A RAPID PROTOTYPING ENVIRONMENT FOR THE DESIGN OF EXTENSIBLE IN-VEHICLE TELEMATICS SYSTEMS

D. Reilly and A. Taleb Bendiab
School of Computing and Mathematical Sciences, Liverpool John Moores University, Byrom Street, Liverpool, L3 3AF, UK.

d.reilly@livjm.ac.uk

Abstract

In view of the sharp uptake of mobile computing and communications technologies, many hardware and software vendors and telecommunication services providers [IBM & Motorola], [Sun Microsystems], [NEWSBYTES] have developed keen interests in the deployment of these technologies in vehicles as so-called In-Vehicle Telematics Systems (IVTS). Design is recognised as a vital discipline to envision innovative applications of new and existing technologies in new products and services that are fit for purpose, aesthetically appealing and delivered on time and to cost. Characterized by their component-based and hybrid systems nature – the rapid development of in-vehicle telematics systems requires a seamless and integrated design approach that supports multidisciplinary design teams. This paper considers the rapid prototyping process for the development of IVTS. The paper describes an Integrated Design Environment developed out of the FRETSET project [FRETSET] to assist in IVTS prototyping and demonstrates its application to an industrial case study.

1. Introduction

With the ever increasing application of information and communications technologies to serve the needs of the community and industry there is a growing demand for the deployment of these technologies in road vehicles as so-called In-Vehicle Telematics Systems (IVTS). IVTS can embrace a range of technologies including: computing, telecommunications, electronics, which need to be accommodated by a practical, yet integrated approach to modelling and design that integrates the contributions of different designers. The design approach should take account of the main attributes inherent in developing IVTS namely that they are an assembly of Commercial Of-The Shelf (COTS) telematics sub-systems.

Typically, IVTS combine both telecommunications and information systems, employing computer hardware, software and network technologies including; mobile and wireless communication system. IVTS applications can range from a simple GPS navigation system, right up to a complex network centric information system complete with a voice recognition system, which may be used to send and receive e-mail and text messages and access the Internet.

IVTS share a great deal in common with their office-based counterparts, but have several significant differences, namely:

- They contain some form of navigational sub-system.
They are inherently reliant on mobile communications technologies.

They are required to be unobtrusive to either the driver or passengers, and work-and-fail safe without affecting any of the vehicle safety-critical systems.

Current research in the field of in-vehicle telematics systems is concerned with several different themes including: the development of generic technology [FRETSET], [IVIS], devices [ANTMOD], [FATCAT], vehicle control and navigation [Robinson et al.], driver and passenger information systems [IVIS], [De Waard] and the study of human perceptions of in-vehicle telematics systems [IVIIS], [Bekiaris]. Based on an ongoing research effort into the requirements for an integrated framework for in-vehicle telematics systems development, [FRETSET], our work has focused on the development of approaches and tools to support the modelling and design of in-vehicle telematics systems. To this end, we have developed an Integrated Design Environment that can be used to prototype IVTS.

The paper is organized as follows: section 2 considers the rapid prototyping process and comments on the support required for the prototyping IVTS. Section 3 then provides a description of the Integrated Design Environment that was developed out of the FRETSET project to facilitate the rapid prototyping of IVTS. Section 4 considers an industrial case study, which demonstrates the application of the Integrated Design Environment for the development of an IVTS. Finally, section 5 concludes with a discussion of several related areas of research in which we have a vested interest.

2 Rapid Prototyping of In-Vehicle Telematics Systems

Rapid prototyping [Sommerville] is a technique used for validating systems requirements by way of experimenting with a prototype, which is refined and evaluated to improve the specification. As shown in Figure 1, the top half of the figure depicts the main prototyping stages that eventually result in a system specification. The prototype development/evaluation stages are iterative and loop until the prototype has been developed sufficiently to meet the requirements of the outline specification. The bottom half of the diagram shows how the results of the prototyping “feed” the design and implementation of a production quality system. Prototypes are often referred to as “throw-away” because it is intended that they are discarded once they have delivered sufficient results to serve as the basis for the development of the “real” system.

---

1 Further details can be found in [Sommerville] and [Pressman].
Figure 1: Prototyping Process

Figure 1 also shows a flow of components from the prototyping stage to the design and implementation of the production quality system and it is this use of components that makes prototyping suitable for IVTS modelling and design. Because prototyping is regarded as a “rapid” process (although this is not always so!) systems are assembled a system using mainly standard components (COTS) systems rather than built from scratch.

2.1 Rapid Prototyping Large Scale Systems

Rapid prototyping becomes more difficult when large complex systems are involved. As opposed to producing one large complex system prototype, it is better to decompose the problem into smaller more manageable sub-systems that can be prototyped separately and used as a means for communication between a multi-disciplinary design team. The development of several prototypes may use hardware and software resources from many different sources, which as remarked by [Bailey], may not be robust or “trustworthy”. Successful prototyping of large systems thus calls for techniques to manage different design resources and design configurations and techniques to assist communications between different designers.

The management of different design resources, design configurations and even different system prototypes can be achieved though an Integrated Design Environment (IDE) that provides tools and storage formats for use and storage of design resources respectively, where the design resources may emerge from different sources. The tools should support the rapid development of system prototypes from re-useable components and storage formats should represent prototype design resources and design configurations in a suitable fashion (e.g. hierarchical) for access by a multidisciplinary design team. Section 2.3 considers the requirements of an IDE that can be used to model and design IVTS and the IDE developed out of the FRETSET project is considered in greater detail in Section 3.
2.2 Support for a Rapid Prototyping Design Environment for In-Vehicle Telematics Systems Design

Based on the results of an industrial case study into IVTS systems development, this section summarises the main characteristics of IVTS systems and introduces the general requirements to support for rapid prototyping.

The main characteristics inherent in the IVTS design process are:

- IVTS are inherently component-based systems in the sense that a particular IVTS consists of a collection of off-the-shelf components. Similar to prototyping, they are assembled rather than built as an architectural design that reflects a certain configuration of components.

- IVTS contain a broad range of different technologies, which may include computer hardware, software, electronics and mechanical components. This suggests that the development of IVTS is likely to be a multidisciplinary activity, which draws on the collective effort of different designers.

- Much of the effort in IVTS design is expended in solving “a design configuration problem” that may be satisfied by several system variants (prototypes) that each have their own merits and limitations and the choice of the optimum configuration is similar to the evaluation stage used in prototyping.

- IVTS designs need to be flexible and extensible, such that they exhibit “plug and play” characteristics, which allow components to be added, removed or upgraded to meet specific end-user requirements or personalization.

These characteristics inherent in IVTS modelling and design share a good deal of common ground with the prototyping process and they can be used as the basis for proposing a design environment to support IVTS modelling and design, which should accommodate the following:

- Component catalogues that provide sufficient relevant design information in terms of both the basic design data such as physical dimensions and power consumption as well as derived or complex information or resources, such as CAD resources, analysis and simulation results.

- Support for rapid assembly/prototyping by providing easy access to components and complete sub-systems and rapid turnaround of different design configurations.

- A graphical means of representing the design of an IVTS, through a Graphical User Interface (GUI) that provides the physical layout of the IVTS as a “front-end” that enables different designers, from a multidisciplinary design team, to visualise the system’s architecture and configuration.

- Easy access to tools and resources, which different designers can use to layout and configure a system as a collection of standard components and explore, contrast and evaluate different configurations.

- Re-use facilities so that libraries or catalogues of re-usable sub-systems (even previous prototypes) can be developed that may be used in other IVTS designs.
Support for multidisciplinary design activities and teams by way of “ease of use” and flexibility so that the IDE can be used by a diversity of different designers, who may have different perspectives of the design including: computer scientists, electronic engineers, mechanical engineers, production engineers.

Based on the above requirements, an Integrated Development Environment (IDE) was developed, out of the FRETSET project that can be used to model and design IVTS following a rapid prototyping COTS approach. The is described further in section 3 below and it’s application to an industrial case study is considered in section 4. A more detailed treatment of the IDE is provided in a FRETSET project internal report [IDE].

3. The Integrated Design Environment

The IDE was implemented as a GUI-based application that offers catalogues of re-usable components together with tools and design resources to support the prototyping process. In view of the multidisciplinary nature of the IVTS design activity the IDE can be used to assist the designers by providing an integrated environment that offers uniform and easy access to existing and future tools and design resources. In line with [Fricks et al], the IDE provides a range of capabilities to designers without the need for learning many cumbersome interface languages and output formats. The IDE can also be used to carry out performance, reliability analysis and even simulations using the same toolkit. The graphical capabilities of the IDE represent the system’s design logically using block diagrams, which reflect the hierarchical system configuration and physically using media such as text, graphics, images, and even animations.

The IDE serves as the focal point for IVTS modelling, design and prototyping. Central to the IDE are its constituent modules and the Product Data Model (PDM), which was introduced to facilitate the interchange of information between: the modules within the IDE, component catalogues, external support tools and other FRETSET modules.

3.1 Design Management

The IDE GUI organization is based on the combination of component catalogues, a layout editor and a configuration manager. The component catalogues provides a “drag and drop” user interface and the layout editor is used to assemble the collection of components accordingly. The configuration manager uses a hierarchical representation to link different design sub-systems as shown in Figure 2, which shows the hierarchical “file-manager-like” representation (left-side) and the corresponding sub-systems of a CARIN navigation system, which is included in the current IVTS design (right-side). These sub-systems are represented as different design layers and these layers can be used to partition a large complex system into several sub-systems that can be prototyped separately.
The management or organization of designs relies on the PDM (considered in section 3.2) that specifies the structure of the design resources, from a conceptual view-point in terms of semantics and the context, which reflects the use of the resource and from a physical view-point as a storage format. This storage format is used both for storing components in component catalogues and for enabling information interchange throughout the IDE. The storage of information in accordance with the PDM facilitates it’s access and use by the various modules within the IDE, through their respective interfaces and by the user through a Graphical User Interface (GUI), which provide access to:

- IVTS ‘off-the-shelf’ components or devices.
- Previous project cases (completed systems) that are available for reuse.

3.2 IDE Architecture

Figure 3 provides a high-level architectural view of the IDE indicating the main functional modules, which are described briefly below along with the PDM, which serves to integrate these modules. The IDE also includes a Tool Interface Module (TIM) that provides an interface to external applications.
3.2.1 Product Data Model

The PDM provides a common format for the exchange of information within the IDE by representing the information on two fronts:

1. Structure of data
   - Specific data about each component – Name, part id.
   - Resources related to each component – CAD models, simulations.
   - About collections of components (sub-systems) – assemblies, relationships, configurations.

2. Meaning of data
   - Semantics / context.

The PDM was developed specifically to integrate the tools and design resources within the IDE and in the broader context of the FRETSET framework, to provide a data interchange format.

3.2.2 Design Catalogues

The design catalogues are essentially databases, which store product specifications according to the PDM through a variety of different media: ASCII data (from data sheets) images, CAD models, simulations and design resources to support the user in creating new configuration designs. The catalogues are used to store:

- Design information for existing components or devices that are being considered for selection in new configuration designs.
- Previous completed projects and sub-systems that will be used to support design re-use.

The conceptual representations of the structure of these design catalogues are listed below.
1. In-Vehicle Telematics Systems Components Data Structure
   a) Functionality
      i) Performance
      ii) Behaviour
      iii) Structure
   b) Interface
      i) System level (functional / behavioural constraints)
      ii) Physical level (Geometrical information)
          (1) CAD models
          (2) Dimensions
          (3) Fixing points
          (4) Morphological shape descriptors
   c) Detailed product Information
      i) CAD models
      ii) Costs
   d) Visual Representation
      i) ‘Captured’ media (photographic stills, video capture)
      ii) ‘Generated’ media (CAD solid models, VRML, animations)

2. Previous IVTS Project Cases (Built Systems) Format
   a) Project Design Specification
      i) Requirements
      ii) Costs
      iii) Risks
      iv) Safety issues
   b) System Configuration
      i) IVTS components (a link to a collection of records of format 1)
      ii) Network layout
      iii) Host vehicle
   c) Visual Representation
      iii) ‘Captured’ media (photographic stills, video capture)
      iv) ‘Generated’ media (CAD solid models, VRML, animations)

This structure is used to provide a hierarchical-type storage format, which is based on parent-child relationships, between components, that is used to represent system configurations and to facilitate the management and display of these configurations through the modules of the IDE. The design catalogues that store resources according to this structure, provide a centralized physical store of all the design information for use by the designers and other framework modules. For these design catalogues to be of use to the designer there is a need for a catalogue management system, which is considered below.

3.2.3 Design Catalogue Manager

The design catalogue manager provides a graphical user front-end to the design catalogues and allows the designer to access and manage the product specifications and images much like using traditional paper-based component catalogues. Taken together, the design catalogues and catalogue management system provide the designer with the following facilities:

- A wide range of different media to be stored and viewed, such as, CAD drawings, solid models, images (scanned drawings, photographs, etc.), textual descriptions, animations and
audio and video clips, performance specifications, and behavioural, functional and physical interface constraints.

- Functionality for “automated” component selection – the entire catalogue can be searched to find sub-systems or complete configuration designs that best match the specified requirements.
- Design resources, which can be directly used in the production of new configuration designs, for example a CAD drawing for a device could be ‘dragged-and-dropped’ from the catalogue into the configuration design.
- Tools for the population, maintenance and general management of design catalogues, such as the addition of products (components), removal of obsolete products and updating of existing products with new design information.

The design catalogue manager is illustrated in Figure 4 below, which shows a 3-d image of a GPS antenna.

![Figure 4: A GPS Design Catalogue Entry](image)

### 3.2.4 Model Editor

The model editor contains a drawing area that provides a layout editor to allow designers to graphically design and visualise the architecture of the IVTS using either a block-diagram (logical) representation or an “iconic” (physical) representation. The editor has the same “look and feel” as other popular drawing layout applications such as Visio.

### 3.2.5 Design Configuration Manager

This is a core module of the IDE, which allows the designers and view the structure of configuration designs -- what components or standalone sub-systems are used and how they are linked together. Essentially, the module consists of a design configuration “tree-structure”, which provides a hierarchical “file-manager-like” view of the systems being designed and integrated.
3.2.6 Tool Interface Module (TIM)

The TIM is an interface module of the IDE, which allows external applications to be interfaced to the IDE. The TIM accommodates different levels of integration depending upon the particular application to be integrated. This ranges from complete integration for those systems, which offer an open interface, to only partial integration, such as simply “launching” the application, in the case of the less flexible legacy applications.

The model editor, design configuration manager together with a design catalogue are illustrated in the Figures 5(a) – 5(d).

![Figure 5(a): Selection of an “off the shelf” CARIN Navigation System](image)

Figure 5(a) shows the design catalogue on the left and the model editor on the right after a CARIN navigation system has been selected from the catalogue and dropped onto the model editor.
Figure 5(b): A Complex IVTS Design

Figure 5(b) shows how the initial design of Figure 5(a) has progressed to represent a complex IVTS complete with several bus systems, CD-ROM drive and the CARIN sub-system of Figure 5(a). This system has been developed simply by selecting components from the catalogue, dropping them onto the model editor and using the configuration manager to configure the overall system.

Figure 5(c): Development of a Bus System

Figure 5(c) shows an alternative view of the design catalogue, which shows four bus system in “iconic” form that are available for use in the current design. The model editor on the right shows that a PCMIA bus has already been used to connect four peripheral devices.
Finally, Figure 5(d) shows how external tools can be added to the IDE, via the TIM, to perform a specific task. Above, we see the ANSYS package is being added, which will allow finite element analyses to be performed on design components.

4. Case Study

The IDE has been applied to develop a solution, to meet the requirements of a fleet haulage customer. The solution was arrived at by developing an initial prototype, using the IDE, which then served as the basis for the development of a production quality system. The study, which was conducted as a joint venture between our own research group and an industrial partner in the FRETSET project, is summarized below:

“A fleet haulage company requires an IVTS to provide a means of remotely monitoring the security and location of high-value goods being transported in trailer units. The proposed system will fulfil the following requirements:

- A facility to prevent the trailer unit from being tampered with, which will send a message from the trailer to alert both the control centre and driver.
- A facility to send signals from the control centre to the trailer in order to interrogate the system for its location and status, which are to be answered by a return message containing the required data.
A facility to monitor the status of the in-vehicle systems and inform the driver and control centre in the event of any system failure.

These requirements are represented in the sketch of Figure 6, to provide an informal outline specification of the system.

![Figure 6: Illustration of IVTS Requirements](image)

The components selected to meet these requirements were:
- GSM–GPS system to provide location and status data.
- Trailer door switch to detect any tampering.
- Trailer-truck link connector to connect the trailer module to the main truck / tractor module.
- Power supply.
- LCD display monitor.
- Cables, connectors and enclosures.

During the design and evaluation of the prototype, design catalogues of components were developed, which satisfied the requirements of each of the six component types. These catalogues were then used to select an appropriate combination of components, which taken together were compatible and offered the optimum design configuration. Obviously, this process of component selection is not carried out by a complete novice and domain knowledge does feature in the development of the system, for example, a design engineer is likely to know from experience, which particular micro-switch, from several, will best meet the needs of this particular system. However, our experience from this study revealed that it is extremely useful for the designer to be able to experiment freely and assemble a design prototype using an integrated, visual design aid, which provides a comprehensive catalogue of components and design resources.

The screen-shots of Figure 7 provide two examples as to how the design was developed and configured using the IDE.
5. Discussion

In this section, we discuss several additional areas for research that came to light during our work, after which we conclude with mention of our on-going research in IVTS and our future directions.

5.1 Conceptual Modelling

Up to now we have described an approach to IVTS design, which is based on using the IDE to develop a prototype of an IVTS. This approach has proved adequate from the point of view of the FRETSET project and the success of the case study provides evidence of it’s suitability. However, we feel that there is a need to instill a degree of formality to the initial conception of IVTS through the use of a semi-formalized requirements analysis and a conceptual modelling and design stage. As mentioned on several occasions, the design of IVTS embraces several different disciplines, each of which may have their own approach to conceptual modelling and design. Although these design approaches may have commonalties the best way forward is to adopt a higher level unified modelling system such as the Unified Modelling Language (UML) [Rational Rose]. The UML has received acceptance as the “de-facto” modelling system for object-based software systems [Booch et al] and is also receiving attention for modelling business systems [Hung], [Hung & Patel], [UML Business Modelling] and embedded and real-time systems [Douglas], [Gullekson & Selic], [Paltour & Lilius].

5.2 Extensible Hardware Architectures for IVTS

IVTS need to be flexible and extensible so that they can accommodate, with relative ease, revisions, upgrades and personalization. Thus there is a need to develop hardware and software architectures that can be used to provide “plug and play” capabilities. From the point of view of hardware
architectures, many of the main ingredients are already available, for example, many modern vehicles contain real-time CAN bus systems, which may be used for engine management systems, power steering and anti-lock braking systems. USB and PCMIA bus systems, both of which provide “plug and play” capabilities, can also be integrated into vehicles to provide a back-plane for navigation and information sub-systems such as GPS and vehicle tracking, so the vision of a flexible hardware architecture is not too distant.

5.3 Extensible Software Architectures for IVTS

However, the realization of an extensible software architecture is less clear. Advancement of software engineering has led to the availability of software components through COM, DCOM, Java Beans and CORBA. These components are being used in the development of office based software systems that are supported by a mature software architecture. However, where IVTS are concerned, there is no mature equivalent available to support software components. Furthermore the range of devices that may be used in-vehicles (embedded systems, palm-tops, lap-tops) leads to a proliferation of different operating systems and protocols (EPOC, WAP, Bluetooth) which complicate the development of a flexible “plug and play” software architecture.

5.4 Future Work

We have been active in two of the above areas and are currently engaged in studying the third, in particular: Work has already been carried out to assess the suitability of the UML to conceptually model IVTS requirements and designs [Semi-Formal Modelling]. Previous work has resulted in the specification of an extensible hardware architecture for IVTS [Hardware Architecture]. Finally, the specification of an extensible software architectures is a major cause for concern in our next follow on project to commence in January 2001.

6. Acknowledgements

The UK EPSRC (GR/M02958) under the Foresight vehicle initiative finances this work. Thanks are also due to our industrial partners for providing the described case-study material.

7. References

[ANTMOD] “Multiband Integrated ANTenna MODules for Vehicle Telematic Systems”, A Project Conducted as part of the UK Department of Trade and Industry’s (DTI) Foresight Vehicle Programme, University of Kent, Harada Industries Europe, Rover Group, Nissan Motors UK.


[FATCAT] “Future Antenna Technology for Cars and Trucks”, A Project Conducted as part of the UK Department of Trade and Industry’s (DTI) Foresight Vehicle Programme, Marconi Research Centre, University of Birmingham and BSH Industries Ltd., UK.


[IVIS] “In-vehicle Information Integration Study”, Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, TN 37831-6364, USA.

[IVIS] “In-vehicle Information System Study”, Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, TN 37831-6364, USA.


