JAUS

cimar

Center for Intelligent Machines and Robotics
University of Florida

OpenJAUS Meeting

Columbus, Ohio

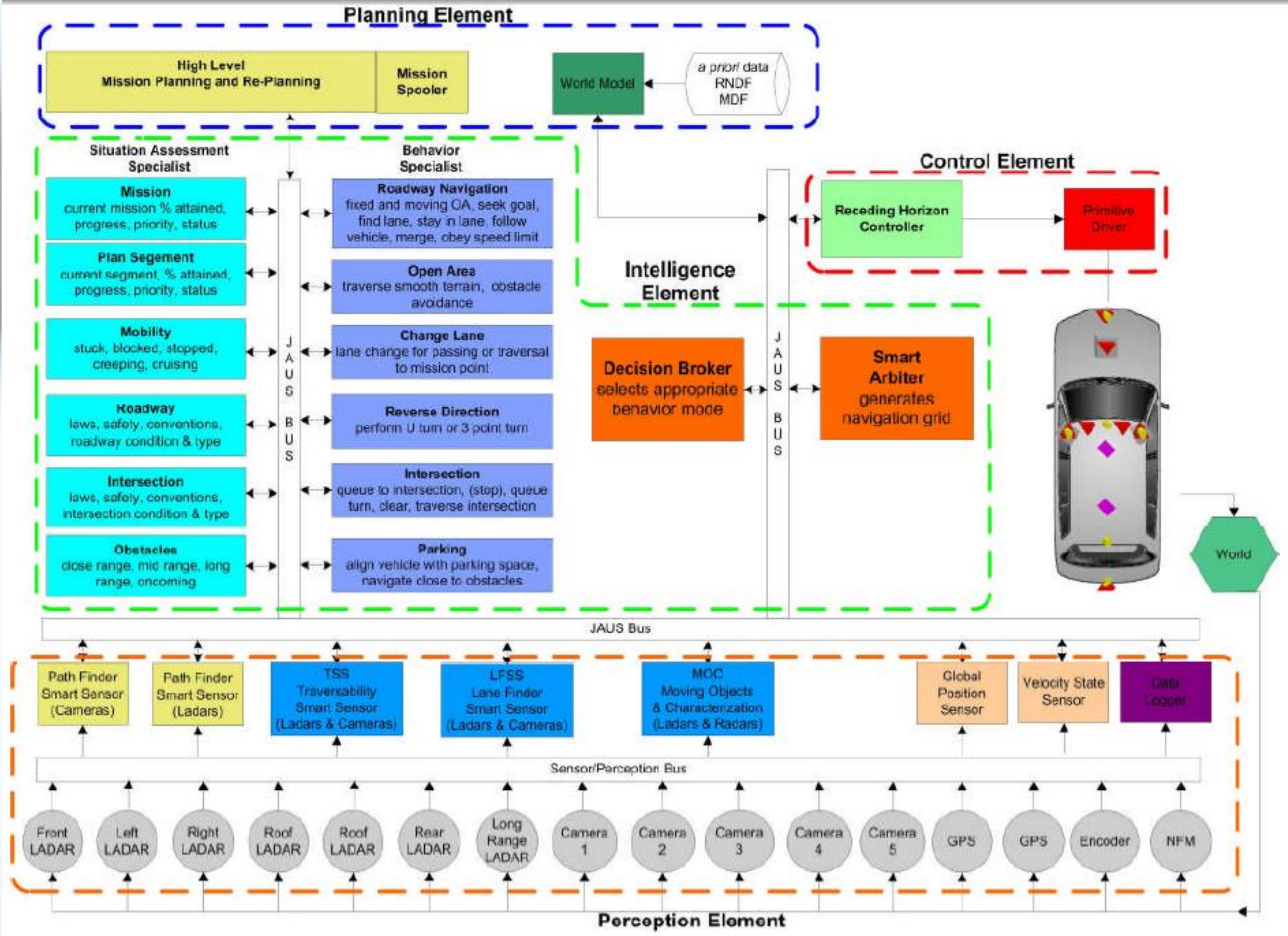
Prepared by: Gregory Garcia

CIMAR JAUS Implementation

- Before OpenJAUS
  - JAUS R.A. 3.2 Message Set
  - Proprietary set of libraries and component structure
  - JAVA Node Manager
  - System validation at DARPA Grand and Urban Challenges
  - Demonstrated interoperability at OPC experiments

Gator Nation Software Architecture

cimar
Center for Intelligent Machines and Robotics
University of Florida



Planning Element

High Level Mission Planning and Re-Planning

Mission Spooler

World Model

a priori data
RNDF
MDF

Situation Assessment Specialist

Mission
current mission % attained, progress, priority, status

Plan Segement
current segment, % attained, progress, priority, status

Mobility
stuck, blocked, stopped, creeping, cruising

Roadway
laws, safety, conventions, roadway condition & type

Intersection
laws, safety, conventions, intersection condition & type

Obstacles
close range, mid range, long range, oncoming

JAUS BUS

Behavior Specialist

Roadway Navigation
fixed and moving OA, seek goal, find lane, stay in lane, follow vehicle, merge, obey speed limit

Open Area
traverse smooth terrain, obstacle avoidance

Change Lane
lane change for passing or traversal to mission point

Reverse Direction
perform U turn or 3 point turn

Intersection
queue to intersection, (stop), queue turn, clear, traverse intersection

Parking
align vehicle with parking space, navigate close to obstacles

Intelligence Element

Decision Broker
selects appropriate behavior mode

JAUS BUS

Smart Arbiter
generates navigation grid

Control Element

Receding Horizon Controller

Primitive Driver

World

JAUS Bus

Path Finder Smart Sensor (Cameras)

Path Finder Smart Sensor (Ladars)

TSS
Traversability Smart Sensor
(Ladars & Cameras)

LFSS
Lane Finder Smart Sensor
(Ladars & Cameras)

MOC
Moving Objects & Characterization
(Ladars & Radars)

Global Position Sensor

Velocity State Sensor

Data Logger

Sensor/Perception Bus

Front LADAR

Left LADAR

Right LADAR

Roof LADAR

Roof LADAR

Rear LADAR

Long Range LADAR

Camera 1

Camera 2

Camera 3

Camera 4

Camera 5

GPS

GPS

Encoder

NFM

Perception Element

What we use from the JAUS R.A.

- Node Manager
  - Dynamic Registry, Message Routing, Service Connections
- Components
  - Command and Control
    - Subsystem commander
  - Platform Components
    - Global Pose Sensor
    - Velocity State Sensor
    - Primitive Driver
- Messages
  - Command Class Messages
    - Core Subgroup
      - Standby, Resume, SC (Create, Confirm, Terminate) Component Control (Request, Release, Confirm, Reject)
    - Platform Subgroup
      - Set Wrench Effort, Discrete Devices, Travel Speed
  - Query Class Messages
    - Core Subgroup
      - Query Component Authority, Component Status
  - Inform Class Messages
    - Core Subgroup
      - Report Component Authority, Component Status
    - Platform
      - Report Global Pose, Velocity State, Wrench Effort, Discrete Devices

R.A. Supplements

- Components
  - High Level Planner: Heuristic based Mission Planning/re-Planning based on Urban Challenge MDF and RNDF formats
  - Local World Model: Consumes Mission plan segments, Moving Object List, Lane offset corrections, outputs specialists findings as MetaData to SSC and grid map to planner
  - Roadway Navigator: Simultaneous Planning and Control, converts motion profile into wrench efforts to be sent to PD
  - Lane Finder: Lines or road edge detection with correction data
  - Moving Object Sensor: Identify and track objects of interest
  - Laser Smart Sensor: Laser Fusion and service provider, behavior based control of articulated sensors
- Messages
  - MetaData to support Adaptive Planning Framework
  - Moving Object List (multipoint object, velocity x-y, position x-y)
  - Lane Offset Correction
  - Report Road Network Message
  - Report Sensor Grid Map Message

Why OpenJAUS?

- Stay Current with Reference Architectures
- Mulitple OS Support
- Reduce software development time
- Facilitates compatibility between different vendors
- Extensive software debugging
- Focus on developing/enhancing vehicle performance and capabilities

How Open JAUS is it used?

- Replace CIMAR Java NodeManager with OpenJAUS NodeManager.
  - Software still built on proprietary libraries using templates
- Successfully Tested Full vehicle autonomous operation through OJNM
  - No noticeable performance anomalies

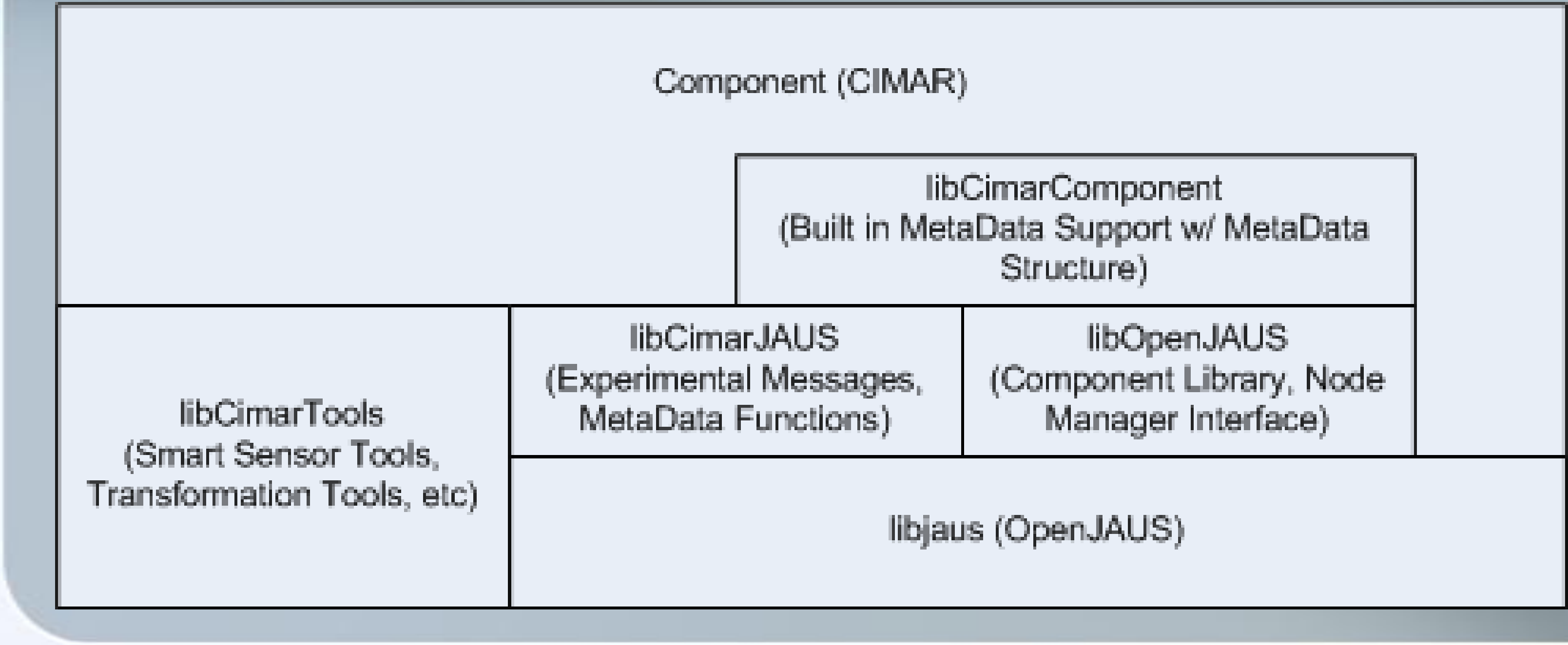
Message Traffic Through NM

- 19 Components across 9 nodes
  - 4 JAUS
  - 15 Experimental Components
- Service Connections: 57 total
  - 18-GPOS 30 bytes @ 50 Hz
  - 18-VSS 30 bytes @50 Hz
  - 7- Grid Map 14656 bytes @ 1-5 Hz
  - 5-Report cmpt Status 5 bytes @ 20 Hz
  - 2- Report Wrench Effort 20 bytes @40 Hz
  - 2 –Report Discrete Devices 5 bytes @ 10 Hz
  - 5 –Proprietary Msgs 2-97 and up bytes @5-40 Hz

Approx. Service Connection Bandwidth: 299630 bytes/sec or 2133 messages/sec

Future of OpenJAUS and CIMAR

- July/August '08:
  - Complete Software Migration Libraries to reference OJ 3.3.0 release
    - Supplemented with proprietary libraries as needed



Component (CIMAR)

libCimarComponent
(Built in MetaData Support w/ MetaData Structure)

libCimarTools
(Smart Sensor Tools, Transformation Tools, etc)

libCimarJAUS
(Experimental Messages, MetaData Functions)

libOpenJAUS
(Component Library, Node Manager Interface)

libjaus (OpenJAUS)

Future of OpenJAUS and CIMAR

- August/September ’08
  - High speed autonomous navigation with Obstacle Avoidance DEMO Using software architecture above
    - Upgrade Perception and Planning elements with maintaining current DUC capabilities