Use of MathWorks Tool Chain for Proof-of-Concept Projects

Lev Vitkin, Delphi
Project Development Stages

1. Proof-of-Concept Project
2. Rapid-Prototyping
3. Production

**Output:**
- Product requirements
- Component’s behavioral descriptions
- Initial set of Data
- Initial set of Test Vectors
- Components and integrated models
- Final set of Data
- Final set of Test Vectors
- Source Code
- Documentation
Features of Proof-of-Concept Projects

- Parallel development of competing approaches
- Severe limitation on development time and financial budget
- Freedom in selection of the development tools
Project Description

- Display new concept of HMI graphics
- Select the color of in-cabin lighting (cluster and buttons) per user’s demand
- Use dual-view LCD display that alternates video streams to left and right viewing perspectives.
- Control driver and passenger side of dual-view display via “common” control buttons of Integrated Control Panel and Touch-screen panel
- Support common functionality like HVAC control, MP3/CD play list, DVD movie selection, Navigation System, USB media, etc.
- Disable driver’s access to view movie when car is in motion
Project Description
Tasks:
- Display multiple graphically-enhanced images
- Develop Driver/Passenger Distinction and Driver Peek algorithms
- Support CAN communication with ICP
- Support RS232 communication with Color Control box
Sensors Profiling and Allocation

Sensor individual and inter-dependency space-time profiling

Sensors reading with MathWorks toolboxes
IR sensors: Data Acquisition Toolbox
Image sensors: Image Acquisition Toolbox
Driver/Passenger Distinction and Driver Peek

- NI Data Acquisition Toolbox interface block from The MathWorks
- CAN interface block from Delphi (or Vector, Inc.)
- Altia interface block from Altia, Inc.
Software Implementation and Deployment

Requirements for the software development process in proof-of-concept projects are different from the production projects. The reduction of the development time has a higher priority compared to other software requirements such as high efficiency, maintainability or reusability.

- Use of Autocode generator
  - Driver/Passenger distinction and Driver Peek algorithms were generated with Real-Time Workshop® Embedded Coder.
  - The software was generated from the system model “as is” without spending much time on optimization of the generated code
- Hand alteration of the autocoded software
  - Use of “dummy” function calls “Call API” in the model as placeholders for the C-language interfaces. The function calls “Call API” were then replaced with the C-language Altia APIs, NI DAQ APIs, CAN APIs and Navigation System APIs
Factors that make MathWorks tools attractive for proof-of-concept projects:
- Extensive number of MathWorks blocksets
- Support of different hardware units for rapid-prototype development
- Availability of third-party plug-ins to MATLAB®, Simulink®, and Stateflow®
- Easy connection of legacy code via S-Functions
- Automatic code generation
- Formalization of requirements in the earliest stage of the project development cycle

Future improvement:
- User-friendly connectivity between Simulink and Stateflow and user’s tools via XML file format

The new concept of dual-view display based HMI was demonstrated at the 2007 Consumer Electronics Show in Las Vegas.