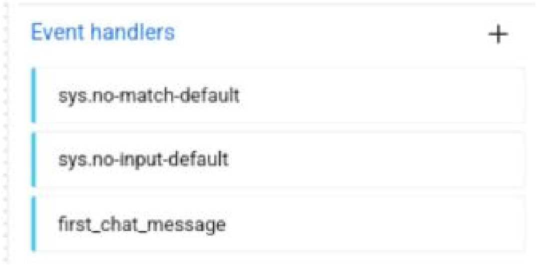
Fig iii. Agent's text response

5.2 State Handlers

State handlers in Dialogflow CX are mechanisms that manage and control the flow of conversation based on specific conditions or events. Each page has a state handler, consisting of the following:

a. Event Handlers

Event Handlers come into play when we want to handle some exceptions like no-match and no-input. Suppose, what if a user asks something and isn't mentioned in routes or data stores. In that scenario, event handlers become important. If we want to handle things that we haven't mentioned in the routes or in the data stores, we use event handlers.



Event handlers

sys.no-match-default

sys.no-input-default

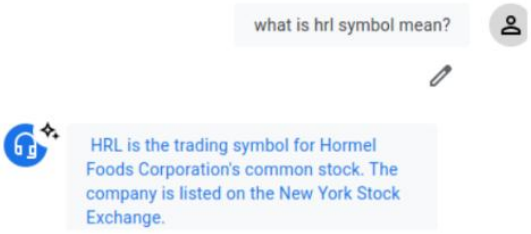
first_chat_message

Fig iv. Event handlers

In the real world scenario, we have no idea what the user might ask. We have to be ready to answer all their queries. For example: When a user asks “hey, how is the weather today?”, we have to respond with something meaningful. In this case, we’re responding that we don’t have any information about it. These exceptions, in dialogflow cx, are handled by the Event handlers.

b. Data Store

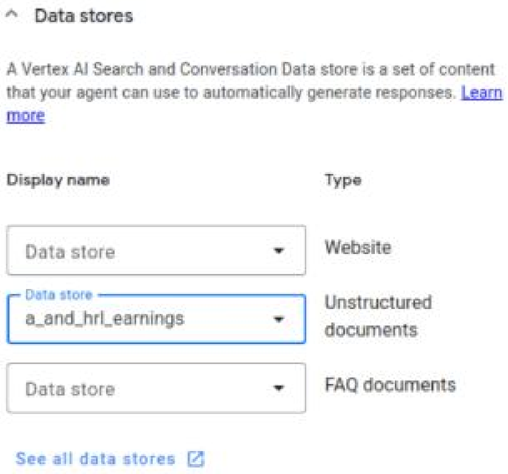
Data stores are very crucial to incorporate NLP capabilities into our Chatbot. When the User queries match the content in the data store, the data store agent makes sure to respond to the user. However, if the content doesn’t match, we will make use of routes to ensure proper conversation between user and the chatbot. This way, our Chatbot becomes less primitive and more dynamic as well as flexible. Users can interact with the data stores to retrieve any information present in the data store.



what is hrl symbol mean?

HRL is the trading symbol for Hormel Foods Corporation's common stock. The company is listed on the New York Stock Exchange.

Fig v. Data store agent in action



Data stores

A Vertex AI Search and Conversation Data store is a set of content that your agent can use to automatically generate responses. Learn more

Display name	Type
Data store	Website
Data store: a_and_hrl_earnings	Unstructured documents
Data store	FAQ documents

See all data stores

Fig vi. Assigning data store to the page

c. Routes

Routes are configured for each page to manage how the conversation moves forward. A page can have multiple routes to handle various intents and conditions.

6. Flow Designer

6.1 Flows

Flow is the highest level component in Dialogflow CX's conversation design. As the name suggests, it represents the direction the conversation is heading towards. We can create many flows. Each flow contains special functionalities. A Flow in Dialogflow CX consists of multiple Pages connected through routes.

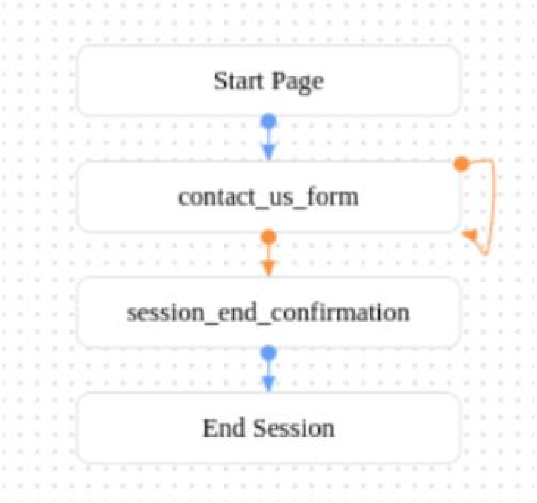


Fig vii. Contact us flow

6.2 Entry and Exit Points

a. Entry Point

Start Page: This is the initial point where the conversation begins. All user interactions start here. In Dialogflow CX, the Default Start Page of the Default Start Flow is where the conversation begins.

b. Exit Point

End Session: This is the final point where the session terminates, marking the end of the conversation.

c. Flow Analysis

After the Start Page, the flow moves to the contact_us_form.From contact_us_form, it loops back to itself if the conditions aren’t met before transitioning to session_end_confirmation. After reaching session_end_confirmation, the flow moves to End Session, which is the terminal state.

6.3 Flow to Flow Connection

In Dialogflow CX, a flow-to-flow connection allows the chatbot to transition from one flow to another. This is useful for organizing complex conversations by dividing them into smaller, manageable parts.

In Dialogflow CX, the start point is always the Default Start Page of the Default Start Flow as depicted in the above picture. Here, we are routing to the Contact_Us flow given the user input matches the intent or a condition. Now, within the contact_us flow, we have multiple pages as shown in Fig vii.

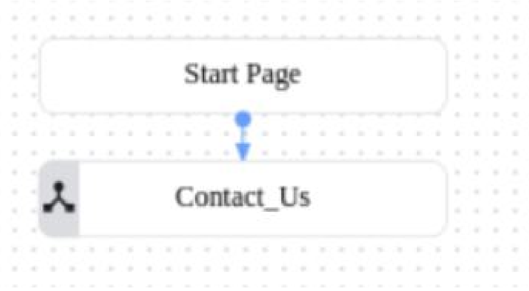


Fig viii. Transitioning from Default Start Flow to Contact_Us flow

7. Parameter System

7.1 System Parameters:

These are predefined parameters that Dialogflow automatically provides. They are typically associated with specific built-in entities (like dates, times, numbers, etc.) They are typically represented by a special character followed by the type such as **@sys.date**, **@sys.time**, **@sys.phone**, **@sys.email**, etc.

@sys.date

@sys.date-period

@sys.date-time

@sys.time

@sys.time-period

Fig ix. System Entities/Parameters

7.2 Custom Parameters

These are user-defined parameters created to store specific data extracted from user inputs such as order_status or company_name.

7.3 Session Variables and Page Variables/ Parameters:

Session Variables in Dialogflow CX are temporary, context-aware variables that exist during the lifespan of a conversation (session). These variables are used to store information and maintain context across the conversation flow. They are automatically managed by Dialogflow CX and are cleared when the session ends.

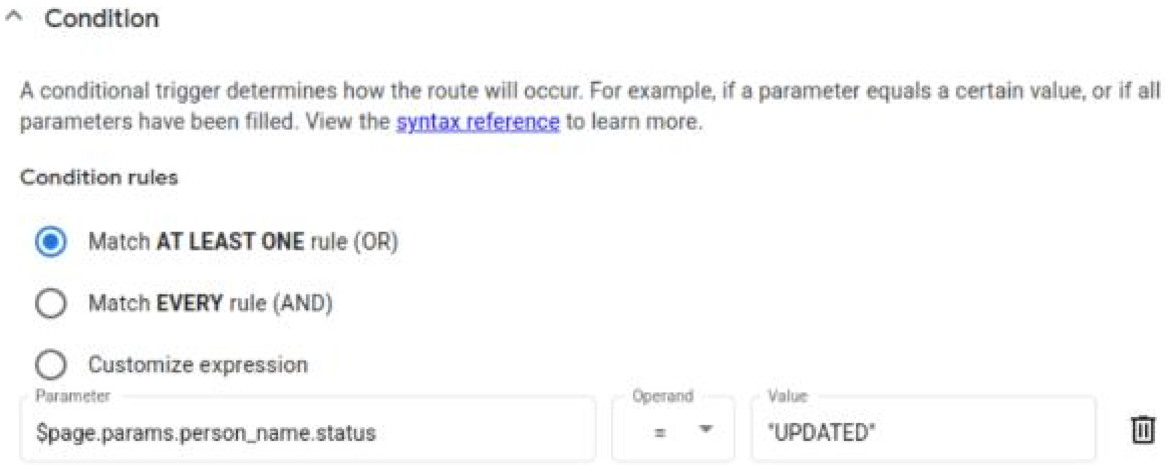
They are typically stored in **$session.params**. Dialogflow automatically assigns session variables when extracting parameters from user inputs.

Page parameters are temporary variables that exist only while the user is interacting with a specific page in the flow. They are cleared when the user transitions to a new page unless explicitly saved to session parameters.

They are typically stored in **$page.params**.

7.4 Condition

In Dialogflow CX, a condition is a set of rules or expressions used to determine the next action in a flow based on the context of the conversation. Conditions are typically used in routes to decide whether to transition to a specific page, flow, or webhook, or to take other actions.



Condition

A conditional trigger determines how the route will occur. For example, if a parameter equals a certain value, or if all parameters have been filled. View the syntax reference to learn more.

Condition rules

● Match **AT LEAST ONE** rule (OR)

○ Match **EVERY** rule (AND)

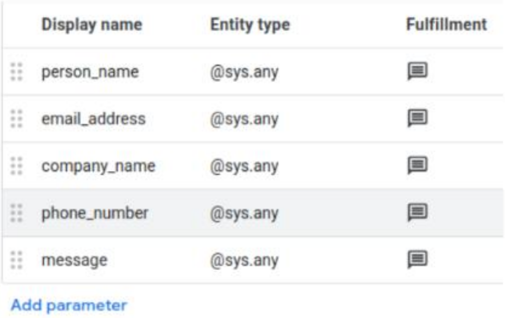
○ Customize expression

Parameter	Operand	Value
$page.params.person_name.status	=	"UPDATED"

Fig x. Conditional Trigger Configuration in Dialogflow CX Routes

In the fig, if **$page.params.person_name.status == “UPDATED”**, we move to other pages or flows. The **person_name** is a form parameter.

8. Creating a Form in Dialogflow CX



Display name	Entity type	Fulfillment
person_name	@sys.any	💬
email_address	@sys.any	💬
company_name	@sys.any	💬
phone_number	@sys.any	💬
message	@sys.any	💬

Add parameter

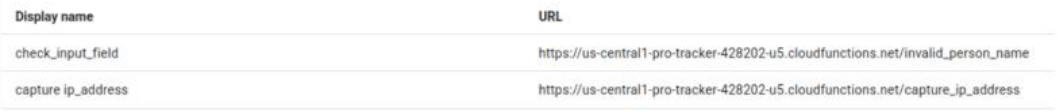
Fig xi. Form parameters

To create a form in Dialogflow CX, navigate to the desired page in a flow, go to the Parameters section, and add parameters. Define each parameter with a display name (e.g., person_name, email_address), an entity type (e.g., @sys.any), and optional prompts to collect user input. You can also enable fulfillment for custom logic or validation. Use system entities like @sys.email or @sys.phone-number for specific input types. Save and test the form in the Test Console to ensure it collects the required data correctly.

9. Interaction with Backend Systems

9.1 Web Hooks

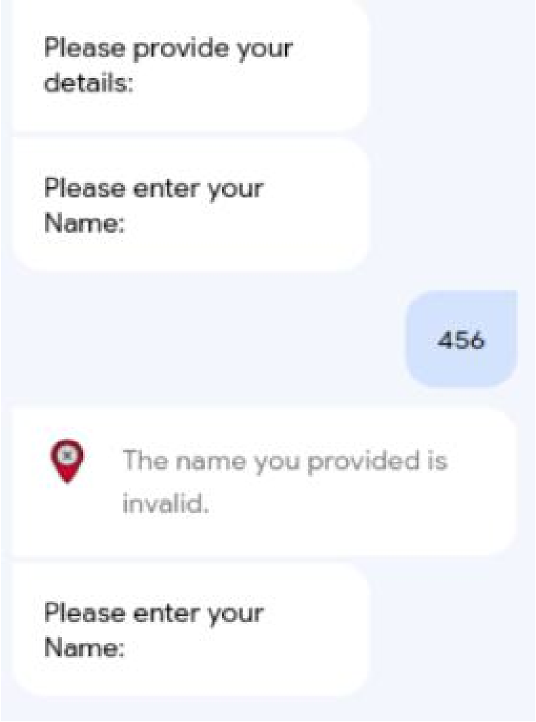
Webhooks are services that host your business logic or call other services. During a session, webhooks allow us to use the data extracted by Conversational Agents (Dialogflow CX)'s natural language processing to generate dynamic responses, validate collected data, or trigger actions on the backend.



Display name	URL
check_input_field	https://us-central1-pro-tracker-428202-u5.cloudfunctions.net/invalid_person_name
capture ip_address	https://us-central1-pro-tracker-428202-u5.cloudfunctions.net/capture_ip_address

Fig xii. Webhook endpoints

A good example of webhook is to validate the user input as shown in the below picture. First,ensure you have a webhook service set up. This service will handle the requests from Dialogflow CX. It can be a REST API hosted on a cloud platform (like Google Cloud Functions, AWS Lambda, or any server that supports HTTP requests).



Please provide your details:

Please enter your Name:

456

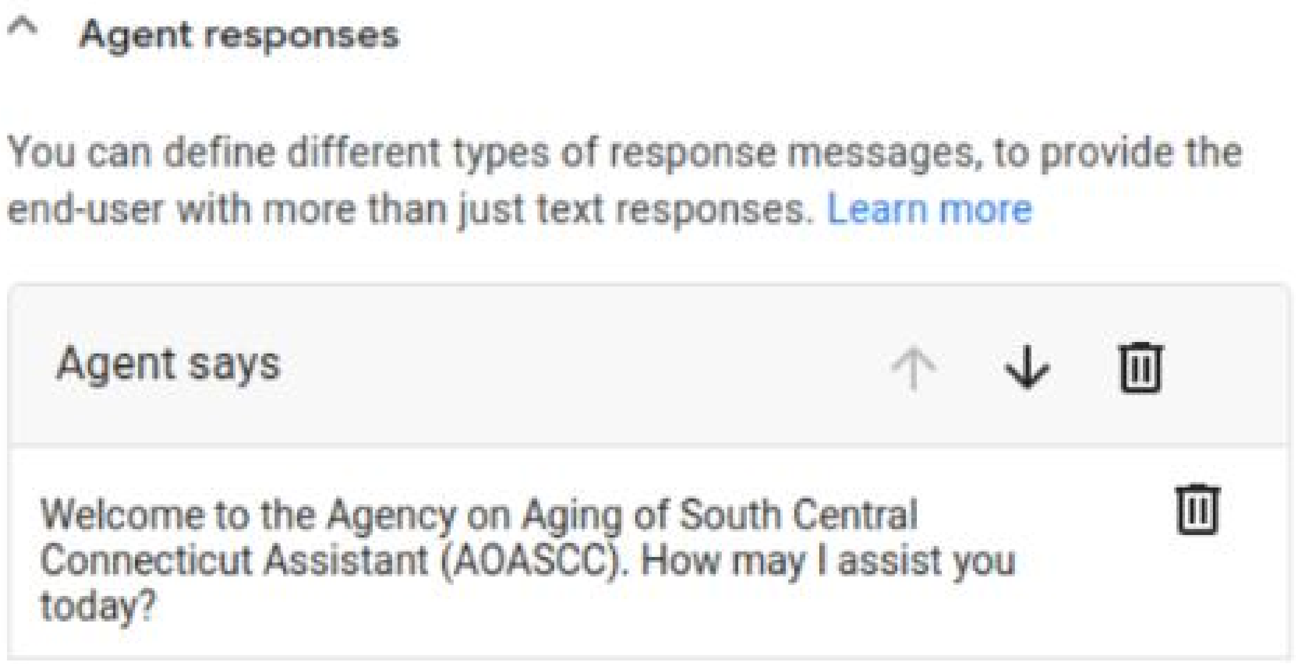
The name you provided is invalid.

Please enter your Name:

Fig xii. Result of validating Form Inputs

9.2 Fulfillments or Agent Response

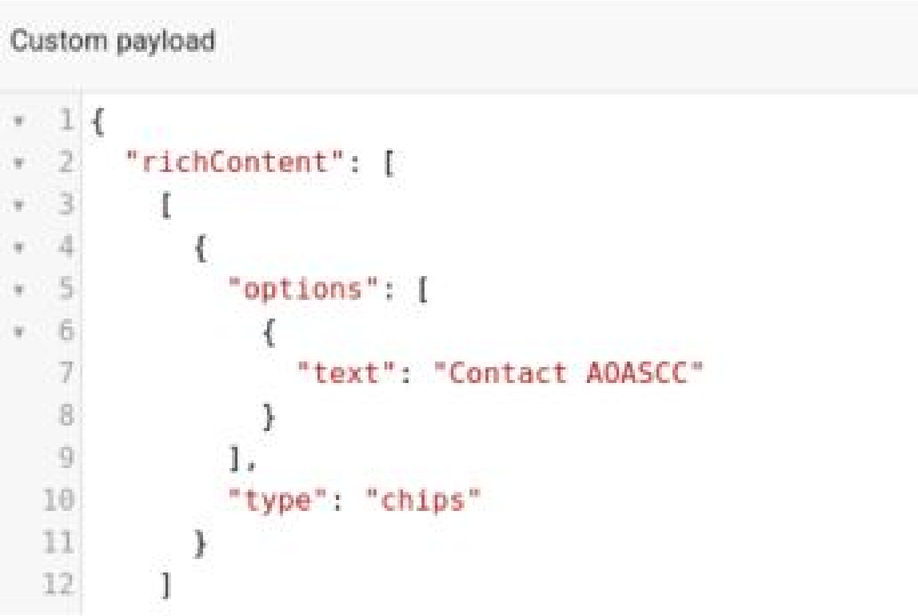
In Dialogflow CX, fulfillments refer to the actions that the agent performs in response to a user's intent. When a user's query or input matches an intent, Dialogflow CX can trigger a fulfillment to take actions such as providing a response. Response could be in the form of payload or text.



You can define different types of response messages, to provide the end-user with more than just text responses. Learn more

Welcome to the Agency on Aging of South Central Connecticut Assistant (AOASCC). How may I assist you today?

Fig xiii. Text response



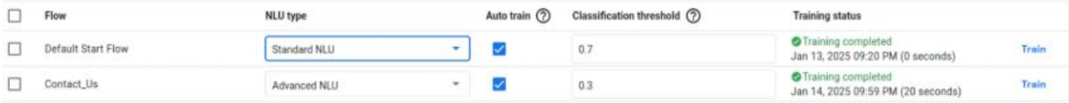
Custom payload

```
{
  "richContent": [
    [
      {
        "options": [
          {
            "text": "Contact AOASCC"
          }
        ],
        "type": "chips"
      }
    ]
```

Fig xiv. Custom payload response

10. Managing NLU Settings for Flows in Dialogflow CX

In Dialogflow CX, Each flow can use either Standard NLU or Advanced NLU to process user inputs. The Auto Train option ensures the model automatically updates when changes are made to intents or training phrases.



☐	Flow	NLU type	Auto train ?	Classification threshold ?	Training status	
☐	Default Start Flow	Standard NLU	☑	0.7	Training completed Jan 13, 2025 09:20 PM (0 seconds)	Train
☐	Contact_Us	Advanced NLU	☑	0.3	Training completed Jan 14, 2025 09:59 PM (20 seconds)	Train

Fig xv. Natural Language Understanding (NLU) settings

The Classification Threshold defines the minimum confidence score required for the agent to match an intent, helping control accuracy versus fallback responses. In this example, the "Default Start Flow" uses Standard NLU with a threshold of 0.7, while "Contact_Us" uses Advanced NLU with a threshold of 0.3.

11. Multi Channel Deployment Strategies

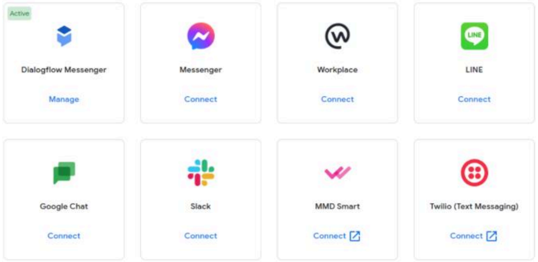


Fig xvi. Text-Based Channel Integrations in Dialogflow CX

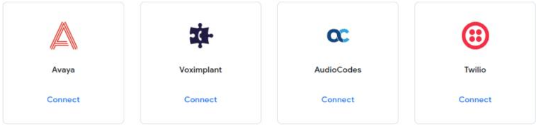


Fig xvii. One click Telephony

Multi-channel deployment in Dialogflow CX enables businesses to connect their virtual agents seamlessly to various communication platforms, enhancing user accessibility and engagement. With prebuilt integrations like Messenger, Slack, Twilio, Telegram, Viber, Discord and Google Chat, agents can be quickly deployed across popular text-based platforms. Each channel can be customized to meet specific user interaction requirements, such as rich responses or platform-specific features. This strategy ensures consistent conversational experiences while allowing flexibility to adapt to the unique needs of each platform. Effective multi-channel deployment boosts customer satisfaction by reaching users where they are most comfortable.

12. Conclusion

Dialogflow CX represents a paradigm shift in how businesses approach customer service and interaction. By leveraging advanced conversational AI, it effectively addresses the challenges of long response times, limited agent availability, inconsistent service quality, and scalability issues, moving beyond the primitive chatbots of the past. Its robust state machine architecture, combined with powerful NLP capabilities, allows for the creation of dynamic, context-aware conversational agents that can seamlessly adapt to diverse user needs. The platform's support for multi-channel deployment further enhances accessibility, ensuring that users can interact with the virtual agent through their preferred platforms. As highlighted, Dialogflow CX provides a practical and versatile solution for enhancing customer experience while simultaneously optimizing operational efficiencies and reducing costs. Businesses can leverage the capabilities detailed in this white paper to create robust, scalable, and user-centric virtual agents, transforming their customer support into a faster, more reliable, and satisfying experience. Ultimately, Dialogflow CX empowers organizations to not only meet but exceed the ever-evolving expectations of today's customers.

About the Authors

Author 1 Name: Yunij Karki
[Author’s Title] at DevDash Labs
Author’s Bio

Author 2 Name: Yunij Karki
[Author’s Title] at DevDash Labs
Author’s Bio

Author 3 Name: Yunij Karki
[Author’s Title] at DevDash Labs
Author’s Bio

DevDash Labs

At DevDash, we bridge AI innovation with business reality. We're more than AI implementers – we're transformation partners. We are a US-based AI research and development company with a global reach. We combine cutting-edge research with practical business solutions to help organizations navigate and capitalize on the AI revolution.