22 : __ __ __

A Structured Approach for Modular Design: A Plug and Play Middleware for Sensory Modules, Actuation Platforms, Task Descriptions and Implementations in Robotics and Automation Environments

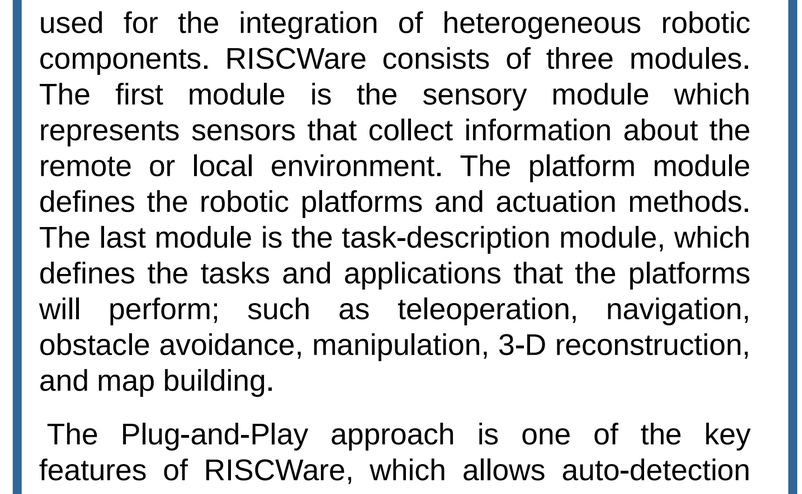
Ayssam Elkady and Tarek Sobh
Department of Computer Science and Engineering
University of Bridgeport, Bridgeport, CT

Abstract

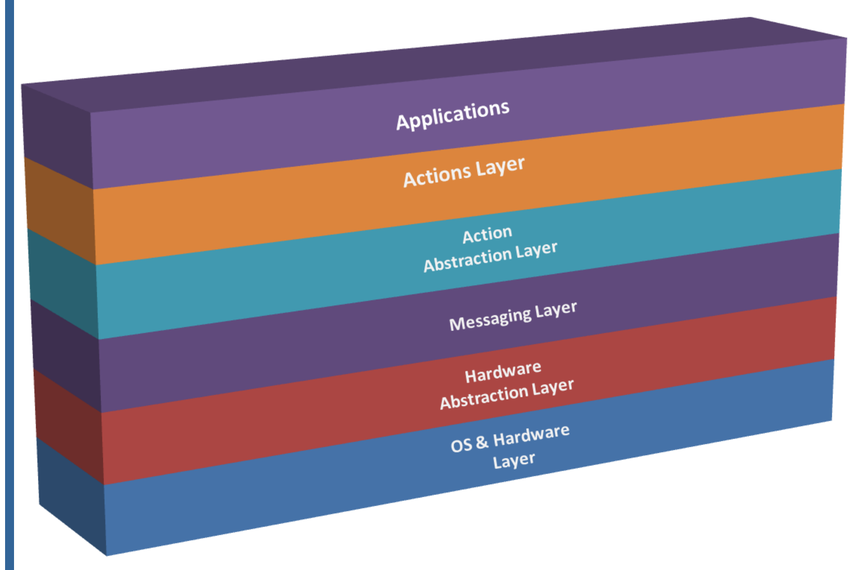
In this paper, the RISCWare framework is developed and described. RISCWare is a robotic middleware used for the integration of heterogeneous robotic components. RISCWare consists of three modules. The first module is the sensory module which represents sensors that collect information about the remote or local environment. The platform module defines the robotic platforms and actuation methods. The last module is the task-description module, which defines the tasks and applications that the platforms will perform; such as teleoperation, navigation, obstacle avoidance, manipulation, 3-D reconstruction, and map building.

The Plug-and-Play approach is one of the key features of RISCWare, which allows auto-detection and auto-reconfiguration of the attached standardized components (hardware and software) according to current system configurations. These components can be dynamically available or unavailable. Dynamic reconfiguration provides the facility to modify a system during its execution, and can be used to apply patches and updates, to implement adaptive systems, or to support third-party modules. This automatic detection and reconfiguration of devices and driver software makes it easier and more efficient for the end users to add and use new devices and software applications.

Several experiments, performed on the RISCbot II mobile manipulation platform are described and implemented to evaluate the RISCWare framework with respect to applicability and resource utilization.



Architecture of the *RISCWare*



• ***OS & Hardware Layer***: Includes the robot hardware system and also the operating system.

• ***Hardware Abstraction Layer***: A platform-dependent component of RISCWare used to hide the heterogeneity of lower hardware devices, provide the interface for the upper layers call and removes hardware and operating system dependencies between the robot and the application in order to assure portability of the architecture and application programs.

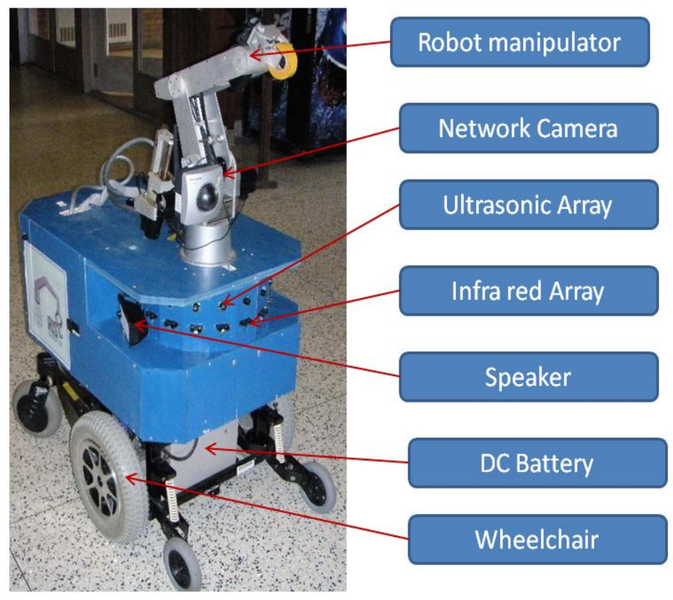
• ***Messaging Layer***: Used to allow communication between the devices and software applications.

• ***Actions Abstraction Layer*****:** Used to connect a software component to the *Messaging Layer* using the channel adapters, since the action module and the messaging system are two separate sets of software modules. The Channel adapter is a special code attached to a component to publish messages or consume messages from a message channel and trigger an action inside the component in response.

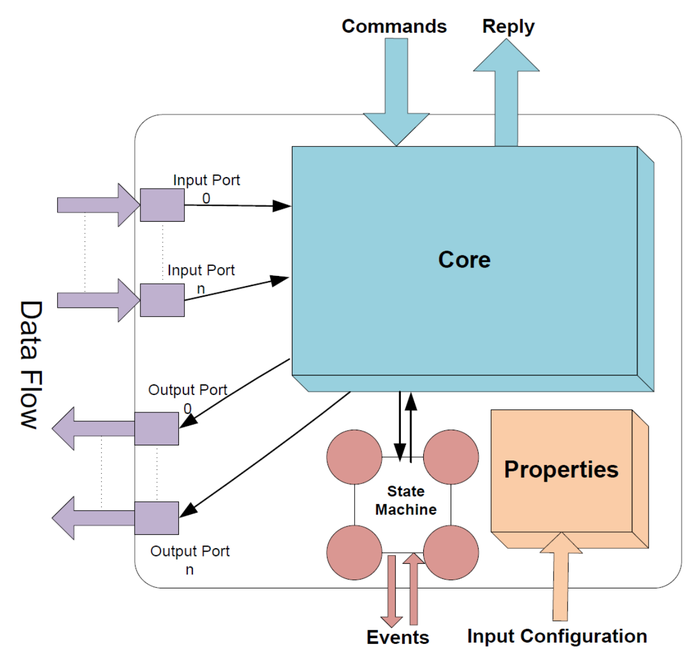
• ***Actions Layer*****:** Contains the tasks (actions),which are software modules used for sensing, decision-making, and autonomous actions; such as motion planning, vision, localization, tracking, motion control, etc. An action may communicate with the others using message channels.

• ***Applications Layer:*** The application layer is a set of tasks that work together. For example, the application template for motion control contains components for path planning, position control, obstacle avoidances and data reporting.

RISCbot II



RISCWare Component



A *RISCWare* component consists of the following objects:

•*Core*. The main process unit. It executes the commands, events and software script in the component.

• *Input/Output data stream ports*. Based on a publisher/subscriber model, an output port (publisher) sends data to all registered subscribers (input ports) through the message channels.

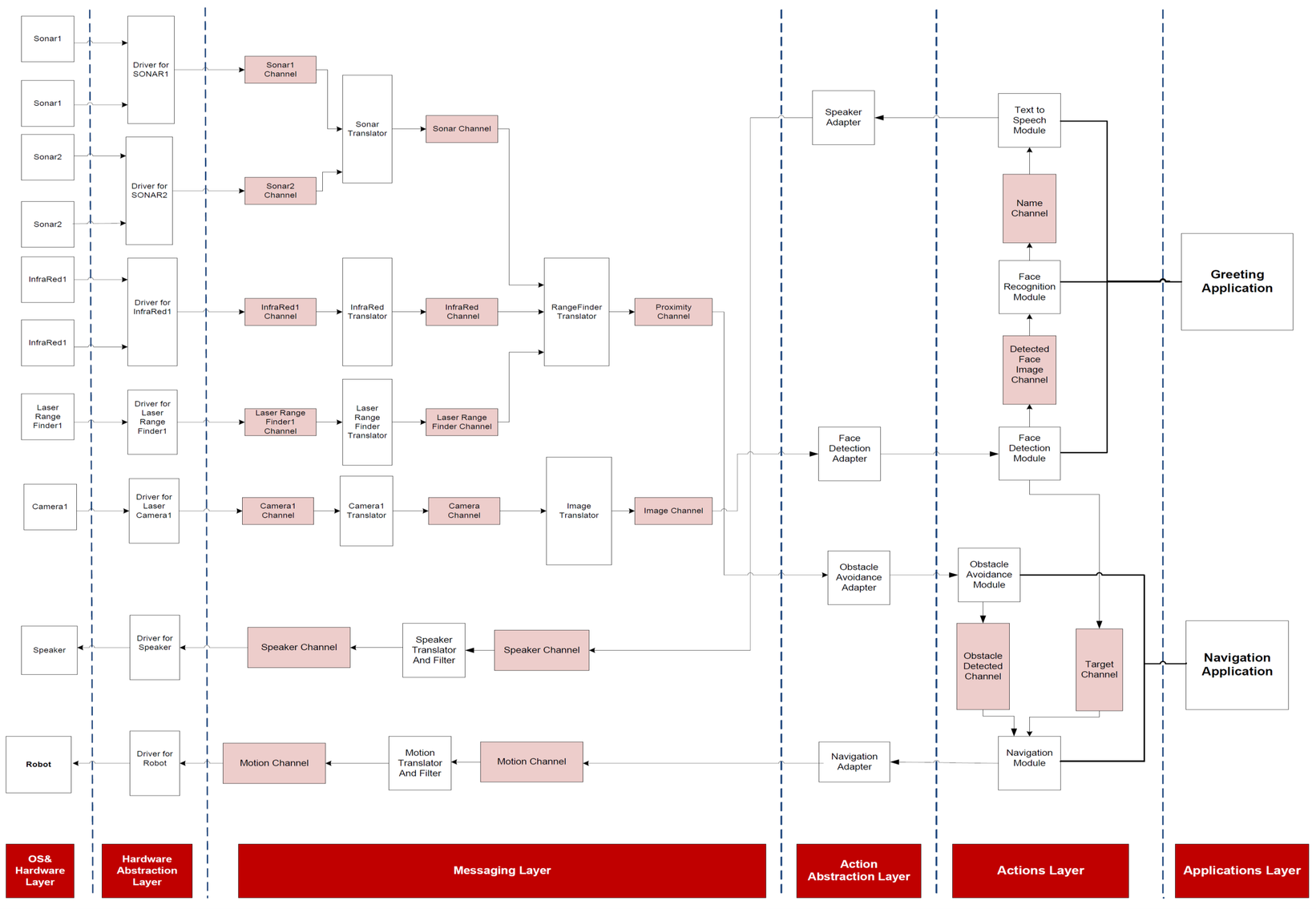
•*Properties*.

•*Finite State Machine (FSM).*

•*Event port*. The Event should be processed immediately to update the FSM of the component by changing the current state to the next state of the FSM based on the received Event.

•*Command port*. Used to receive the commands from the RISCWare middleware.

Overview of the *RISCWare* Middleware



Features

- Modularity.
- Device, Algorithm, and Platform independence.
- Hardware Architecture Abstraction.
- Scalable and Upgradeable.
- Ease of use.
- Real-time system.
- Plug and Play and Dynamic wiring.
- Security.
- Support for Parallelism and Multithreading.
- Robustness and Fault Detection.
- Resource Sharing.
- Asynchronous.
- Transparency.
- Open Source.
- Guaranteed delivery.
- Namespace; to support multi-robot control.