Dependable Automotive Systems based on Model Certified Components

Tony Larsson[1], Walid Taha[1, 2] and Karl-Erik Årzen[3]
1) Halmstad University, Sweden
2) Rice University, Houston, USA
3) Lund University, Sweden

Advances in Intelligent Transport Systems (ITS), intelligent vehicles and cooperative systems are enabling traffic and transport solutions that are both safer and more environmentally acceptable. While it is well known that high dependability is a desirable feature it has a price, and the level of dependability needed varies from application to application. Typical examples are cooperative safety applications, in particular traffic situation aware vehicle driver warning and assisting systems, enabled by vehicle-to-vehicle and vehicle-to-infrastructure wireless communication in combination with the use of geographical map information, Differential Global Positioning System (DGPS) and vehicle-carried sensors. This kind of applications depend on the reception of satellite data (combined with on-board sensor data from, e.g. odometer and accelerometer) for positioning of the own vehicle and on periodic broadcasting of this information to all neighbors in range. Both the wireless communication with satellites and the one between vehicles can have severe difficulties, for example due to hills or high buildings directly hindering both kinds of radio transmission. There are techniques to make the solutions more robust, e.g. by information fusion for temporary or local communication outage compensation. However, further development is needed and since coverage problems can be made known in advance and related to geographical areas, such information can also be explored.

An important issue for all safety-critical applications is to be able to specify the robustness needs from the application perspective in a standardized way, easy to understand for developers. Developing dependable automotive ITS applications requires taking into account the hybrid (continuous/discrete) nature of these systems. A major challenge for building the systems is to consider CPS aspects at all levels of the design, where it makes sense. For example, at the most basic level such systems involve digital-to-analog (DA) and analog-to-digital (AD) conversion and related sampling/update to connect software and computer-controlled sensor and actuator data with the physical objects monitored and controlled. But there are similar problems showing up at higher-level vehicle position and heading estimation. CPS aspects also turn up in other forms, ranging from the dynamics at the vehicle and vehicle component levels to sensor and wireless communication disturbances at the overall system level. In a cooperative vehicle or transport system each vehicle is to be seen as an autonomous system. However, the vehicles are so far controlled by human drivers, ultimately responsible for the driving; each with different behavior that also varies over time. Still such systems must be able to cooperate following common rules, stated in laws but also influenced by local and personal habits and the actual praxis of the law. All these real world aspects have variability and one can thus speak about the normal behavior and then distinguish potentially risky anomalies or deviations from such an acceptable norm.

It is a challenge both to design and verify the dependability of a solution based on the cooperation between several embedded systems. This both within a specific vehicle and between different vehicles that cooperate via communication that under some environmental conditions can become stressed and unreliable, for example caused by human reaction induced delays or by communication disturbances due to physical “line-of-site” obstacles. It is important that the modeling method and related tools can make the involved designers aware of problems related to the analog/continuous physical real-world by help of suitable abstraction and visualization means. To cope with this there is a need for methodologies aimed for development and certification, especially functional safety, of dependable cooperative (network connected) embedded systems (incl. applications and components) as well as

supporting robust run-time architectures. Such architectures should give support for partitioning of concern and safe-guarding between different applications, components and needs. Application services provided must be able to run concurrently and will have need for different but also context-dependent priorities and priority handling techniques, e.g. considering potential consequences including their probability and severity. System level monitoring and policy handling components that can enforce controlled handover and/or failover to alternative backup solutions (made available in space or time). This kind of desired graceful failover must also be possible to model, simulate, implement and verify. To cope with behavior transitions and possible mode changes caused, for example by the many possible sources to sensor and communication disturbances it is a clear need for development methods that facilitate the modeling of such possible discontinuities and their effects.

Another issue, from a CPS point of view quite fundamental, is when it is a need to model the physical world in continuous/analog terms and when one can get by equally well or better by applying a discrete, sampling based, way of viewing and modeling the physical kinematic or dynamic phenomena, directly. An interesting complement to an initially discrete approach can then be to transform in the reverse direction and also be able to describe a discrete system in continuous terms and with analogous visualizations such as springs, masses, and dampers or more domain specific objects coupled in proper ways. The choice from a design process and method perspective probably depends on who you are and thus the mechanical engineer might prefer the continuous models and the computer scientist the discrete while a control system engineer can help to map between these two views.

Either way the development methods and tools needed should cover support for the modeling of:

- Vehicle kinematics and dynamics
- Sensor and communication disturbances causing jitter, delay or lost/degraded information
- Communication disturbances (in the air), e.g. fading, damping, reflexes, noise
- Driver behavior and human perception related time lags
- Traffic situation (incl. surrounding vehicles behavior and heading)
- Geographical view of road network and environment
- Safety margins in time and space
- Probabilities and distributions

The so far mentioned issues indicate a level of complexity telling that there is also an obvious need for component encapsulation and composition of many different (and partial) models. A productive and still safe development method shall thus enable both a divide and conquer approach and then, based on this, also a complementary compose/reuse approach.

Another very important issue related to component based design is to what extent traffic and safety risk situation awareness related models can be encapsulated in components and be dealt with in a modular and compositional way. Such components must also be possible to adapt in a flexible modular way to traffic laws and praxis applicable in different regions.

Today there are significant discrepancies between the models used for design and the ones used for simulation and testing. In the future, the key to bridging this gap will be to advance the state of the art in simulation models by extending them with new abstractions that meet the demands of four key concerns: First, these abstractions must be closer to the models used early on in the design process. They must also be well integrated otherwise this leads to a fragmented design process. It is natural for equation based modelling languages to serve as the least common denominator with a well defined semantics. To facilitate this, higher level tools (such as graphical or geometric tools) must define their

underlying analytical models or become replaced. Second, modelling abstractions must enable much higher flexibility. Today's tools generally support the modelling and simulation of systems that contain a fixed number of elements, with fixed structural connectivity. Third, because of the high computational cost of simulation and virtual experimentation, it is essential that these abstractions are naturally amenable to parallel execution so that emerging multi-core and many-core technologies can be utilized. Finally, there must be abstractions and tool-support that allow modelling of the interaction between different software components caused by resource-sharing, e.g., multiple threads sharing the same CPU or software components communicating over a shared communication link.

The development time needed and related cost of building high fidelity simulation codes for cyber physical systems is also a great challenge for a development process aimed at safety critical products and services. At the technical level, we also believe that modelling languages and formalism should support the broad spectrum of modelling technologies needed to describe cyber physical systems, including multi abstraction level modeling and mapping. As such, these languages and abstractions must support the description of:

- Conditional mode/model change and related scheduling
- Speed/distance/risk adaptive sampling and related behavior
- Disturbance/discontinuity handling (possibly by conditional mode/model change)
- Information fusion
- Probabilistic condition change or event handling
- Changes of environment
- Robustness and margins to changes in sampling period
- Adaptation to variants in design parameters
- Learning from gathered samples and adaptation to road and traffic situation

Tony Larsson, is professor of embedded systems at Halmstad University. After having received a Master in Mechanical Engineering 1974 from University of Linköping, Sweden, he worked in industry for more than 28 years mainly at Ericsson AB in the areas of real-time, dependable, network distributed, -embedded systems; both with hardware and software. During the period at Ericsson he was appointed technical expert in system design methods and also received his PhD in Computer Science at Linköping University, Sweden 1989. After a short period for the defense material administration in Sweden on system architecture for the network based defense he in 2003 became Professor of Embedded Systems at Halmstad University. His current interests are focused around software solutions supporting real-time cooperative network enabled embedded systems and especially for intelligent transport, vehicle and mobile sensor network applications relying on different forms of wireless communication. E-mail: tony.larsson@hh.se WWW: http://www2.hh.se/staff/tola

Walid Taho is a professor at Halmstad University and a part-time adjunct professor at Rice University. His current interest is in modeling, simulation, and verification of cyber physical systems. He was the principal investigator on a number of research awards and contracts from the National Science Foundation (NSF), Semi-conductor Research Consortium (SRC), and Texas Advanced Technology Program (ATP). He received an NSF CAREER award to develop Java Mint. He is the principle designer Java Mint (javamint.org), Acumen (acumen-language.org), MetaOCaml (metaocaml.org), and the Verilog Preprocessor. He chaired the 2009 IFIP Working Conference on Domain Specific Languages. He founded the ACM Conference on Generative Programming and Component Engineering (GPCE), the IFIP Working Group on Program Generation (WG 2.11), and the Middle Earth Programming Languages Seminar (MEPLS). E-mail: walid.taha@hh.se

Karl-Erik Årzen is a professor in Automatic Control at Lund University. His current research interests include embedded control systems, adaptivity in embedded systems, cyber-physical systems and modeling and simulation frameworks, in particular techniques that allow modeling of software systems interacting with physical systems. He is leading the activity on Adaptivity in embedded systems within the European Network of Excellence ArtistDesign on design of embedded systems. He is the program chair for the Euromicro Conference on Real-Time Systems (ECRTS) 2011. E-mail: karlerik@control.lth.se WWW: http://www.control.lth.se/user/karlerik/