RobotRaconteur®

Robot Raconteur Version 0.8: An Updated Communication System for Robotics, Automation, and the Internet of Things

Dr. John Wason

Wason Technology, LLC
PO Box 669, Tuxedo, NY, 10987
wason@wasontech.com

August 23, 2016

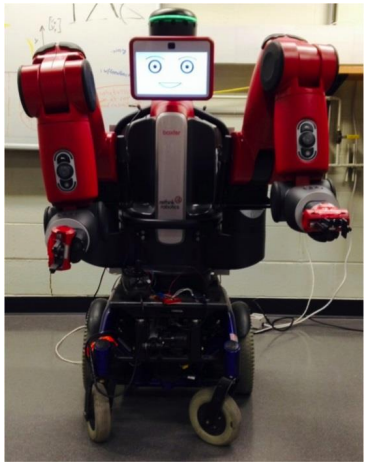
http://robotraconteur.com

RobotRaconteur®

Introduction

Language, platform, and transport independent communication system for Robotics and Automation Systems

- Compatibility
  - 22 platforms/architectures
  - 7 languages
  - 5 transport technologies
- "Plug-and-Play", "Instant-On" capability
- "Augmented Object-Oriented" model
- Transactional, streaming, and "most recent"
- Client-service and service-client
- TLS, certificates, and password security
- Compatible with Web and Cloud
- Node and service discovery
- Version 0.8 ready for commercial use



RobotRaconteur®

Motivation

A new robot and Kinect in the lab;

or multiple spectrally controllable lights, multiple spectral sensors, and multiple cameras;

or multiple robots with individual controllers and force/torque sensors …

How to get them to talk each other and program and control them?

RobotRaconteur®

Design Objectives

- Peer-to-peer client-service communication
- Ease of use
  - Minimal “boiler-plate” software development
  - Automatic type generation at runtime or design-time
  - Model similar to “object-oriented” systems
  - Rapid modification of services
- Maximum compatibility between devices over long timespan
  - Decades, not months of compatibility
- High performance, low latency
- Industry standard security (TLS, certificates, etc.)
- Automatic service discovery
- Asynchronous operation
- Minimal dependencies

RobotRaconteur®

Choices of Distributed Architectures

- General Purpose (using Remote Procedure Call, RPC)
  - Distributed Component Object Model (DCOM) ➔ .NET Remoting
  - Java Remote Method Invocation (RMI) ➔ Common Object Request Broker Architecture (CORBA)
- Robotics
  - Robot Operating System (ROS)
  - Data Distribution Service (DDS)
  - ZeroMQ/Protobuf (Ignition Transport)

- Expose functions or objects across network boundary
- Transparency achieved through serialization
- Some platform dependency

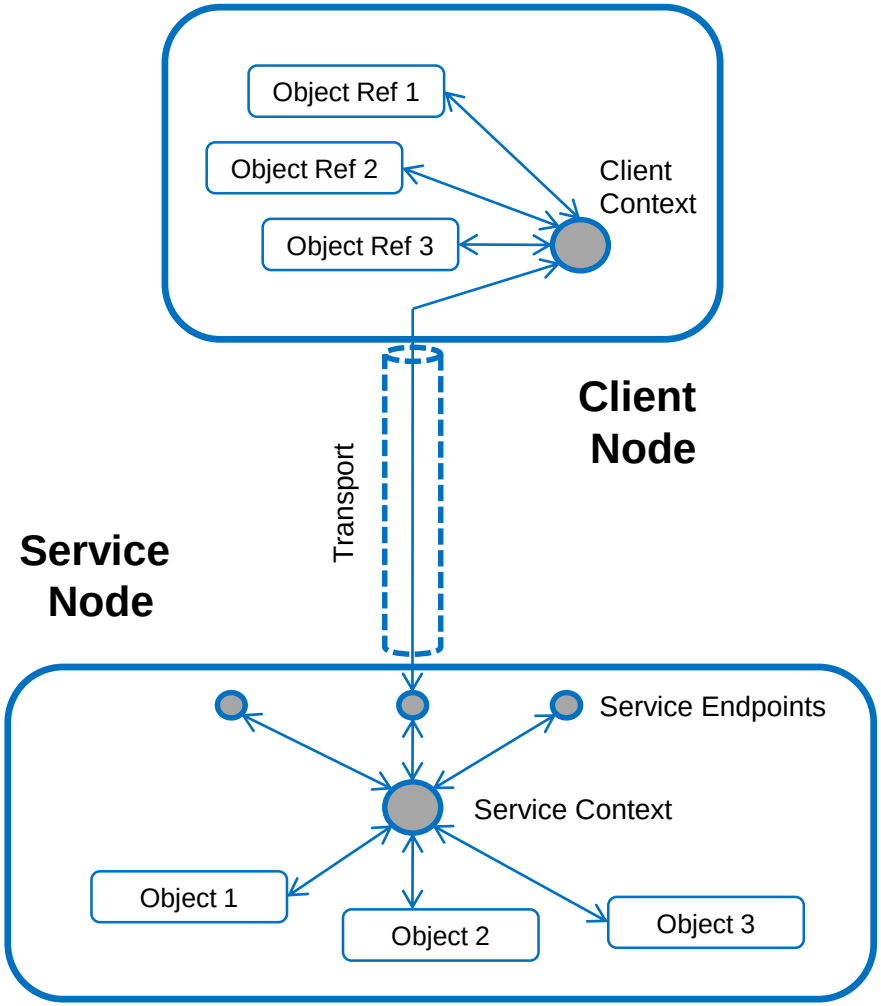
RobotRaconteur®

Compatibility

- Platforms/architectures
  - Windows (x86, x64)
  - Linux (x86, x86_64, ARM hard-float, ARM soft-float, PowerPC, MIPS
  - Mac OSX
  - Android (ARM, x86)
  - iOS
  - Web Browser (Chrome, IE, Edge, Firefox, Safari)
  - ASP.NET Web Server
  - Linux PREEMPT_RT
  - MathWorks xPC Target
  - Particle Photon embedded device
  - Arduino embedded device
- Languages
  - C++, C#, Java, MATLAB, JavaScript, LabView
- Transports
  - TCP, Local, Cloud, USB, PCIe
- Compatibility constantly expanding, grey are experimental

RobotRaconteur®

Client-Service Model



- Service: Base object reference with members, and references to other objects.
- Service definition file: Define object and structure members
- Supports try/catch error transmission across boundary

RobotRaconteur®

Augmented Object-Oriented Model

- Object types are defined in service definitions
- Member types:
  - Property
  - Function
  - Event
  - ObjRef (Object Reference)
  - Pipe
  - Callback
  - Wire
  - Memory
- Can use “import” to use structures and object types in other service definition
- “implements” statement used in object to inform clients that it is compatible with previous service definition version

RobotRaconteur®

Supported Value Types

- Value types always serialized and sent to the remote node
- Primitive Types
  - double, single, int8, uint8, int16, uint16, int32, uint32, int64, uint64, string
  - Single dim arrays of numeric types (ie double[])
  - Arrays can be fixed, variable, or have a maximum size (double[10], double[], double[10-]
- Structures defined within service definitions
  - Contains “fields” of other primitives
- Maps with int32 or string
  - (double{int}, string{string}, mystruct{string}, double[]{string}
  - Dictionary in C#, cell in MATLAB
- Lists
  - (double{list}, string{list}, mystruct{list}, double[]{list}
  - List in C#, cell in MATLAB
- Multidimensional complex or real array (double[*])
- varvalue – wildcard value type

RobotRaconteur®

Example Robot

- iRobot Create 1
- Raspberry Pi 3 ARM Computer
- Dual Webcams
- Power Converter
- Camera Mast and Cover
- Exposes two services
  - iRobot Create control
  - Webcam control and acquisition

RobotRaconteur®

iRobot Create Service Definition

```
#Service to provide sample interface to the iRobot Create

service experimental.create

option version 0.5

struct SensorPacket
    field uint8 ID
    field uint8[] Data
end struct

object Create
    option constant int16 DRIVE_STRAIGHT 32767
    option constant int16 SPIN_CLOCKWISE -1
    option constant int16 SPIN_COUNTERCLOCKWISE 1

    function void Drive(int16 velocity, int16 radius)
    function void StartStreaming()
    function void StopStreaming()

    property int32 DistanceTraveled
    property int32 AngleTraveled
    property uint8 Bumpers

    event Bump()

    wire SensorPacket packets

    callback uint8[] play_callback(int32 DistanceTraveled, int32 AngleTraveled)
end object
```

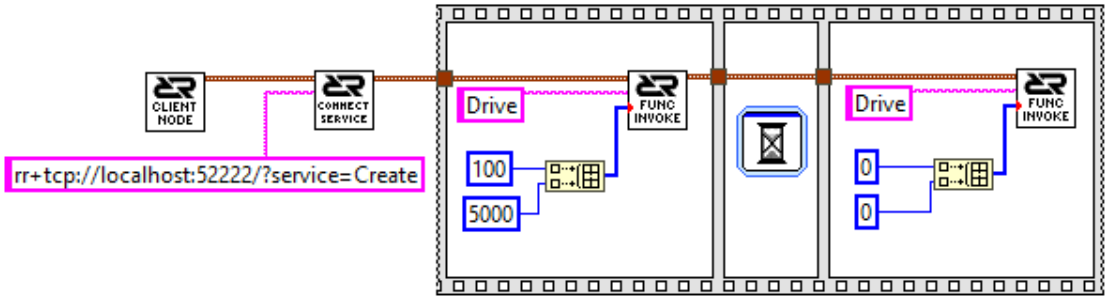
Example Clients

```python
from RobotRaconteur.Client import *
import time

obj=RRN.ConnectService('rr+tcp://localhost:52222/?service=Create')

obj.Drive(100,5000)
time.sleep(1)
obj.Drive(0,0)
```

Python

```matlab
o=RobotRaconteur.Connect('rr+tcp://localhost:52222/?service=Create');
o.Drive(int16(100),int16(5000));
pause(1);
o.Drive(int16(0),int16(0));
```

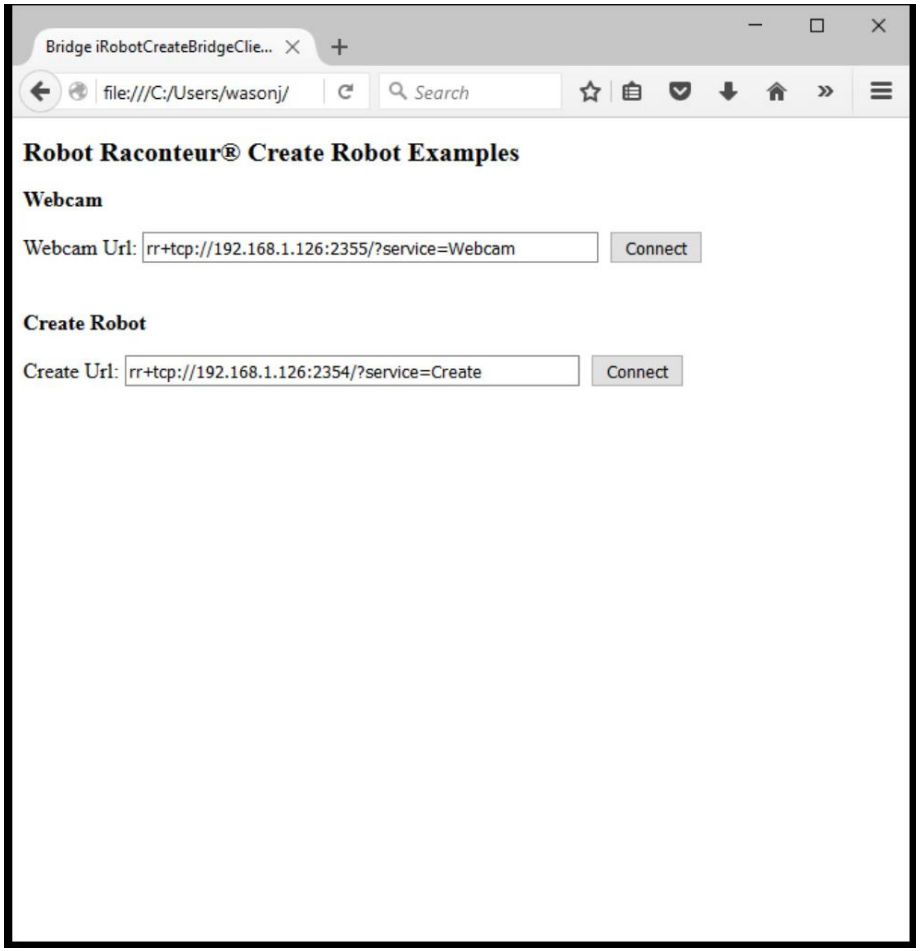
MATLAB



LabVIEW

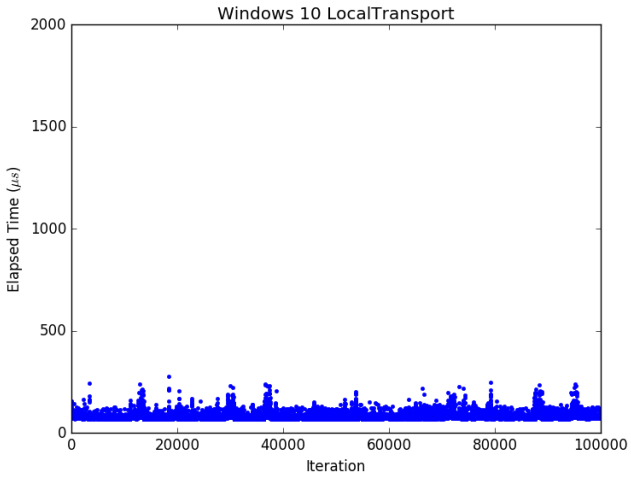
RobotRaconteur®

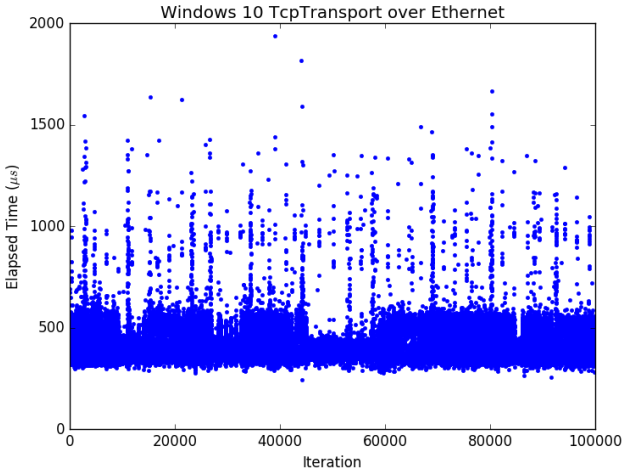
Web Browser Client Video



RobotRaconteur®

Performance



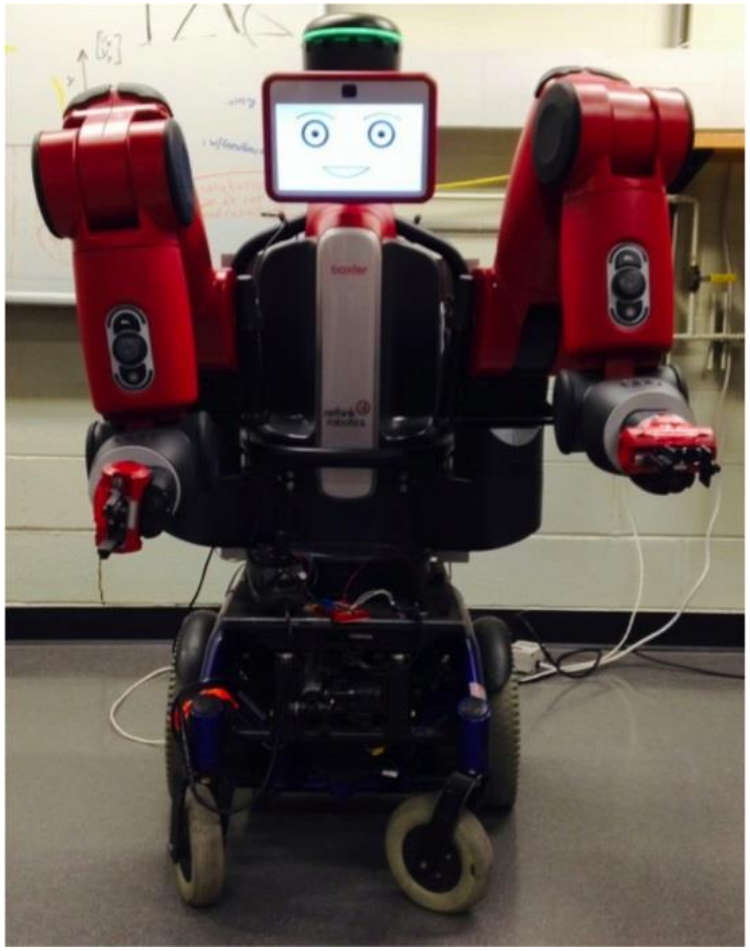


Configuration	Mean (μs)	Min (μs)	Max (μs)	σ
Windows 10 `LocalTransport`	77.1	66.4	275.9	8.9
Windows 10 `TcpTransport` Loopback	113.1	102.2	683.7	10.5
Windows 10 `TcpTransport` with TLS Loopback	141.2	128.6	931.8	12.6
Windows 10 `TcpTransport` over Ethernet	417.4	243.5	3754.9	84.7
Windows 10 `TcpTransport` with TLS over Ethernet	492.7	306.0	5859.8	79.9
Ubuntu 14.04 `LocalTransport`	136.6	126.8	956.3	8.2
Ubuntu 14.04 `TcpTransport` Loopback	166.7	151.3	1674.1	8.8
Ubuntu 14.04 `TcpTransport` with TLS Loopback	278.0	264.1	1481.4	7.9
Ubuntu 14.04 `TcpTransport` over Ethernet	518.0	303.3	7449.7	32.6
Ubuntu 14.04 `TcpTransport` with TLS over Ethernet	687.3	477.4	1605.7	47.5
Ubuntu 14.04 ROS TCP Loopback	1848.8	1362.7	14325.544	883.2

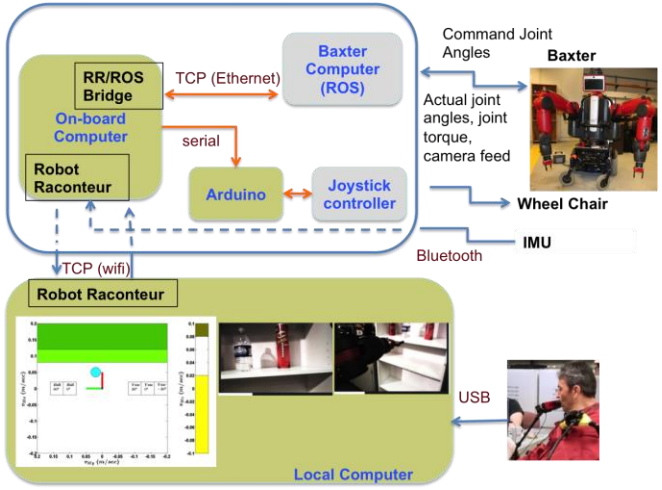
RobotRaconteur®

Baxter on Wheels

- Baxter robot with Wheelchair
- Used with a JamBoxx controller for disabled support experiments



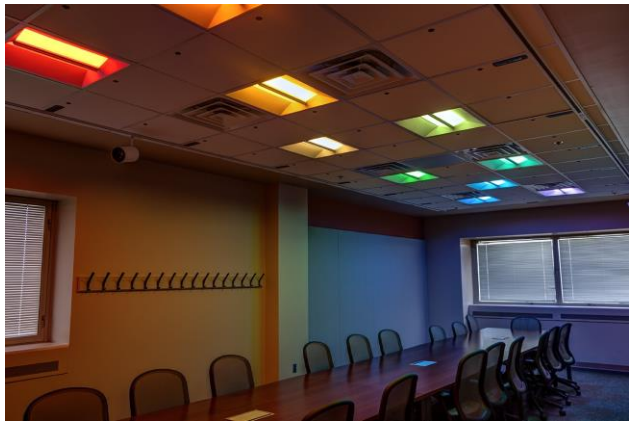
Center for Automation Technology and Systems, Rensselaer Polytechnic Institute, Troy NY



RobotRaconteur®

Smart Conference Room

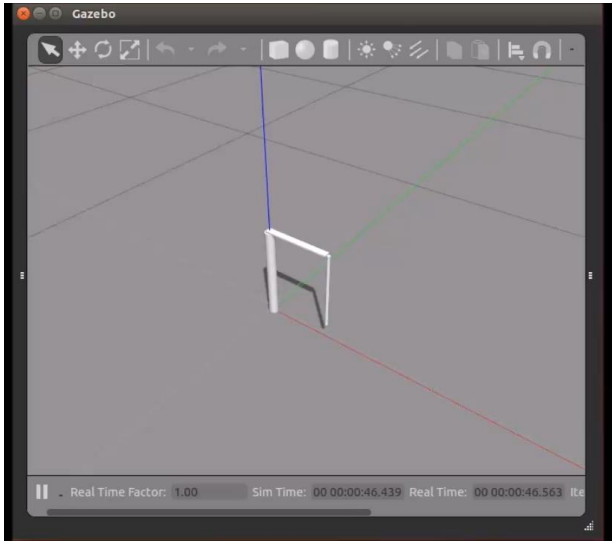
- Designed to test next-generation lighting control algorithms
- Color-tunable lights
- Light color sensors
- Depth cameras
- Climate control
- Uses Robot Raconteur for communication between components



Smart Lighting Engineering Research Center,
Rensselaer Polytechnic Institute, Troy, NY

RobotRaconteur®

Gazebo Robot Simulator Plugin



- System level Server plugin
- Provides Object-Oriented interface similar to internal C++ API
- Does not require modifying robot or world SDF files
- Access to world, controllers, and sensors
- Capable of soft real-time control

RobotRaconteur®

License

- Commercial license, Royalty Free for most uses
- Open Source projects are prone to:

Ecosystem Fragmentation

- Open Source projects tend to fork or have poor quality versions
  - Not a problem for most projects, but can be fatal to a distributed system
- Fragmentation examples:
  - Crypto-currency
  - Android
  - HTML/JavaScript
- Organizational model based on other commercial consortiums
  - USB, PCI-SIG, EtherCat, CANopen, etc.

RobotRaconteur®

Conclusion

- Robot Raconteur is a powerful communication library for system integration
- Provides “plug-and-play” capability
- Numerous advantages over existing technology
- Significant use at Rensselaer Polytechnic Institute and other university/research centers around the world
- Future work:
  - Hardware transports
  - Cloud transports
  - Real-time communication
  - Support for additional platforms/languages
  - Industrial consortium with standardization committee

RobotRaconteur®

Acknowledgement

Special thanks to the Smart Lighting Center and the Center for Automation Technologies and Systems, both at Rensselaer Polytechnic Institute, Troy, NY, for the use of example images in this presentation.