Edited by **Kevin Jost**

Managing for software success

The automotive industry is providing ever "smarter" vehicles. Intelligence in the car now does everything from ensuring a comfortable temperature zone independently for each occupant to enabling the creation of a whole new class of vehicles called hybrids. As embedded control functionality grows, managing the risks of the software development process requires robust management strategies.

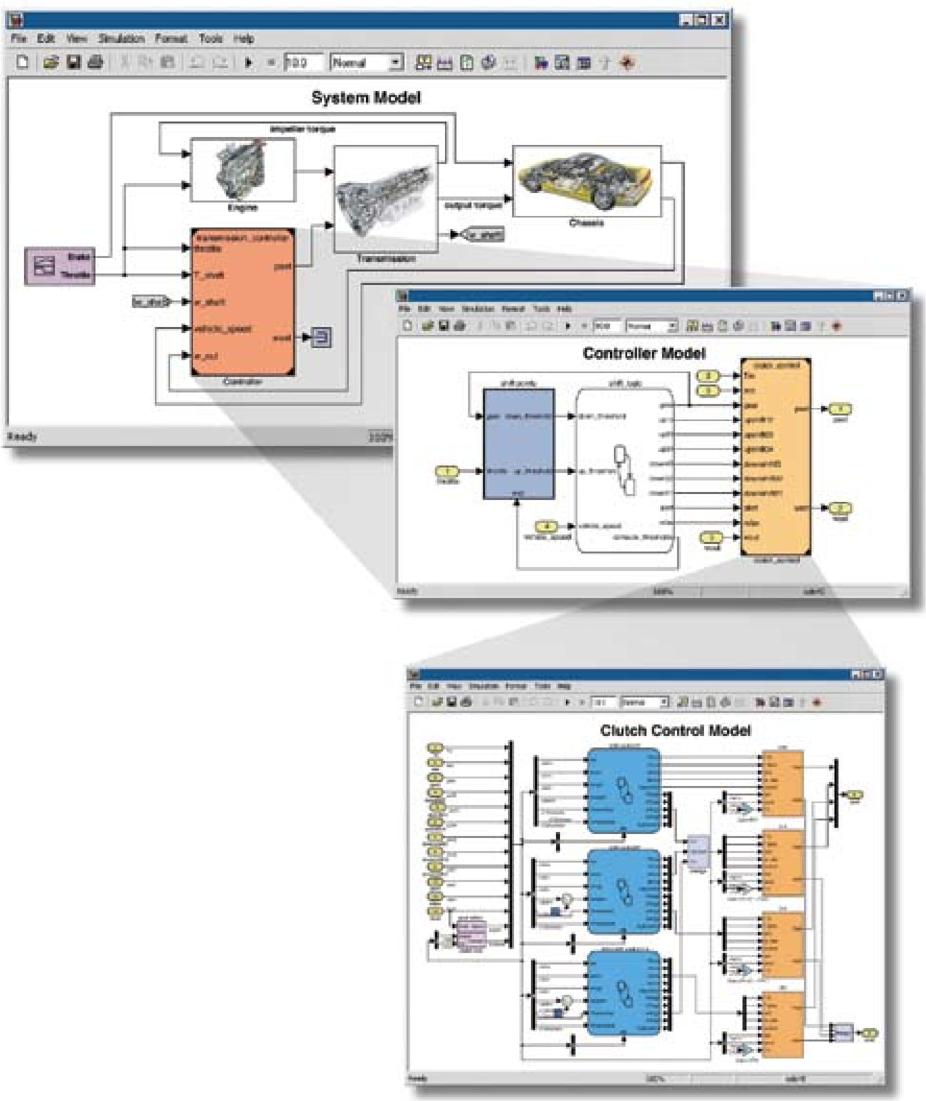
A lot of functionality means a lot of software. Current estimates for the software load in a vehicle range from 500,000 source lines of code (SLOC) for powertrain functions alone to over 1,000,000 SLOC if all functionality is included.

"The growth of the top end of automotive embedded software, as for the PC world, has generally been exponential," said Robert Gee, Director of Strategy for Systems and Software within **Motorola**'s Automotive Business. "Some automotive devices alone today may contain several million lines of code." Predictions have been offered that some high-end cars may have up to 100 million lines of code in them by 2015.

Not only is the software load growing in volume, it is becoming more integrated than ever. Functions that were once developed separately, such as powertrain and vehicle stability controls, now interact with each other. The trend toward standardized bus architectures instead of point-to-point wiring also increases the potential for software integration issues.

As the growth in functionality results in more code and integration points, meeting reliability targets becomes difficult. Unlike PC software users, automotive customers demand highly reliable systems. "Complete verification simply becomes impossible within reasonable schedules or cost due to the high number of combinations of system behavior," said Gee. This sentiment is echoed by Jim Kolhoff, Director of Software Engineering at **GM Powertrain**. "It is simply no longer possible to validate and verify functionality by brute force testing techniques alone," he said.

Various management techniques have been developed in the last decade to contain these risks. "At GM Powertrain, if we were still using the same practices of 10 years ago, we couldn't achieve nearly the same level of quality with the increased functionality we are providing," said Kolhoff. Some of these management

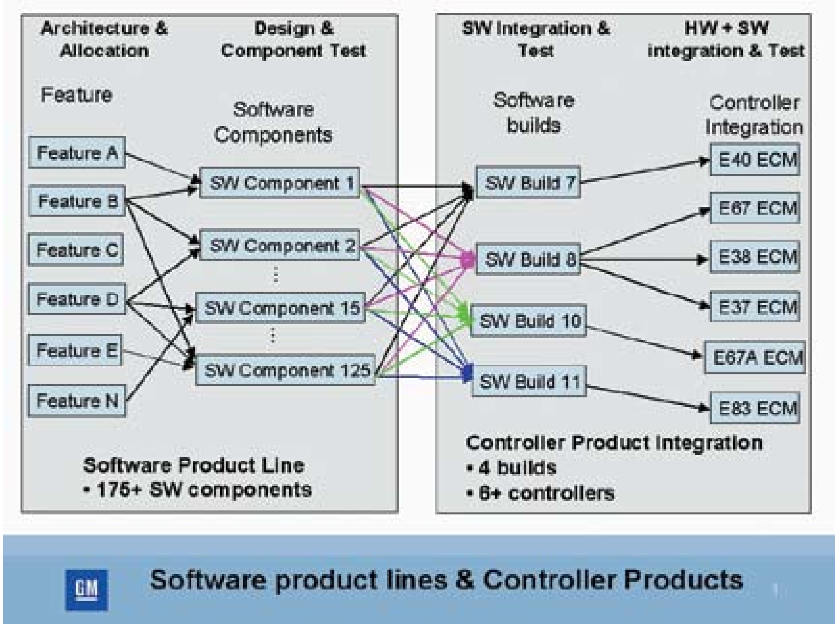


Model-based algorithm development uses graphical elements, such as block diagrams and finite state notations, to model system behavior. The advantage of this approach is that an algorithm design can be specified by a model that can then simulate and test algorithms, whereas written specifications can not. Shown here is one example, Simulink, a platform for model-based design available from The MathWorks.

trends include: 1) High-level architectural planning of products, including product planning several model years and generations into the future, 2) Model-based algorithm development, 3) Automatic generation of code from models, and 4) Industry-wide standardization of interfaces and common module functionality.

The approach that must be followed for success is "a well-conceived technical architecture designed for maximum reuse and extensibility," said Motorola's Gee. GM's Kolhoff also reports that GM Powertrain has evolved a management strategy of designing and deploying a family of control units that are matched against future GM Powertrains. The goal is to create a single software product line that then can be tweaked for the special needs of individual applications.

Once a product plan has been established to maximize reuse, there is still the challenge of producing reliable systems. How can the system be tested effectively without being tested completely? One



To reduce cost and risk, GM Powertrain is implementing an extensible software architecture composed of basic software features. This architecture gives engineers the flexibility to test and verify features prior to integration into the appropriate controllers defined by the vehicle electronic architecture.

answer is careful algorithm modeling before detailed software design or even hardware specification begins.

"The normal or traditional software development cycle involves lots of testing after coding," said Jorgen Hansson of the **Software Engineering Institute**. He advocates the increased use of modeling as a way of reducing the amount of testing required. GM's Kolhoff confirms that strategy. "GM Powertrain is aggressively implementing model-based algorithm development," he said. When a model-based system is simulated in software, the algorithm can be tested and validated prior to writing the software that will implement it.

An emerging technique that might lead to a better modeling environment is the development of the **SAE** Architecture Analysis and Design

briefs

STMicroelectronics is sampling a new digital AM/FM radio receiver chipset for use in automotive applications. Sophisticated digital signal processing techniques allow it to feature excellent reception quality while reducing interference even in the presence of challenging signal conditions such as weak field and strong multipath (reflection from mountains and buildings). Jointly developed by ST and **Bosch**'s subsidiary **Blaupunkt**, the advanced digital receiver integrates audio signal processing and radio data system decoding to satisfy the demands of today's mid- and high-end car radios for world-class quality, high performance, and system cost optimization.

TRW Automotive has unveiled an advanced pre-crash system combining its latest-generation adaptive cruise control (ACC) and active control retractor (ACR) seatbelt technologies to help prepare occupants before a crash occurs. The AC20 radar, which detects moving objects with an enhanced range of up to 200 m (660 ft), can identify target vehicles and how rapidly the acquiring vehicle is approaching them. If the system detects that the gap between the vehicles is closing too quickly, it signals the ACR to remove any seatbelt slack, which helps to better position the occupant for a potential crash and helps to warn the driver of a potential impending event.

SMSC has introduced a full line of MOST50 products that doubles the bandwidth available for automotive infotainment applications using the MOST (Media Oriented Systems Transport) standard from 25 to 50 Mbit/s. In addition to delivering this significant speed enhancement, the new product line offers transmission over an unshielded twisted pair of copper wires. This delivers greater flexibility to carmakers that design in-vehicle networks for more of a plug-in, component approach. The technology enables higher performance in-car multimedia systems and allows multi-channel DVD players, satellite receivers, digital A/V players, digital storage devices, telematics systems, and emerging rear-seat entertainment applications to operate over a single plastic optical fiber (POF) or an unshielded twisted pair of copper wires.

Language (AADL), advocated by Hansson.

Once an architecture has been defined and it is time to cut and test code, automated code generation can increase quality and reduce cost. "Software development to date has been handcrafted," said Liang Downey, the Automotive Industry Marketing Manager for **Wind River**. "This creates a lot of opportunity for hand-off errors when transitioning from architecture definition to design and design to coding to hardware bring-up." She advocates the use of the open-source Eclipse standard, model-based system design and autocode generation, comparing the approach to a CAD-based system for mechanical systems.

Future advances in cost reduction and quality improvement may come from industry-wide standardization of components and interfaces.

On the horizon is the evolving set of standards that is being developed by the **AUTOSAR** (Automotive Open System Architecture) consortium. Although many OEMs and Tier 1s may have, out of necessity, created their own set of standards, the AUTOSAR objective is to create a universal standard. This is accomplished by defining modularity, interface standards, and a standardized run-time environment. GM's Kolhoff compares it to everyone in the world running the Windows operating system on desktop PCs. "The question is, 'Will every control module in your car run the AUTOSAR basic software as its underlying operating system?'" said Kolhoff.

In May, the AUTOSAR development partnership published a set of the results of its first development phase. Further development is expected with the completion of its test and verification phase projected for this month, according to the AUTOSAR Web site.

Taming the risks as automotive control system software grows will continue to be a challenge to developers. The trends of using tools, such as model-based algorithm development and automatic code generation, coupled with well-thought-out product plans, are well established and should be considered by any not now using them.

Bruce Morey