

# BotSense: An Open-Source Middleware Framework for Unifying and Enabling Intelligent Robotic Systems

Kim Wheeler<sup>†</sup>, Jerry Weinberg<sup>‡</sup>, Robin Knight<sup>†</sup>, William Yu<sup>‡</sup>, Collin Horvat<sup>†</sup>, Jason Tennyson<sup>‡</sup>, Casey Kuhns<sup>†</sup>, Aaron Parker<sup>‡</sup>, and Jeff Croxell<sup>‡</sup>

<sup>†</sup>RoadNarrows LLC, <sup>‡</sup>Southern Illinois University Edwardsville

RoadNarrows LLC, 1151 Eagle Drive #140, Loveland, CO 80537

Southern Illinois University Edwardsville, Department of Computer Science, Edwardsville, IL 62026

kim.wheeler@roadnarrows.com, jweinbe@sie.edu

## Abstract

The *BotSense* software package provides a client-server framework for clients to remotely interface to proxied server devices over IP. Additional proxied devices may be easily added by developers using the open and documented programming interface.

## Introduction

The *BotSense* effort is motivated by the need of RoadNarrows to support diverse robotic systems and remote embedded devices in unified and virtualized frameworks over the ubiquitous Internet. Client applications using *BotSense* may then be implemented rapidly to a consistent, common, open API.

Figure 1 illustrates the *BotSense* framework. The bsProxy server plus any specific module plug-ins operate on the target system. The target system has direct access to the devices and resources to be proxied. The client applications may run either on or off target. Compiled C/C++ applications are linked to the *BotSense* libraries. Python clients use the modules imported from the *BotSense* python package.

## Objective

The objective is to create an IP-based, open-source, near real-time client-server middleware framework that readily supports multiple computing architectures including embedded processors with limited resources. The proxy server supports concurrent clients. Both request-response and streaming-out message exchange patterns are supported. The extensible framework easily supports application-specific new proxied (pseudo) devices and robotic systems.

---

Copyright © 2010, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

## Architecture

The **BotSense** architecture is summarized in the following sections.

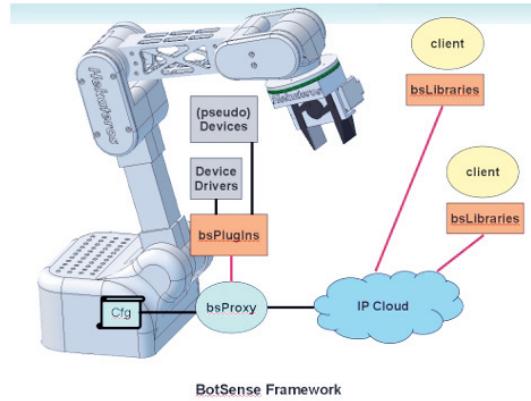


Figure 1: BotSense Framework

## Project Constraints

- Provide open-source with well documented and published interfaces.
- Core system written in C to the POSIX compliant interface standards.
- Use the cross-compiler, multi-platform RoadNarrows **rnmake** make system built around GNU make.
- Use RoadNarrows **NetMsgs** package for client-server and module-device message XML specifications with auto-code generation.
- Fully doxygen source documentation.
- Provide a Programmer's Guide. Use subversion for source control.

## Invariant Core

- Software written in C to the POSIX compliant interface standards.
- **BotSense** interface module plug-ins support multiple languages: C, C++, and Python (to be expanded in the future).
- Proxy server and core client library are multi-threaded, multi-client, and real-time optimized.
- Approach is validated by support of legged and wheeled robots and other intelligent devices.

## BotSense bsProxy Server

- Multi-threaded, multi-client, and real-time optimized.
- Multiple instances can run on one target server – each listening to a different IP port.
- XML configuration file(s).
- 16 clients per bsProxy server (default, configurable).
- 255 virtual connections per bsProxy server.
- Message Exchange Patterns: request-response, stream-out.

## BotSense libbotsense Client Library

- Multi-threaded safe, multi-client support, and real-time optimized.
- Automatic receive message ordering.
- Message Exchange Patterns: request-response, stream-out.

## Application-Specific BotSense Plug-Ins

- **BotSense** plug-ins support multiple languages.
- Well defined plug-in interfaces for server.
- **BotSense** Standard Modules: raw serial, raw I2C, pipes, shared memory.
- Client-server and module-device message XML specifications with auto-code generation.

## Supported Platforms

### Proxied Robots

The list of current and planned near-term support for **BotSense** proxied robots and intelligent systems include:

- K-Team's Khepera III and Koala robots.
- RoadNarrows RoboSight neural-network camera.
- RoadNarrows Hekateros robotic manipulator.

- RoadNarrows SkewlZone legged robots.



### Third Party Software Interfaces

The list of current and planned **BotSense** support for third party software includes:

- Tekkotsu: An open-source robot framework.
- OpenRave: Open robotics automation virtual environment.
- ROS: Willow Garage's Robot Operating System
- Myro: Institute for Personal Robots in Education python software.

## Results

The SIUE team is using CMU's Tekkotsu – an open-source robotic development framework – with BotSense on legged robots. Implementation and results key points:

- Mirage simulator used to design and test behaviors.
- Video streaming fed into built-in vision engine.
- Finite State Machine parser for complex behaviors.
- Kinematics are computed for robot behaviors.
- Robot driver calls **BotSense** function to move servos.
- **BotSense** polled for sensor data and servo positions.

## Acknowledgements

**BotSense** and the SkewlZone projects are partially supported by the National Science Foundation Small Business Technology Transfer Research (STTR) Program under Grant No. 0848762.

## References

RoadNarrows BotSense source and on-line documentation:  
<http://www.roadnarrows.com/distro/botsense>

RoadNarrows Research and Engineering Projects:  
<http://www.roadnarrows.com/robotics/engineering-research-projects>

Tekkotsu Open-Source Robotic Framework home page:  
<http://www.tekkotsu.org>

Computer Science @ SIUE  
<http://www.cs.siue.edu/>

Willow Garage home page:  
<http://www.willowgarage.com/>

IPRE Wiki main page:  
[http://wiki.roboteducation.org/Main\\_Page](http://wiki.roboteducation.org/Main_Page)