CLARAty: Coupled Layer Architecture for Robotic Autonomy

Daniel M. Gaines, Issa A. Nesnas

Jet Propulsion Laboratory,
California Institute of Technology

In Collaboration with

Ames Research Center
Carnegie Mellon University
University of Minnesota

ICAR 2005 – Navigation and Manipulation for Mars Rovers
July 17, 2005
Would like to support …

Custom Rovers

Manipulators

Reconfigurable Robots

COTS Systems
Problem and Approach

• Problem:
  – Difficult to share software across systems
  – Different hardware/software infrastructure
  – No standard protocols and APIs
  – No flexible code base of robotic capabilities

• Approach
  – Unified robotic framework
  – Capture and integrate legacy algorithms
  – Enable faster technology development
  – Operate heterogeneous robots
A Two-Layered Architecture

CLARAty = Coupled Layer Architecture for Robotic Autonomy

**THE DECISION LAYER:**
Declarative model-based Mission and system constraints Global planning

**INTERFACE:**
Access to various levels Commanding and updates

**THE FUNCTIONAL LAYER:**
Object-oriented abstractions Autonomous behavior Basic system functionality

Adaptation to a system
Adapting to a Rover

Decision Layer

Connector

Multi-level access Connector

Generic Functional Layer

Simulation Hardware Drivers

Rocky 8 Models/Heuristics

Rocky 8 Specialized Classes & Objects
The Functional Layer

- Navigation
- Path Planning
- Estimation
- Transforms
- Motion Control
- Input/Output
- Rover
- Behaviors
- Manipulation
- Vision
- Math
- Communication
- Hardware Drivers
- Simulation
- Locomotion
- Science
- Sensor

Adaptations
- Rocky 8
- FIDO
- K9
- Rocky 7

July 17, 2005
Architectural Traverse Example

- **Rover**
  - K9 Rover

- **Navigator**
  - Gestalt Navigator

- **Mapper**
  - Grid Mapper

- **Terrain Sensor**
  - Stereo Engine
    - Stereo Camera
      - Camera R
      - Camera L
  - Stereo Processor
    - JPL Stereo

- **Locomotor**
  - R8_Locomotor

- **Decision Layer**
  - Commanding and State Updates

- **Path Information**
  - Global Cost Func
  - Path Planner

- **Pose Estimator**
  - EKF Pose Estimator

Asynchronous:
- e.g. Rate Set at: 5 Hz
- e.g. Rate Set at: 8 Hz
- e.g. Rate Set at: 10 Hz
  - used by other activities

Synchronous/or Asynchronous:
- e.g. Rate Set at: 10 Hz

<active>
Architectural Traverse Example

- **Rover**
  - K9 Rover

- **Navigator**
  - R7/Soj Navigator
  - Asynchronous e.g. Rate Set at: 5 Hz

- **Mapper**
  - Obstacle Mapper
  - Asynchronous e.g. Rate Set at: 8 Hz

- **Terrain Sensor**
  - Stereo Engine
  - Stereo Camera
    - Camera R
    - Camera L
  - Synchronous/or Asynchronous e.g. Rate Set at: 10Hz used by other activities

- **Stereo Processor**
  - JPL Stereo

- **Locomotor**
  - ROAMS_Locomotor

- **Path Information**
  - Path Planner

- **Decision Layer**
  - Commanding and State Updates
  - Tangent Graph

- **Decision Layer**
  - Asynchronous
  - E.g. Rate Set at: 10Hz
  - Asynchronous e.g. Rate Set at: 5 Hz
  - Asynchronous e.g. Rate Set at: 8 Hz
  - Asynchronous e.g. Rate Set at: 10Hz

July 17, 2005

ICAR2005 - D.M.G 8
Locomotion

Rocky 7

Rocky 8
Reusable Wheeled Locomotion Algorithms

General flat terrain algorithms and specialized full DOF algorithms

(a) Skid Steering (no steering wheels)
(b) Tricycle (one steering wheel)
(c) Two-wheel steering
(d) Partially Steerable (e.g. Sojourner, Rocky 7)
(e) All wheel steering (e.g. MER, Rocky8, Fido, K9)
(f) Steerable Axle (e.g. Hyperion)
Goals of Estimation Architecture

- Facilitate development of estimators
  - Infrastructure common to estimators
  - Infrastructure common to Kalman Filters
  - Support for periodic computations

- Facilitate resuse of estimators
  - Treat estimators as virtual sensors
  - Common API enables swapping estimators
  - Adaptable to other CLARAty platforms
Generic Estimation

- Few assumptions made about how estimation is performed
- Actual estimators created by implementing interfaces
- Controllable Interface: acquire/translate control
- Measurable Interface: acquire/translate sensor reading
- Estimator Update Interface
  - Responsible for updating state and uncertainty
  - Uses data from Measurable or Controllable
  - Executed at user-specified frequency
Example: Estimating Temperature
Kalman Filter

- **System Model interface:**
  - Perform state transition
  - Provides system noise information

- **Sensor Model interface:**
  - Make sensor prediction
  - Provide sensor noise information

- Runs each model at user-specified frequency
Example: Estimating Voltage

1Adapted from “An Introduction to Kalman Filtering,” Welch & Bishop, UNC Chapel Hill, 2003.
Fido EKF Pose Estimator

- Initial implementation (Rocky 8)
  - About 3 months
  - Including estimation architecture
- Adaptation to Fido
  - 1 day
- Adaptation to Roams simulator
  - ~2.5 weeks
  - Including design/implement CLARAty/Roams interface

2Fido EKF designed by E. T. Baumgartner, H. Aghazarian, and A. Trebi-Ollenu (JPL)
Current Work - 6DOF Pose Estimator

6DOF EKF designed by Stergios Roumeliotis (University of Minnesota)
Comparing Pose Estimators

- **Comparison done to verify correctness and not to validate performance**

Pose Estimators:
- **Sojourner**: integrates z-axis gyro with wheel odometry (flat terrain)
- **FIDO EKF**: filters z-axis gyro bias & combines wheel odometry (flat terrain)
- **Sun sensor**: wheel odometry with sun sensor heading corrections
- **6DOF EKF**: *incomplete version* - 3-axis IMU with flat terrain kinematics
- **Wheel odometry**: integrated delta encoders
- **Visual Odometry**: uses hazard cameras to estimate ego-motion
- **Ground Truth** – measured using a total station at every interval

- Tested on four runs
  - 2 m straight line traverse over small rocks
  - 2 m straight line traverse over larger rocks
  - 2 m arc with 0.5 rad heading change over small rocks
  - 2 m arc with 0.5 rad heading change over large rocks

Heading relative to beginning of move
IMU mount not finely calibrated relative to rover frame
Pose Estimators – (a) 2 m straight; small rocks

![Diagram showing Pose Estimators](image-url)
Pose Estimators – (c) 2 m arc; small rocks

Arc 2m 0.5 radians, small rocks X, Y

Arc 2m 0.5 radians, small rocks Heading

Arc 2m 0.5 radians, small rocks Euclidean Error

Arc 2m 0.5 radians, small rocks Heading Error
Pose Estimators – (d) 2 m arc; large rocks
**Pose Estimators – Some Observations**

- Performance of all estimators except wheel odometry is comparable
- A gap exists between most pose estimators and ground truth – there is a significant potential for research to close that gap
- Occlusions from fixed mast impact sun sensor
Summary

- CLARAty provides a repository of reusable software components at various abstraction levels
- It attempts at capturing well-known robot technologies in a basic framework for researchers
- It publishes the behavior and interfaces of its components
- It allows researchers to integrate novel technologies at different levels of the architecture
- It will result from collaborative efforts of the robotics community
- It runs on multiple heterogeneous robots