Middleware and Service Assurance for Cyber-Physical Systems

Tom Bowen, Francesco Caruso, Brian Coan,
Balakrishnan Dasarathy, Bob Horgan and Josephine Micallef
Telcordia Technologies
One Telcordia Drive, Piscataway, NJ
Contact: Balakrishnan Dasarathy, Email: das@research.telcordia.com, Phone: +1.732.699.2430

Tom Bowen is a Chief Scientist at Telcordia Applied Research. His current interests include real time application-specific intrusion detection and reaction.

Francesco Caruso is a Director and Senior Scientist at Telcordia Applied Research. He specializes in system integration.

Dr. Brian Coan is a Director and Senior Scientist at Telcordia Applied Research. His current interests include algorithms to tolerate network and processor faults and database replication.

Dr. Balakrishnan "Das" Dasarathy is a Chief Scientist at Telcordia Applied Research. His current interests include real-time middleware and QoS for network-centric systems.

Dr. Bob Horgan is a Chief Scientist at Telcordia Applied Research. His current interests include software dependability, software testing, and software reliability.

Dr. Josephine Micallef is an Executive Director at Telcordia Applied Research. Her current interests include design and tools for large enterprise-critical, distributed applications.

Position Statement in a Nutshell

Our position is that middleware and middleware services for Cyber-Physical Systems (CPS) must support service assurance through techniques such as intrusion detection, proactive fault prevention, specifically software rejuvenation, and auditing. Certain end-user services to support multi-modal user interactions, such as user notifications in various forms (SMS, voice, email), and voice recognition also should be offered through middleware. Middleware model-integrated computing is yet another desirable feature. This position is based on our development experience in automotive telematics — a specific instance of a CPS.

CPS Domain of Interest

Telcordia is a large Telecom software vendor specializing in network operations and management support systems, and network services for both commercial and government markets. Lately, we have been developing, testing, and fielding a portfolio of CPS applications and services in the area of automotive telematics. The underlying middleware and middleware services for telematics and other CPS systems is of significant value as it will help us develop and deploy these applications in a cost- and time-effective manner.

Telematics is broadly the marriage of mobile communications and computers. Automotive telematics has come to mean the exchange of voice and data among vehicles, networked applications, drivers and call centers to support services such as real-time vehicle tracking, troubleshooting, emergency assistance and navigation. Two applications that Telcordia is currently developing and deploying are: (1) Telematics Service Provider Call Routing Service (TSPCRS) that routes a distress call through a wireless service provider's mobile switching center to the 911 network nearest to the vehicle that forwards the call to a human operator; and (2) Fleet Management Service (FMS) that includes diagnostics and prognostics, access to traffic and other information, navigation assistance, and emergency calls. Companies supporting

TSPCRS development and trials include a major North American vehicle security, communications, and diagnostics service provider, and two large wireless network service providers. A large North American automobile company is using Telcordia FMS as part of our national *go green* campaign to service their newly introduced hydrogen-powered fleet. We are also exploring the applications of our FMS offerings to various government agencies (e.g., Tank Automotive Research, Development and Engineering Center).

Middleware Infrastructure Needs

Telematics applications are multi-tiered. Sensors and actuators in vehicles and roadways constitute the rear-tier. The intermediate (data) tier provides collection, storage and processing of the sensed data. The front tier presents the data to a human operator through a portal providing multi-modal interactions (e.g., voice, streaming video) and report generation functionality. The Telcordia Telematics team is currently focusing on developing reliable end-to-end networking that is location- and content-aware over multiple vehicles and roadside units.

Middleware supporting telematics needs to provide the following over various mobile access and wireline backbone networking technologies: (1) typical communications services including asynchronous pub/sub messaging and synchronous messaging with real-time guarantees, and (2) standard distributed services such as naming and data replication. Streaming videos, speech recognition, speech-to-text, text-to-speech synthesis, and user notifications in various forms (SMS, voice, email) are some of the services/building blocks needed to support multi-modal interactions. These services are not typically known as middleware services, as they support mostly end-users. CPS applications with human-in-the-loop for control need to leverage these types of services in a run-time middleware.

A key problem we are often encountering is that it is not clear upfront which telematics application is both feasible and commercially viable. This implies a need for rapid and robust prototyping. Many proposed applications may never see the light of the day in a commercial sense. However, if an application gets acceptance, there will be an enormous market force and deadline pressure to evolve the prototype into a production quality system with 24/7 support. For instance, a version of Telcordia's FMS supporting a hydrogen fuel based fleet was prototyped using a scripting language. As the prototype gained commercial acceptance, it became evident that important QoS guarantees were missing, including production-grade high availability. Software development for CPS needs to address this dichotomy between prototyping and production needs. One possible approach is middleware model-integrated computing for robust prototyping by domain experts which is then replaced in stages by code written by software developers using a spiral development model that increasingly meets the various QoS needs including reliability, fault tolerance, real-time performance and security.

Verification and Assurance Features and Their Integration into Middleware

Three areas of focus in our assurance effort are: (1) Intrusion Detection, (2) Rejuvenation of Software, and (3) Auditing. All these areas are important to the automotive telematics applications domain as they involve safety and security of a large number of vehicles and people.

Intrusion Detection: Telcordia's position on intrusion detection is based on ten years of continuous NIST and DoD funded research into computer security resulting in several independently validated prototypes. We believe that despite the value of standard security techniques, such as authentication and encryption, automotive telematics system will contain latent vulnerabilities that attackers can exploit to compromise the confidentiality, integrity, and availability of automotive telematics services. Software vulnerabilities, such as those allowing buffer overflow and SQL injection attacks, have proven difficult to remove, but moreover, the possibility of insider or socially engineered attacks cannot be eliminated. Vulnerabilities will exist in all three tiers of the automotive telematics system, in both middleware and applications. To compensate, automotive telematics systems require intrusion detection capabilities including: (1) anomaly- and specification-based monitoring and control to detect and prevent improper behaviors, (2) data provenance techniques to properly attribute items in persistent storage, and (3) quarantine techniques that rapidly isolate, heal, and restore corrupted components. Middleware has a large role to play in all these three areas. Middleware needs to support intrusion detection by allowing interposition between modules. Interposition allows all aspects of every module's external behavior to be analyzed, and if need be, controlled. Middleware must also provide efficient communication services between intrusion detection components. A middleware Interface Definition Language can be used and extended to define standardized intrusion detection reporting and control messages and to simplify addition of new applications and address new threats.

Rejuvenation: Rejuvenation is a periodic, pre-emptive restart/reset of a running system to prevent faults such as memory leaks from becoming a cause for a catastrophic failure of the entire system or one of its critical components. These types of faults introduce decay and finally cause systems to crash. Rejuvenation is a powerful proactive failure prevention technique for any system, especially for a CPS, as it controls a physical world. Rejuvenation in its earliest form was employed to maintain high availability of electronic switching systems. The software for these systems was so complex that accumulation of fault states and progressive resource exhaustion caused excessive maintenance downtime and its associated unavailability. Careful software to manage selective restarts and resource renewal allowed those systems to be the first to achieve 99.9999% availability. Software rejuvenation, however, incurs overhead and should be done as infrequently as possible to minimize service interruptions. The research challenge is to transform rejuvenation from an engineering technique into a middleware service. For networked telematics applications, monitoring the health and availability of required resources across the network in middleware will allow control of system availability. Rejuvenation middleware can also be used to schedule and trigger repair of potential failures or renewal of resources before failures occur.

Auditing: Auditing is an automatic software analysis technique that monitors and, potentially, reacts to certain application conditions. Telcordia used auditing techniques to develop a logging service for an IP-based softswitch to detect and prevent various types of distributed denial of service attacks. Automotive telematics applications can involve several organizations accessing common information and sharing each other's resources. Auditing can orchestrate correct interactions among the potentially conflicting tasks in this scenario by monitoring system behavior and determining when pre-established policies are violated. An audit system can also track credential usage and data access for the purpose of post mortem analysis and forensics. Typical reusable components supporting an auditing service include: (1) libraries or APIs to efficiently log events with different, configurable levels of details, (2) repositories where logs are normalized, stored and archived, and (3) tools to analyze or mine log data for finding violations of system policies and the root causes of such violations. The first two can be done in middleware. Tools to analyze or mine audit data to identify and analyze policy violations or the causes of a rare and unexpected situations require application knowledge. How one leverages a common middleware to achieve these auditing functions is a critical research issue.