

ChatterbotProposal

Abstract

[ChatterBot](#) is a lightweight, multiprotocol framework and container for chat responders.

Proposal

[ChatterBot](#) is a framework for developing chat responders (applications that respond to messages received via online chat) and a container for deploying them. It is written in Java SE, and runs as a Java application. Chat responders are built by extending a single class and modifying a configuration file to reference the new class. [ChatterBot](#)'s focus is on the following characteristics:

- Small: The current framework consists of eight core classes.
- Standalone: [ChatterBot](#) does not require external servers to operate.
- Portable: [ChatterBot](#) should work as run from any Java-capable machine. For full functionality that machine should have internet access, but localhost and console connectivity are possible. It should be possible to run multiple instances of [ChatterBot](#) on the same machine or on separate machines with no loss of functionality.
- Extensible: An instance of [ChatterBot](#) can support multiple message parsers and protocols. Adding more is done by editing a configuration file.
- Dynamic: Activating and de-activating modules should be possible during runtime.
- Multi-user access: Multiple users, over multiple protocols, should be able to access deployed applications.

Rationale

A chat responder can serve as a user interface to applications, either those built into the responder or external applications with which the responder communicates. Such an interface is more secure than interfaces such as Telnet or web services since it does not require open ports in the firewall; the container connects out through the firewall to the chat server, rather than allowing users to connect in.

A lightweight chat responder can be installed on any system to allow command-line access to users over whatever protocol a user may have access to. Thus applications can be accessed from web interfaces, instant-message systems, text messages, email, etc. A scalable container can allow as many or as few access protocols as are desired.

[ChatterBot](#), therefore, has value for those circumstances where a user interface is needed but a server-based or enterprise solution is either not possible or not desired. It also can serve as a bridge between applications, where one or more uses a chat protocol such as XMPP to communicate.

Background

[ChatterBot](#) began in 2005 as a thin-server approach to online multi-user board games, implemented as applets sending gamestate changes to one another via message relaying. The idea was to make as general-purpose a server as possible, so that multiple games could be built that employed the same message-relaying system.

Version 0.2 of the server was then refined in 2008 to allow for more varied and functional message-handlers, and was used to implement a room system that allowed for room-specific applications – parsers that checked the user's room before handling a command and sent responses to other room occupants. This version was structured entirely around XMPP. The global namespace was introduced to allow modules to communicate with relatively limited coupling.

Version, 0.3, as of late 2009, functions with XMPP and has the capacity to function with whatever other protocols channels are coded for. V0.3, though, uses a custom shell, with rudimentary module lifecycle capability.

This proposal introduces version 0.4, to be based on OSGi for module lifecycle management and event-driven module synchronization. Applications originally built for v0.2 will be ported to v0.4.

Current Status

Meritocracy

Peer review and alternate ideas are welcomed in this project with open arms. This project was intended specifically as an alternative to traditional server-based or enterprise architecture; however, it is recognized that tried-and-tested principles established in enterprise architecture may be applicable here.

Core Developers

As of late 2009, there is one developer, Donald Whytock (dwhytock at gmail dot com). Donald Whytock has several years of experience as a software developer, working in a variety of languages, including C, Java, Perl, PHP, [JavaScript](#) and SQL. He develops both professionally and casually; [ChatterBot](#) has been an independent, voluntary effort.

Alignment

[ChatterBot](#)'s primary potential alignment with ASF is that of a framework for internet-accessible applications. As command parsers can be built to interface with other applications, [ChatterBot](#) can be employed as a general-purpose remote console operating over instant messages.

ASF projects we anticipate using in [ChatterBot](#) include:

- Felix: to replace the v0.3 Shell as a module lifecycle management framework
- UIMA: to aid in parsing/analyzing messages, either in individual command parsers or in the parser dispatcher

Initial Goals

[ChatterBot](#) v0.3 exists as a functioning prototype, but does not conform to existing standards in many areas, and needs expansion in its functionality. To this end, the project recognizes the following goals:

- Conversion to Apache Felix as a core framework. This will replace the existing Shell and affect all other modules.
- Proposal of a standard for chat-protocol handlers, independent of [ChatterBot](#)'s specific needs.
- Development of handlers for multiple chat protocols, compliant with the proposed standard, for use by [ChatterBot](#).
- Identification/development and assembly of tools useful for parsing, interpreting and processing text commands.

Known Risks

Orphaned Products

Currently the project has only two committers, though Donald Whytock has been working on the code for a few years and is committed to seeing a functional product available.

Inexperience with Open Source

While the developer has experience working with open-source products, this is the first time opening up a project for open-source collaboration. As modular as the project is, however, open-source collaboration should not be a problem. It is greatly desired that this project not be developed in a vacuum.

Fascination with the Apache Brand

Association with the Apache brand is not sought for personal publicity; rather, it is sought for the associated community and access to collaboration and peer review. This project will see its full potential through public use and refinement, and a product more refined for everyone's use is a more refined product for the developer's use as well.

Initial Source

Original code developed by Donald Whytock.

Required Resources

Mailing Lists

- chatterbot-private
- chatterbot-dev

Subversion Directory

<https://svn.apache.org/repos/asf/incubator/chatterbot>

Issue Tracking

JIRA [ChatterBot](#)

Initial Committers

- Christopher Brind (brindy at windy dot org dot uk)
- Donald Whytock (dwhytock at gmail dot com)