Designing a Control System for a Multifunctional Service Robot

Overview

Revision 6

January 28, 2008

TABLE OF CONTENTS

TABLE OF FUGURES

1 Introduction

The document outlines a generic architecture of a control system of a multifunctional service robot based on Skilligent technology, including a description of a typical electronics configuration, sensors and actuators.

Besides that, the document provides introductory information about the architecture of the Skilligent Robot Learning and Behavior Control System and describes what kind of custom software/firmware needs to be developed in order to integrate the off-the-shelf control system with a chosen set of commercial off-the-shelf or custom-made electronics.

2 Advantages of learning service robots

The learning capabilities of the Skilligent software give a unique capability to build truly *multifunctional* service robots. *"Multifunctional"* means that the robot can perform many functions, not just one or two. The robot can learn new behaviors when its users need it[1].

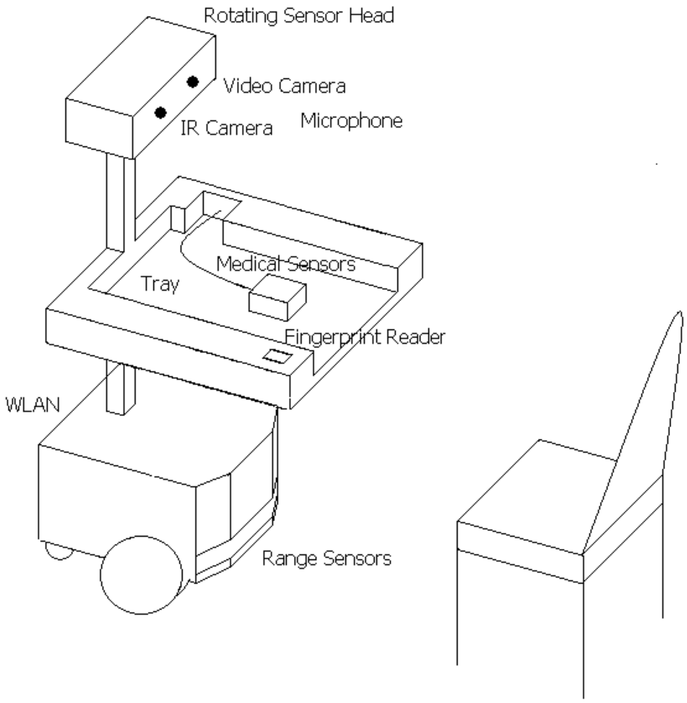

Figure 1 A Healthcare/Eldercare Service Robot

[1] Many of today's service robots perform just one function (e.g. cleaning) and are not able to do anything else.

A multifunctional autonomous service robot sketched in Figure 1 has an aluminum/plastic body and equipped with sensors, actuators, payloads (e.g. medical sensors), an onboard computer and interface controllers.

The onboard computer of the robot runs under control of the Skilligent Robot Learning and Behavior Control System, a trainable robot control system which gives the robot a capability to be trained to carry out new assignments.

For example, if the robot is bought to automate scanning of patients' health at a hospital, the robot can also be trained to carry out new assignments such as delivering medicine or newspapers around the hospital or doing security patrols.

It doesn't take any technical expertise to train a service robot as the Skilligent software enables the robot interacts with and learn from its users in a natural way – just like a dog learns new tricks.

A multifunctional service robot can be equipped with a robotic arm (Figure 2) which allows the robot to manipulate objects.

Figure 2 A Service/Research Robot Equipped with a Robotic Arm Manipulator

The ability of the service robots to acquire new skills increases the usefulness of the robots, reduces the cost of ownership and helps to faster recover the investments into the *service automation*.

3 Architecture of a Robotic Control System

Although the multifunctional service robots might have different shapes, forms and designs, they share a lot in common – their control systems follow the same design pattern. This section describes the architecture of a generic control system of a multifunctional service robot.

3.1 High-level Block Diagram

The generic robot control system consists of *a control computer* or *computers*, *interface controllers*, *sensors*, *actuators, off-the-shelf* and *custom-built software*.

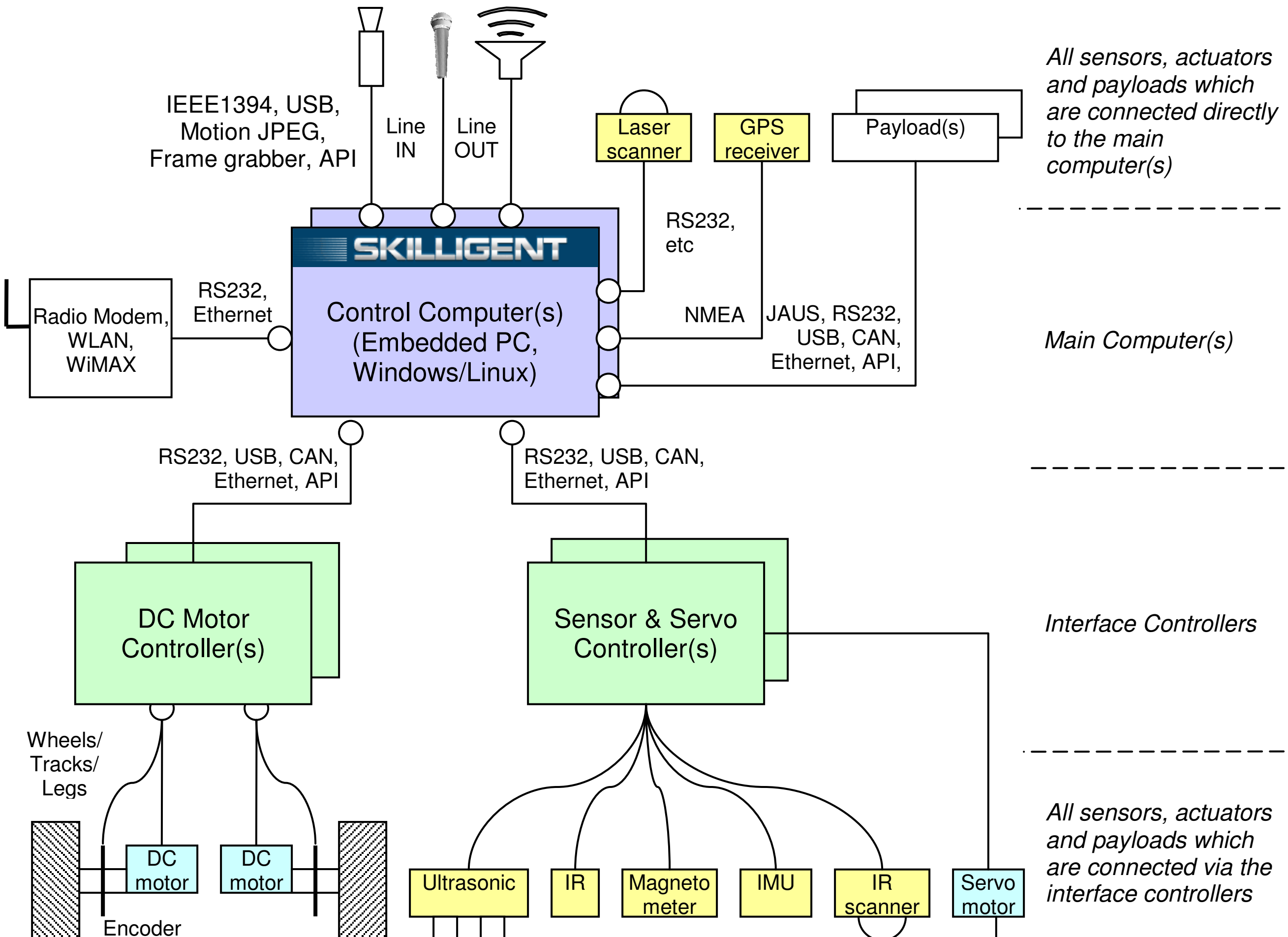

Figure 3 High-level Block Diagram[2]

[2] Not all the sensors shown on the diagram are needed for a each service automation application

The Skilligent Robot Learning and Behavior Control System is the key component of the control system as it enables the robot to autonomously behave, interact with the robot users, and acquire new skills.

The Skilligent software runs on the embedded control computer. The control computer is typically a Mini-ITX board, a PC/104 single-board computer or a telecom-grade blade server. In order to run the computer vision algorithms, a powerful CPU is required[3].

Several control computers can be connected together to form an onboard computing network (Section 3.2 Clustered Configuration, page 7). This might be needed if a single computer does not provide enough computing power to carry out complex calculations such as sensor fusion.

All *sensors* and *actuators* are connected to the main computers in the following ways:

- Directly via IEEE1394, RS232, USB or Ethernet;
- Via the *interface controllers (I/O controllers)* such as a *DC motor controller* or *a servo controller.*

The *interface controllers* (*I/O controllers*) are off-the-shelf third-party or custom-designed electronic boards/devices which are intended to perform the following functions:

- Take sensor readings, preprocess the data and forward the results to the main control computers for use in the learning and behavior control algorithms;
- Receive the commands from the control computers and send proper signals to the actuators.

The *interface controllers* are connected to the main computers via RS232, USB, CAN or other available ports. The main computers shall have enough ports to communicate with the interface controllers, payloads and sensors directly connected to the computers.

The main computers use hard-drives or solid state disks to store the binaries of the Skilligent Robot Learning and Behavior Control System and knowledge data acquired through training or experience. The main computers run under control of Linux or Windows operating systems and execute the code of Skilligent Robot Learning and Behavior Control System as a control application.

The main computer(s) can be connected to a radio modem or to a WLAN.

[3] The recommended computer configurations are published on Skilligent web site.

3.2 Clustered Configuration

Two or more main computers can be connected together to form an onboard computing network (Figure 4) in order to increase the overall available computing power. An Ethernet switch/hub needs to be used to interconnect the computers.

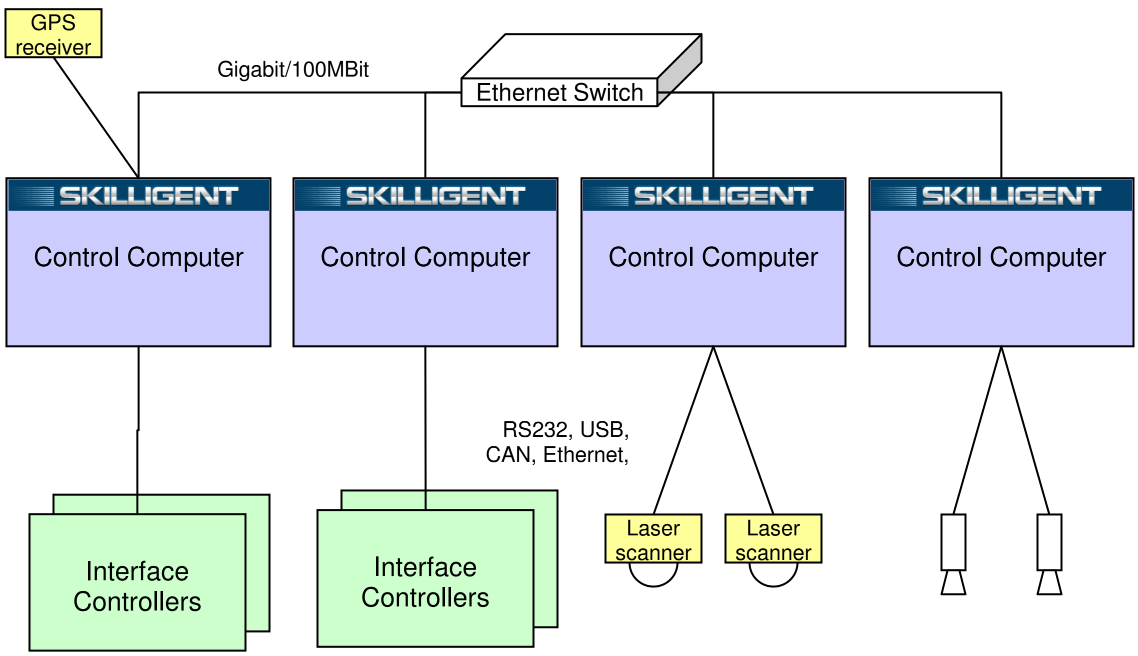

Figure 4 Clustered Configuration

This approach allows allocating dedicated computers to certain demanding tasks such as the image processing, laser scanning or sensor fusion. The Skilligent software binds all the computers together in a computing cluster[4] by providing a distributed shared memory facility, a messaging system and a unified cluster management interface.

3.3 Redundant Configuration

The Skilligent software supports[5] redundant configurations (Figure 5 Redundant Configuration). This feature enables building fault-tolerant control systems able to

[4] Microsoft Robotics Studio environment can also be used to provide such a distributed computing environment. The Skilligent software has been integrated with the MSRS platform.
[5] This feature is under development; please contact Skilligent for details.

tolerate major failures without much degradation of the overall performance - including a main computer malfunction, a software crash, or an interface controller malfunction.

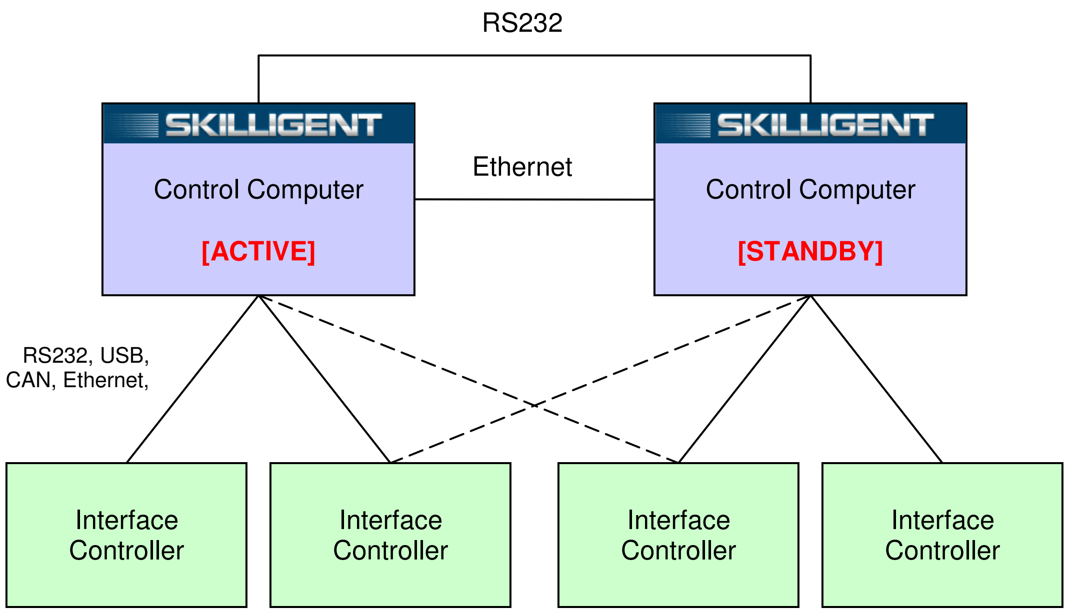

Figure 5 Redundant Configuration

In a redundant configuration, a pair of computers works together in an Active-Standby fashion. The mated computers are interconnected by an Ethernet connection accompanied by a RS232 link.

3.4 Wireless Control Configuration

It is possible that the control computers reside outside of the robot's body. The computers can be wirelessly connected to the onboard interface controllers. This way a farm of control computers grouped together in a base station controller can control multiple wireless robots (Figure 6)

The wireless controller configuration is viable when the robots (due to technical or price constraints) cannot carry onboard computers powerful enough to run the robot learning and behavior control software.

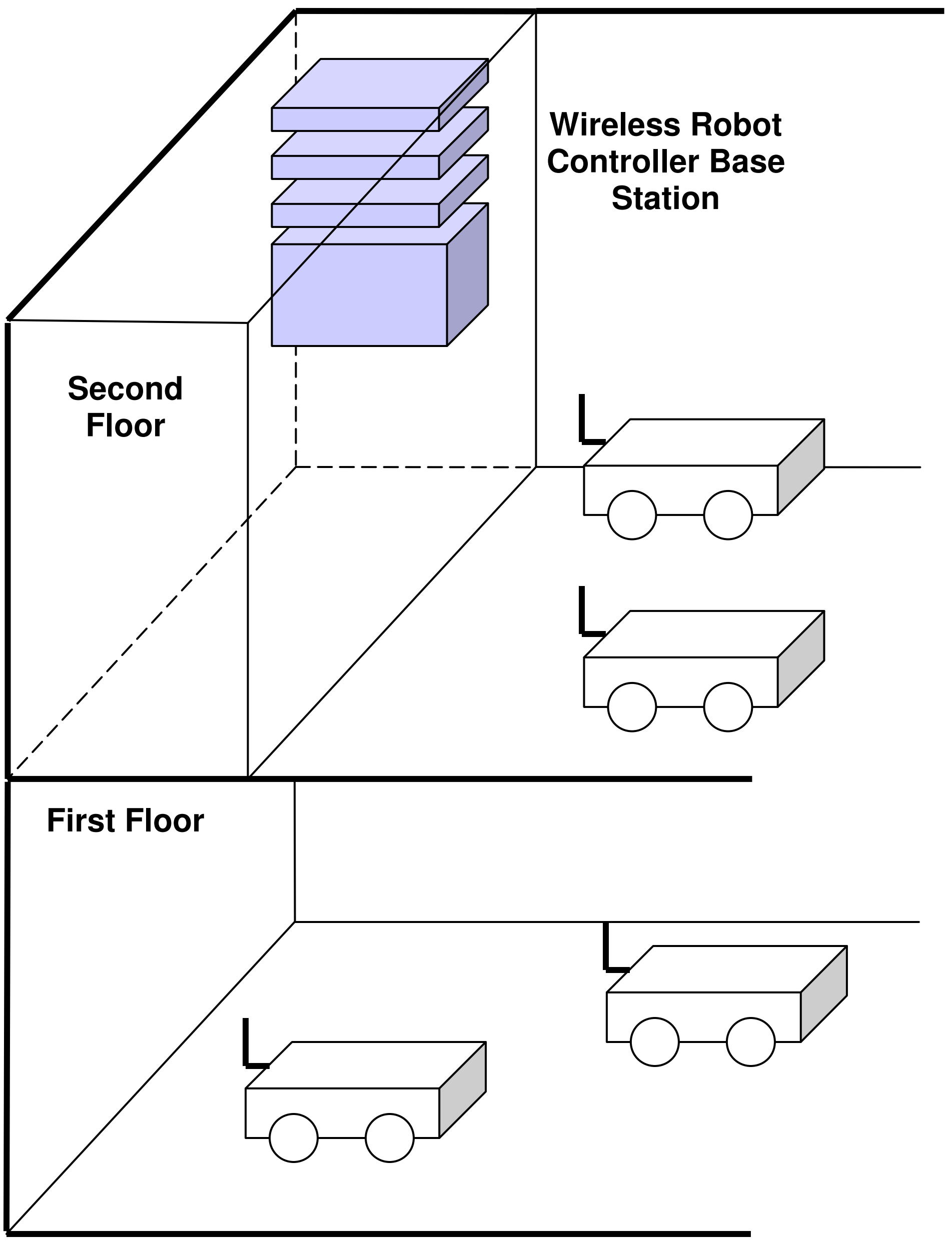

Figure 6 A clustered base station controls multiple robots in a multistory building

To reduce the price and complexity, small service robots, toy robots, low-cost surveillance robots, telepresence robots can be controlled by a control system running on a PC connected to the robot via a WLAN. Multiple control computers can be grouped together into a base station to enable sharing of its computation resources between multiple robots. The base stations can be made redundant and geographically distributed to enable operation of the robots during disasters and emergencies such as fire or an earthquake (Section 3.3 Redundant Configuration, page 7).

3.5 *Architecture of the Skilligent control software*

Skilligent Robot Learning and Behavior Control System is a software package for control systems of trainable service robots. The system consists of a number of interacting components and subsystems (Figure 7). The software components form a multilayered hierarchy of the information processing and decision making system. On the very bottom, the *device abstractions* form the lowest layer of the hierarchy, while the *behavior control*, *robot learning*, *social interaction* and *joint intention* components reside on the very top.

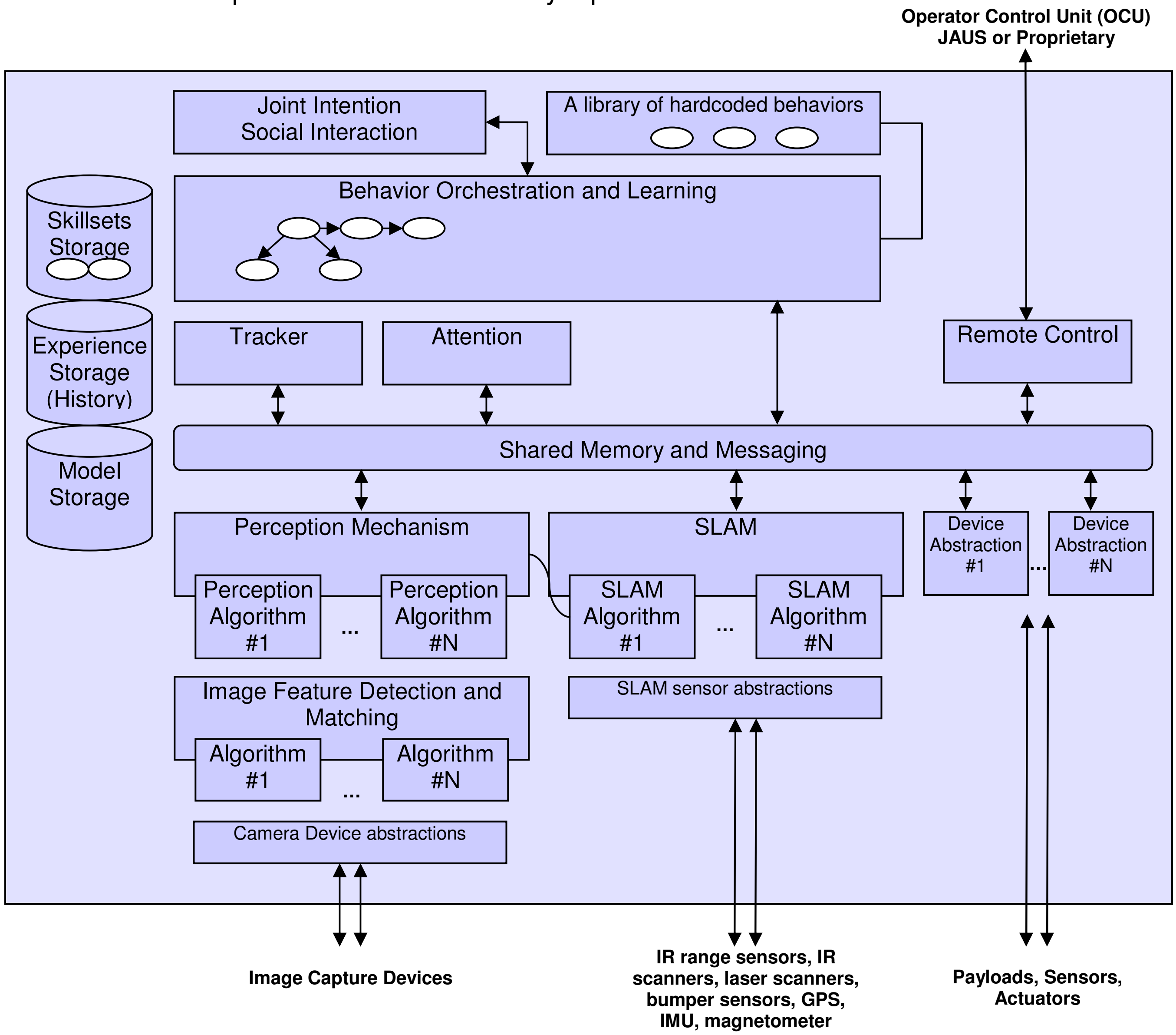

Figure 7 Logical View of Skilligent Software Architecture (some modules are not shown)

All modules are logically grouped into several subsystems such as a computer vision subsystem, a SLAM subsystem, an audio processing subsystem and a robot learning and behavior control subsystem.

The *computer vision* (CV) subsystem consists of the following modules:

- A number of *camera device abstractions* including a DirectX, Video4Linux and File Camera abstractions;
- An *Image Detection and Matching* module, a software framework for image feature detection, feature description and matching algorithms.

The *Perception Mechanism* is a software framework for various recognition methods including the *vision-based object recognition, gesture recognition*, *object selection*, *visual attention, sound recognition* and others. Its operation is based on the concept of *perceptual triggers*.

The SLAM subsystem[6] is a framework for several navigation and mapping algorithms and obstacle avoidance algorithms.

The robot learning and behavior control capabilities of the software are provided by the following modules:

- A *Behavior Orchestration and Learning* module executes behaviors, generates new behaviors and modifies existing behaviors. The module also extracts useful concepts and artifacts and various relationships between them.
- An *Attention* component selects objects, places, sounds and sensor readings which are potentially relevant to the task being learnt (bottom-up attention).
- A *Tracker* component monitors the state of the artifacts selected by the attention system and by the behavior control system (top-down attention).
- *Joint Intention and Social Interaction* modules determine goals and enable the robot to intelligently interact and collaborate with the users.
- A number of *databases* keep and process the knowledge acquired by the robot during trainings and experimentation.

From the architectural point of view, the software is based on the combination of *reactive* and *deliberative* control paradigms supported by both *event-driven* and *looping control* designs.

[6] Some parts and features of the SLAM system are under development. Please contact Skilligent for details.

4 Systems Integration

4.1 Integration with the Interface Controllers

The main control computers expose a set of standard PC ports and buses such as RS232, Firewire, USB, Ethernet, CAN or PCI, as the interfaces to other devices.

In contrast, most sensors, actuators and some payloads provide analog I/O interfaces only – and cannot be connected directly to the main computers to be used by the behavior control and robot learning software. In order to connect those sensors to the main control computers, a mediator interface controller needs to be employed. There is a large selection of commercial off-the-shelf I/O interface controllers available.

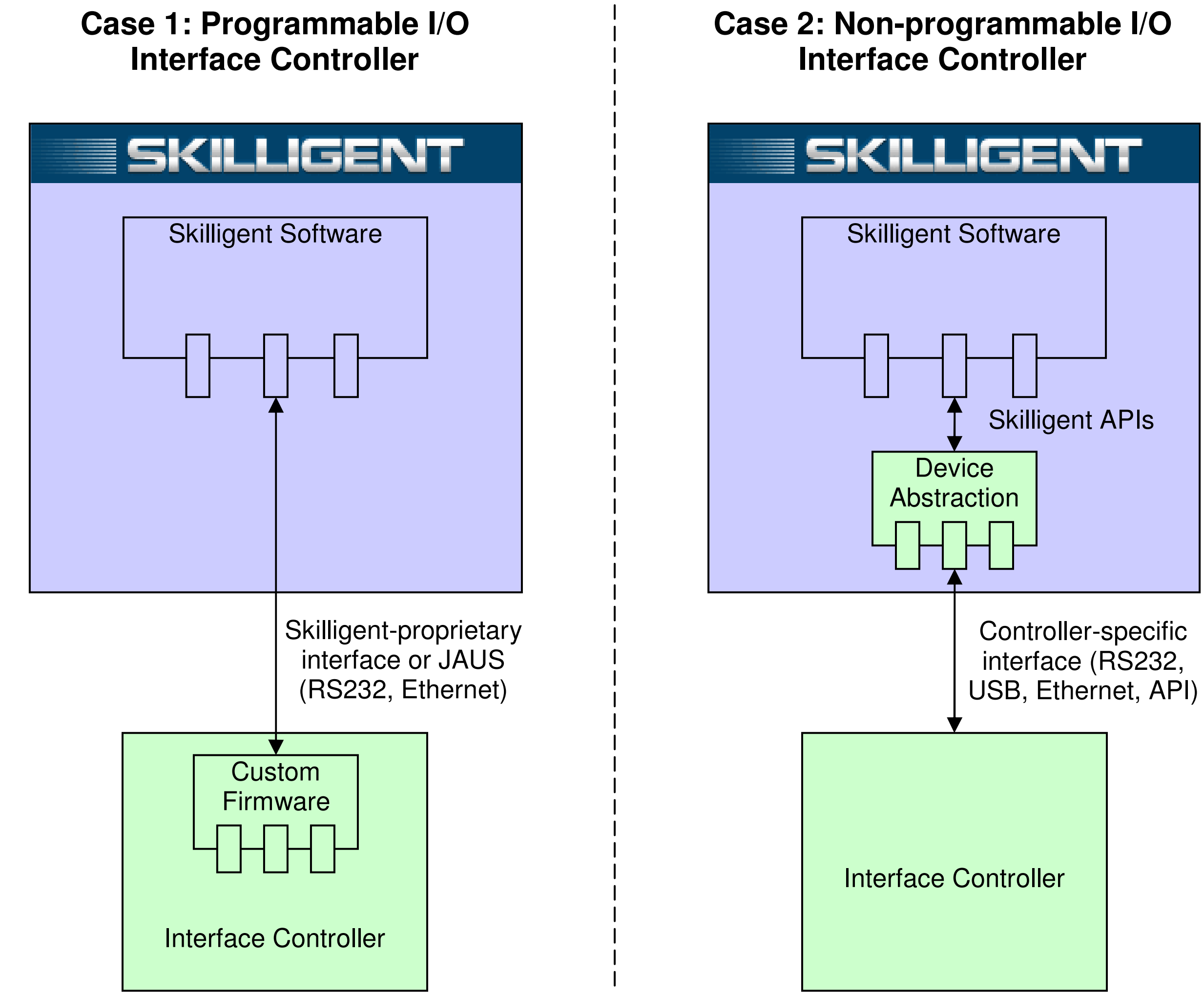

Figure 8 Integration with the interface controllers

A typical interface controller has multiple analog I/O ports which allow it to interface sensors, motors and actuators directly, and a PC port (such as CAN, RS232, Firewire or USB) which allows the controller to talk to the main computer.

When it comes to integrating the controller with the Skilligent Robot Learning and Behavior Control software (Figure 8), all the interface controllers can be divided into *programmable controllers* and *non-programmable controllers*.

The non-programmable controllers[7] use their own protocol to communicate to the main computer. In order to integrate such a controller with the Skilligent software, a special *device abstraction* code needs to be developed (Figure 8, Case 2). The device abstraction code is written in C++ according to a Skilligent-defined API specification. Look at the logical position of various *device abstractions* at the multilayered architecture shown in Figure 7.

The programmable controllers can be programmed to support any protocol to communicate to the main computer. Those controllers need to be programmed to interface to the Skilligent software via a Skilligent-defined standard protocol or through the military-standard JAUS protocol[8].

4.2 Integration with custom software libraries

Skilligent Robot Learning and Behavior Control System is designed as a expandable framework for various artificial intelligence and control algorithms. Various subsystems including the *computer vision*, *perception*, *sound recognition*, *robot learning*, *behavior control*, *social interaction* and *joint intention* can be expanded with custom-designed algorithms and the software. Please contact Skilligent for the corresponding API definitions.

4.3 Integration with Payloads

The learning capabilities of the Skilligent software allow building truly multifunctional robots. What a particular multifunctional service robot can learn to do is limited by two main factors – the capabilities of the software to acquire new skills and the limitations of the available hardware such as sensors and actuators. As the designer cannot predict all future uses of its robot, it makes sense to provide an open interface for plugging new payloads[9] to the robot.

[7] Such as Roboteq DC Motor Controllers, http://www.roboteq.com
[8] The JAUS feature is under development. Please contact Skilligent for details
[9] Sample payloads are – robotic arms, sensor packages, medical devices, vending devices, weapons, etc

The Skilligent software supports[10] military-standard JAUS interface (RS232, TCP/IP or UDP) for interacting with payload modules.

When a new kind of payload is added to the robot to enable it to perform new tasks, the payload needs to be interfaced to the Skilligent control system. This can be done in the following ways:

- If the payload is simple enough and the Skilligent software can learn how to use it directly, for example via learning a simple control law (skill), the payload can be integrated with the software via the *Device Abstraction API*
- If the payload is complex to learn how to operate, it is necessary to build "hardcoded skills" which help the control software use the payload (look for the module called *"A library of hardcoded behaviors"* in Figure 7). The skills can be built using C++ or learned dynamically in a simulation environment. The resulted skills (learned or hardcoded) which accompany a particular payload are called *skillsets*.

4.4 Integration with existing robotic software platforms

The Skilligent software can be split into components and integrated into existing control system frameworks or robotics software platforms[11]. For example, the computer vision, social interaction, behavior control, robot learning and other libraries can be used independently in an environment provided by an existing robotics control framework.

4.5 Integration with Microsoft Robotics Studio

Microsoft Robotics Studio (MSRS) is a platform and a development environment for rapid creation of robotics applications. Robotics Studio supports a growing number of robotics hardware platforms and provides a unified distributed computation environment for building robotics solutions.

Skilligent integrated the product with MSRS in order to provide the academia with a platform for research in the frontier areas of artificial intelligence such as robot learning, social human-to-machine interaction, speech emergence, visual servoing and others.

---

[10] The JAUS feature is under development. Please contact Skilligent for details

[11] Such as Microsoft Robotics Studio, open-source Player/Stage or proprietary platforms

5 About Skilligent

Skilligent develops and markets a trainable robotics control system with robot learning capabilities for applications in the service robotics, industrial robotics, and home robotics industries.

Skilligent LLC is headquartered in Dallas, Texas and operates an off-shore research and development center.

Please visit our web site for robot learning videos and photos:
http://www.skilligent.com/products/downloads.shtml

Interested parties, please contact us online:
http://www.skilligent.com/contacts/index.php