

Technical Disclosure Commons

Defensive Publications Series

December 19, 2016

Executable Knowledge Base for Virtual Chat System

Wen-Ying Yang
Hewlett Packard Enterprise

Rui Han
Hewlett Packard Enterprise

Ding-Jun Chen
Hewlett Packard Enterprise

Xun Wang
Hewlett Packard Enterprise

Yuan-Chun Liao
Hewlett Packard Enterprise

Follow this and additional works at: http://www.tdcommons.org/dpubs_series

Recommended Citation

Yang, Wen-Ying; Han, Rui; Chen, Ding-Jun; Wang, Xun; and Liao, Yuan-Chun, "Executable Knowledge Base for Virtual Chat System", Technical Disclosure Commons, (December 19, 2016)
http://www.tdcommons.org/dpubs_series/357



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

Executable Knowledge Base for Virtual Chat System

Abstract

A virtual chat system enables the end user to interact with knowledge base by chatting with a virtual assistant. Besides knowledge article, a virtual assistant can also perform automation flows such as restart a virtual machine, reset the password for a PC. In many virtual chat systems, AIML (Artificial Intelligence Markup Language) is used to train the virtual agent to interact with human beings. It is also possible to integrate knowledge system and automation flow system with AIML interpreter to quickly empower virtual assistances with various domain knowledge. The disclosure provides a method to convert or link an automation flow to virtual agent understandable and executable format and enable them to perform and interact seamlessly with the users, the knowledge base system and the automation system.

Description

This disclosure relates to the field of virtual chat system which enables end users to chat with virtual assistants in order to resolve problems. A method to enable virtual assistants seamlessly interact with the users, the knowledge base system and the automation system is disclosed. Prior to this idea, a virtual assistant of the chat system can response an end user with a link or the content of a knowledge article. Through AIML and RESTful APIs, virtual assistant is able to interact with external systems, execute automation flow, and respond the message received from users. However, three major flaws make the application user unfriendly and unsustainable.

1. Knowledge system and automation system are two separated systems which are used to train the virtual agent, but they contain duplicated contents which reduce the performance of virtual assistants.
2. An automation flow can be used in different scenarios, which require the virtual chat system to maintain multiple AIML patterns to match the AIML template which will execute the automation flow. AIML is not a strong relation markup language, causing it difficult for the admin to maintain these patterns.
3. Some automation flows require multiple parameters for one-time execution. In order to write correct AIML templates to interact with the user, the AIML editor should be familiar with such automation flows. Also, the description of an automation flow is often too generic for the end user to understand. The complexity of AIML maintenance will increase while its testability will decrease.

The idea discloses a method to embed an automation flow to a knowledge article and enable virtual assistants to interact seamlessly with the users. Consider Fig1 as an example of a service desk system.

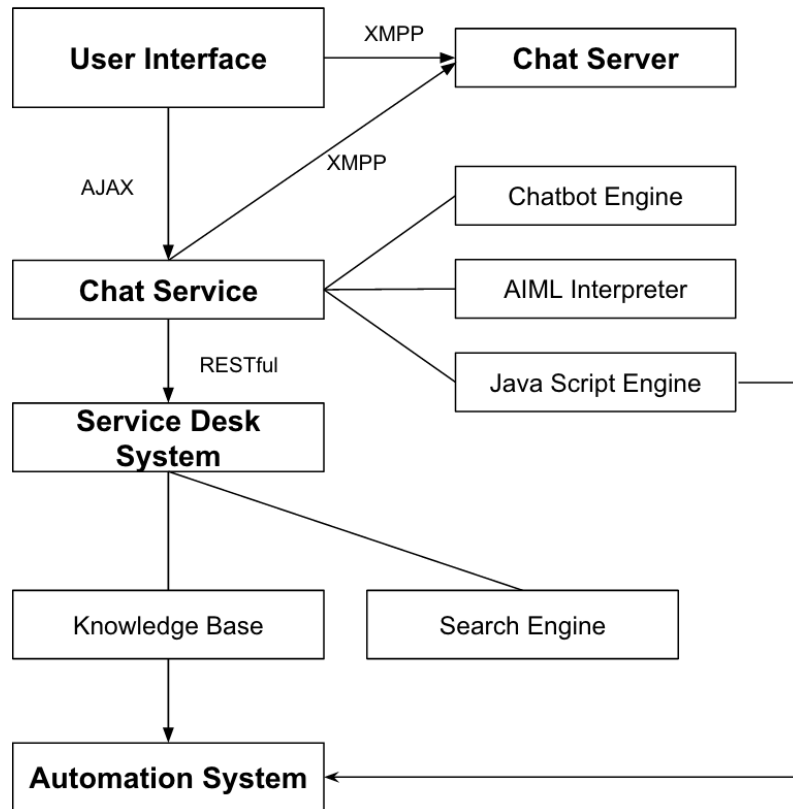


Fig. 1 Service Desk System Components

- User interface is where end users chat with virtual assistants, and it can be a chat window of a web portal or a chat widget of a mobile application.
- Chat server is an XMPP (Extensible Messaging and Presence Protocol) communication server which can be implemented to support other real-time communication protocols.
- Chat service manages chat sessions and integrates with service desk system and automation system. It includes three components.
 - Chatbot engine maintains the lifecycle of virtual assistants.
 - AIML interpreter is compliant to AIML standard and can be implemented by JAVA, Python or any other programming languages.
 - The JavaScript Engine can run java scripts with input parameters when an AIML template is matched and contains JavaScript tag.
- Service desk system includes a knowledge base and search engine, and it also has a ticketing system to maintain the support cases and related information such as end user profile, live agent profile, assets, etc. The system also provides web service APIs for external systems to search the knowledge base.
- Automation System is a system containing the automation flows and the procedures connecting these flows. It provides web service API for an external system to retrieve and execute automation flows.

Fig 2 illustrates the flow of an interaction between a user and a virtual assistant of using executable knowledge article to execute an automation flow.

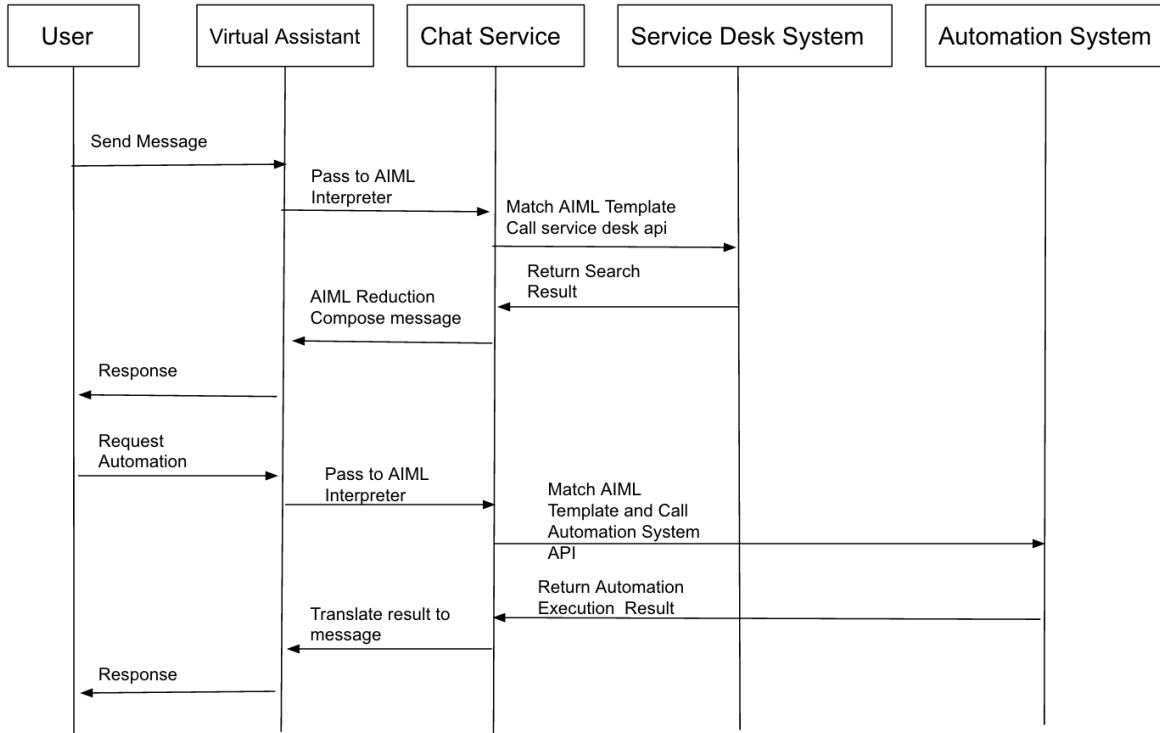


Fig. 2 Flow Chart of Virtual Chat System

The method requires automation flows manually or automatically to be converted or linked to a knowledge article in the knowledge. The knowledge article with an automation flow should store necessary information including a RESTful URL of the automation flow, the parameters used to execute the automation flow, and at least one successful return and one failure return of the automation execution result. The virtual assistant in the chatbot engine can suggest the knowledge article to an end user and trigger the automation flow linked to the article according to the user response. The execution result will be converted from a machine-readable message to a human-readable message and sent to the user. This idea reduces the complexity of creating and training a virtual agent, harnesses the power of existing knowledge and automation systems, offers better end user experience by enabling virtual agents to resolve issues automatically.

Disclosed by Wen-Ying Yang, Rui Han, Ding-Jun Chen, Xun Wang and Yuan-Chun Liao, Hewlett Packard Enterprise