Interfaces for Integrating Cognitive Functions into Intelligent Vehicles

Matthias Goebl and Georg Färber  {goebl,faerber}@rcs.ei.tum.de

Cognitive Functions

- Intelligent behavior requires a large set of cognitive functions
- Challenges in the integration process:
  - Every function needs several software modules
  - Interdependencies between different modules
  - Software contributed by different (groups of) developers
  - Deployment to several vehicles
- Manageable interfaces essential for successful collaboration:
  - Transparent communication for easy debugging
  - Clear design for fast comprehension
  - Stability and easy handling provides maximum benefit

Interfaces for Integration

- Automotive industry: Busses (CAN, FlexRay, AUTOSAR VFB,...)
  - Transient information, limited bandwidth, serialization, static
- Robotics: Frameworks (OROCOS, Corba, Carmen, CLARAty,...)
  - Sizeable functionality, often component design according to programming paradigm, big libraries, intensive occupation
- Cognitive Automobiles: Lightweight fast data based approach
  - Acceptable to distinct research areas (AI, CV, EE, ME, ...)
  - Database with unified interface as central information hub
  - Publication of all available qualitative and quantitative data
  - Situation awareness gain from consistent information pool
- Real-Time Database for Cognitive Vehicles (KogMo-RTDB):
  - Used as outer integration framework
  - Easy specification of data structures in RTDB objects
  - Hard real-time capable, integrates also non real-time (GUIs,...)
  - API: insert, update, search, retrieve, wait for, delete object
  - Efficient implementation (update: 8.3μs retrieve: 4.6μs IPC: 29.6μs)
  - Utilizes “cluster-in-a-box” hardware platform ( multicore/HT)
  - Absolute timestamps to guarantee a consistent view
  - Data history preserved for specified time
  - Lock-less isolation between real-time and non-rt modules
  - Consistent view for knowledge processing with lower cycles
  - Sensor data association and interpolation

Simulation and Logging

- RTDB-Recorder logs all (selected) data and changes
  - Enhanced AVI (any size)
  - Videos play everywhere
  - Speed optimized (>40MB/s)
  - Precise timestamp (1 ns)
  - Tools to cut or filter logs
- RTDB-Player replays (selected) events into (another) RTDB
  - Offline analysis of (raw) data, timing, results, activity, runtime
  - Mounting position and configuration of sensors in database
  - No difference for connected modules in replay and simulation
    - Evaluation of different algorithms
    - Same GUIs and Tools usable

Integration and Results

- Successful application in several vehicles and at DARPA UC
  - Reference platform (close specification for binary compatibility)
  - Definition of modules sets for each individual vehicle
  - Configuration for road following with an active camera platform:
    - Video timestamp used to retrieve matching gaze direction from history
    - All camera parameters available
    - Video annotations in separate object, GUI shows overlay:
- Video annotations in separate object, GUI shows overlay:
- Real-time watchdog monitors and controls system via RTDB
- RTDB on embedded system (ARM,266MHz,32/8MB RAM/Flash)