

Safe Design Methodology for an Intelligent Cruise Control System with GPS

Ouassila Labbani, Éric Rutten and Jean-Luc Dekeyser
University of Lille, LIFL/INRIA Futurs/Dart/West
Email: labbani, rutten, dekeyser@lifl.fr Url: www.lifl.fr/west

Abstract—In this paper we study the application of a safe design methodology in the case of an automotive system. This methodology is based on a clear separation between control and data parts. It allows to facilitate the specification and to have a better readability. We present the advantages of this methodology on a GPS cruise control system.

Keywords— Intelligent Cruise Control with GPS, Localisation-based Design, Control/Data Flow Separation, Safe Design.

I. INTRODUCTION

According to several statistics, high-speed and disrespect of speed limit cause 40% of fatal accidents and increase their severity. In spite of this, more than 60% of drivers do not comply with speed limits on urban roads or trunk roads. The worst situation is when a trunk road passes through a village. There, almost 80% of drivers break the speed limit. In order to give to drivers the means for controlling the speed of their cars, several constructors have developed various systems such as the speed limiter or regulator already present in some cars. In this field, research continues to give more effective systems, like speed regulator systems with GPS (Global Positioning System) which allows to adapt the speed of the car to that authorized in its zone of localization.

The safety-criticality of these systems requires reliable and efficient tools and methods for their safe design. Failures and bugs in these systems can lead to data or time losses, incidents that can potentially be catastrophic. Modeling and design of these systems is therefore a difficult and important activity.

One of the principal characteristics of these systems is that they combine control and data processing. Several approaches have been proposed to specify this kind of systems. However, existing approaches do not follow any clear separation design methodology and lead to a mixture of control and data flow representation. This mixture can make difficult the understanding of the system and the re-use of existing applications. To fill this gap, we have proposed a new design methodology [1] based on the running mode concepts [2]. It is based on the precise and clear separation between control and data flow parts to facilitate separated study of the system.

In this paper, we apply our control/data flow separation methodology to an Intelligent Cruise Control with GPS system (ICCG). We present the advantages of this methodology in the case of a real system such as the modular development, the readability, and the safe design.

II. CONTEXT

A. Intelligent Cruise Control Systems with GPS

Many European countries have launched different projects and experiments on speed regulation systems, denominated for the majority ISA (Intelligent Speed Adaptation) or EVSC (External Vehicle Speed Control). The obtained results are generally diversified and the comparisons is difficult since each experiment has specific objectives and protocols. Thus, technologies used and the nature of systems vary from one project to another. However, some common conclusions indicate the possible benefit of such systems to reduce the number of accidents and their dangerousity.

Among these projects, we can find the french project LAVIA [3] (www.lavia.fr). The LAVIA system is an intelligent regulator which automatically adapts the car speed to the speed limit according to the car position. This position is determined using a navigation device which combines dead reckoning data with the GPS ones to assess the car position and then matches them with a digital map in order to obtain accurate car localisation coordinates. The Swedish National Road Administration conducted a road information project involving ISA in urban areas [4] (www.vv.se/isa). Its aim was to learn more about drivers attitudes and how they use the system, the impact on road safety and the environment, and the integration of the system on the cars.

Another European project, PROSPER [5] (www.prosper-eu.nl) explicitly identifies that advanced assisted driving technology and technology relating to speed limitation devices can be an important measure. An ISA trial took place in the city of Ghent, Belgium [6]. ISA-Ghent project progressed in parallel with PROSPER project but it is not a part of it in spite of their common objectives. The tested ISA device is the Limit Advisor M2002 developed by the Swedish companies.

Another similar project is the INFATI Denmark project [7] (www.infati.dk) mainly bound to the study of the techniques and the means for the realization of the system, system specification, prototype development, tests and informations.

EWGOSC, an European Work Group On Speed Control, groups researchers implied in ISA experiments annually.

B. Design Methodologies for Hybrid Systems

The most realistic embedded systems, like speed limiter or regulator systems, combine control and data processing (*hybrid* systems). To specify these systems, several approaches have been proposed mixing control and data representations.

Generally, in the system design field, developers use safe specification languages to describe their systems. These languages are based on different models and can be classified into two main families: *declarative* (data oriented) and *imperative* (control oriented) languages. In this field, synchronous languages like Lustre, Signal and Esterel [8] are often used for the specification of embedded systems. These languages use formal techniques allowing to define efficiently a set of tools for modeling, simulating and verifying critical systems.

Synchronous languages are also classified into declarative and imperative languages. The design of hybrid systems needs the development of approaches using the two types of languages. Among these approaches we can find the *multi-languages* approach which combines imperative and declarative languages. It is based on a linking mechanism and allows the re-use of existing code. However, when using several languages it is very difficult to ensure that the set of corresponding generated codes will satisfy the global specification.

Based on these approaches, several model-based development environments are available and widely used in the automotive industry, like for example Matlab Simulink/Stateflow (www.mathworks.com), Scade (www.esterel-technologies.com) and UML (www.omg.org). These works allow the control and data processing combination. However, all these studies do not follow a clear separation design methodology and lead to a mixture of control and data flow representation. This mixture can make difficult the understanding and the study of the system. To fill this gap, we have proposed a new design methodology [1], using Scade and the Mode-Automata concepts, which allows to have a clear separation between control and data flow parts.

III. ICCG SYSTEM DESCRIPTION

We propose to study an ISA system that we call ICCG (Intelligent Cruise Control with GPS). Its main role consists in limiting the car speed automatically to the local prescribed speed given by the driver or by the GPS. The ICCG can be seen as an electronic help which facilitates the control of a car. It informs the driver about the various changes of speed limits and, in some cases, obliges him to respect them.

The studied system can operate in five different modes: Alarm, Limit, Cruise, LimitGPS and CruiseGPS. The interaction with the system and the activation or deactivation of different modes are done through a set of buttons: On, Off, Set, Resume, QuickAccel, QuickDecel, GPS and Cruise (More details on the functioning of this system can be found in [9]). Informations can be displayed on a dashboard to inform the driver about the activated mode, the speed limit and the current speed of the car as shown by figure 1.

Globally and independently of running modes, the system can be in one of the four states: SysOn (activated), SysOff (deactivated), SysSTDB (suspended), and GPSFail (GPS signal lost). The switch between states of the system is done according to the brake pedal, the accelerator pedal, and the GPS signal (figure 2). Thus, in this system and for safety



Fig. 1. System representation

reasons, we define an interval speed (Between) at which the system can be activated. Outside this interval, the system is systematically deactivated and switches to SysSTDB state. For example, if the current speed exceeds 160km/h or it is lower than 30km/h, we consider that the intervention of the system is not important and the driver has total control over the car.

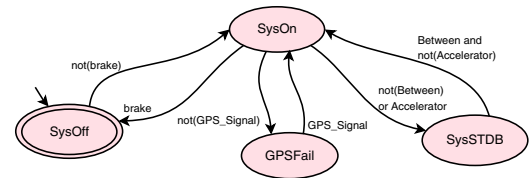


Fig. 2. Different states of the system

Initially, in SysOn state, the system is in Alarm mode. The switch between the various modes is given by figure 3.

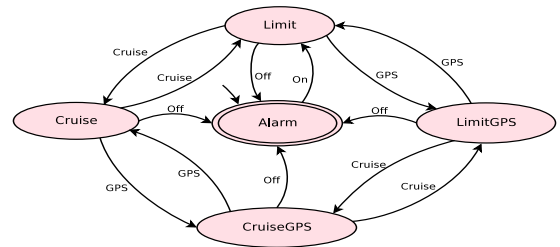


Fig. 3. Different modes of the system

The Alarm mode is only an informative mode. It indicates the driver, by an audio or luminous signal, if the allowed speed limit given by a GPS is exceeded. In Limit mode the system does not allow to exceed the speed limit fixed by the driver who can always control the car by accelerating or braking. The Cruise mode maintains the car at a constant speed given by the driver who does not control the speed of the car via the accelerator pedal but rather by using a set of buttons. The LimitGPS mode is similar to Limit mode, where the speed limit is the minimum of the speed required by the driver and that given by a GPS. The CruiseGPS mode has the same behavior as the Cruise mode, where the speed limit is calculated in the same way as in LimitGPS mode.

IV. CONTROL/DATA FLOW SEPARATION METHODOLOGY

We present our design methodology using the Scade (Safety Critical Application Development Environment) development environment commercialized by Esterel Technologies..

A. Control/Data Flow Combination Disadvantage

To illustrate the control/data flow combination disadvantage, we propose to study the example of a generic task pattern using Scade tool. In this example, the system can be in three different states: H, M and L as shown by figure 4. These states are differentiated by some characteristics such as time cost and quality. We can think of applications such that we have: H (highest quality and time), M (medium) and L (low). These three modes can be switched between according to the transitions and their conditions c_k . An example is a task computing at each cycle an expression summing three terms: $E = E_1 + E_2 + E_3$, where E_3 and E_2 can be approximated by 0. Each of the modes corresponds to: H: the full sum, M: an approximation $E_1 + E_2$, L: a degraded version E_1 .

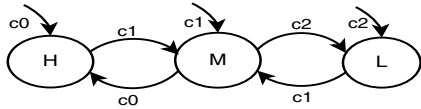


Fig. 4. Simple example of task functioning

This example contains both control and data processing and can be specified using Scade tool. A possible solution for this system is given by figure 5, inspired by that proposed by Esterel Technologies to specify the Climate example [1].

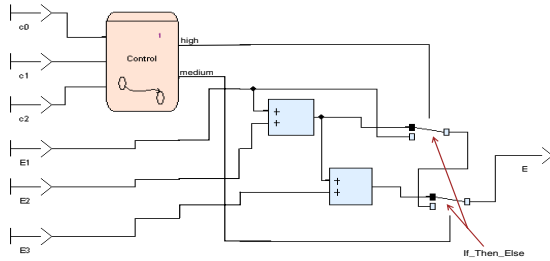


Fig. 5. System specification in Scade

The system possesses three inputs relative to conditions: c_0 , c_1 and c_2 , and three inputs relative to parameters E_1 , E_2 and E_3 . As output, it provides the result E . Input values c_0 , c_1 and c_2 pass through a control part represented by the SSM (Safe State Machine) *Control*. This SSM allows to activate the calculation parts by two different signals: *high* and *medium* which correspond to the activation of state functions H and M respectively. If the two signals *high* and *medium* are false, the system is in L state. We notice that the model contains a mixture of calculation and control. This mixture can make difficult the understanding of the system, as well as the use of already existing tools, dedicated exclusively to processing the calculation part or the control part. Furthermore, if we want to modify the behavior of one of the system states, it is difficult to locate the concerned part of the system. To avoid these problems, we propose a design methodology based on a clear separation between control and calculation parts.

B. Control/Data Flow Separation Concept

The example represents a simple case of systems whose behavior is mainly regular but can switch instantaneously from a behavior to another. In this case, the global system is usually composed of a high level control oriented sub-system which executes different data processing for each state of the system. For these systems, it can be important to study separately control and data parts which gives a more structural view of the model and facilitates the modification and the re-use of different parts. We have proposed a new design methodology [1] allowing to have a clear separation between control and data flow parts. The idea consists in using the concept of running modes to facilitate the specification of the mainly regular systems and to give a more readable design.

The generic task pattern example, introduced in section IV-A, can be easily specified using multi-mode concept and our design methodology as shown by figure 6. To make this,

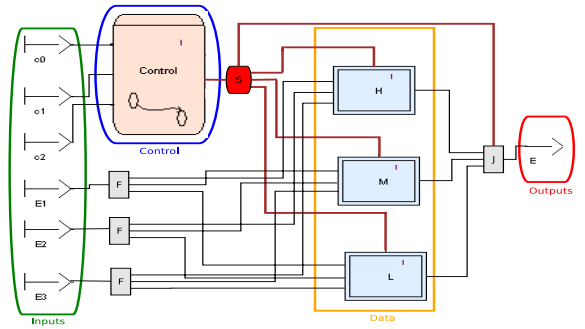


Fig. 6. Control/data flow separation model

we have divided the problem into three sub-problems that correspond to the functionality of the different state modes: H , M and L . The activation of each part is made by the SSM *Control* depending on the input values of c_k conditions.

In this approach, we can clearly distinguish inputs and outputs of the system, control parts, and data parts. Contrary to what its name indicates, the data part does not only designate an exclusive data processing. It can also contain a SSM followed by a data part, or only the control part. The lowest level in the hierarchy represents an homogeneous part that can exclusively contain the control or the elementary calculation.

C. Benefits of the Control/Data Flow Separation

The introduction of a design methodology separating clearly control and data flow parts allows to have a more readable model. Moreover, a modular specification of the different parts of the system allows to benefit from the modular development. It facilitates the re-use of existing applications, the modification, the introduction and the deletion of modes.

This technique allows to simulate and verify separately the different parts of the system, and consequently have a considerable gain in verification time and memory capacities since the number of states of the verified module is much smaller than that of the complete system. The automaton structure is also exclusive, and to each state of the automaton is

associated only one activity. The different activities are then exclusive and can be studied separately by using the most appropriate tools for each part. This methodology facilitates also the localisation of the different errors while avoiding the modification of the whole application.

V. APPLICATION TO THE ICCG SYSTEM

The specification of the ICCG system consists of two essential parts: ICCG represents the behavior of the cruise control with GPS, and `CarSimple` specifies the behavior of the car with which our system will interact [9].

The ICCG system takes as input a set of values representing the different buttons (On, Off,...), the pedals of the car, the speed limit of the current zone and that of the following zone. As output, the system provides an information on its state which can be active (`SysOn`), inactive (`SysOff`) or suspended (`SysSTDB`). It also provides an information about the selected mode, the fixed speed limit (`SpeedLimit`), the speed requested by the driver (`DriverSpeed`), the current speed of the car (`CurrentSpeed`), an alarm signal (`AlarmSignal`) to indicate that the speed limit was exceeded, the `StopAccelerator` signal to block the accelerator pedal, and the `GPS_Fail` signal indicating the loss of the GPS signal or the non-identification of the zone.

A first solution for the specification of the ICCG system has been proposed [9]. The corresponding design model was achieved in a very intuitive way since the main goal was just to develop a functional model without following any control/data flow separation methodology. It contains a mixture of control and data processing. It does not allow to clearly distinguish the various modes of the system and the switch conditions between modes. This model is very ambiguous since a small modification in the behaviour of a given mode requires the modification of the whole application. Indeed, it is difficult to extract the mode from the total specification. This is also valid for the introduction of a new mode or the deletion of an existing one. Thus, the application of formal verification techniques on such models is very difficult. Errors are more and more serious and the resulting system will be unstable.

For these reasons, we have modified the specification by adopting our methodology which allows a good separation between control and data flow parts. The specification diagram relating to ICCG system, following this methodology, is given by figure 7.

This model is composed of three main parts. The first part is a *pre-calculation* part represented by `Calculation` model which contain the common processing for the different modes. This calculation part is always executed independently of the selected mode. It gives as output a set of values which can be used by the control part, the different execution modes, or be directly displayed on the dash-board. The second part is the *control* part represented by the `SSM ModeState`. This part allows to select the mode to be activated according to the value of input buttons. The third part is a *mode-calculation* part. It is composed of five calculation parts related to the different modes of the system. The execution modes have

always the same interface which can facilitate the introduction, the deletion and the modification of modes.

VI. EXPERIMENTATION OF THE ICCG SYSTEM

A. Simulation and Verification

Based on the model presented by figure 7, and by using the Scade tools, we have performed some simulation and verification tests to assure the correct functioning of the ICCG system [9]. We have applied the verification process for the two design models: our model given by figure 7, and the model proposed in [9]. The results show that the separated verification of the different modules gives the same result if we verify all the system [9]. Moreover, the verification time of separated modules is much faster since the number of states of the checked automaton is much smaller. This difference in time relates also to the fact that the formulas of the verified properties are simpler in the case of modular verification. This verification technique, that we call *modular verification*, makes it possible to locate the errors well, to gain time, to gain memory space and to reduce state explosion problems.

B. Prototype

To test the ICCG system in a real situation, we have developed a simulation prototype using Visual Studio .Net and the C# languages. The first application of our prototype is a simulation map allowing to define a roadmap and to create the data base relating to this map. Generally, this application consists in representing graphically on a roadmap the speed limit of the different zones. For safety reasons, the critical points of the roadmap such as the exits of motorways can be treated in a more precise way. We can zoom on these zones to increase the accuracy and then apply the same optimization process.

The second application is to simulate the functioning of the ICCG in real conditions. We have connected our system to a GPS of type Garmin, which sends the car position each second to the ICCG. The prototype computes, in real time, the speed of the car, its direction and its current position using geographical data (latitude and longitude). According to these informations and to the data base of the roadmap, the prototype locates the current and the next position of the car and then the speed limits to be respected.

C. Field Test

A real test of our application was performed in Villeneuve d'Ascq using a laptop and a Garmin GPS embedded in a car (figure 8). It is a simple test of the system since it does not react directly on the speed of the car. In our case, the ICCG system informs only the driver by an alarm signal if the speed limit is exceeded. To display the different informations on the state of the system and the speed changes, we have proposed a graphical interface giving the roadmap, the car position, the next car position, the car speed, the speed limit,... We have also proposed an interface for the different buttons of the ICCG System as shown by figure 8.

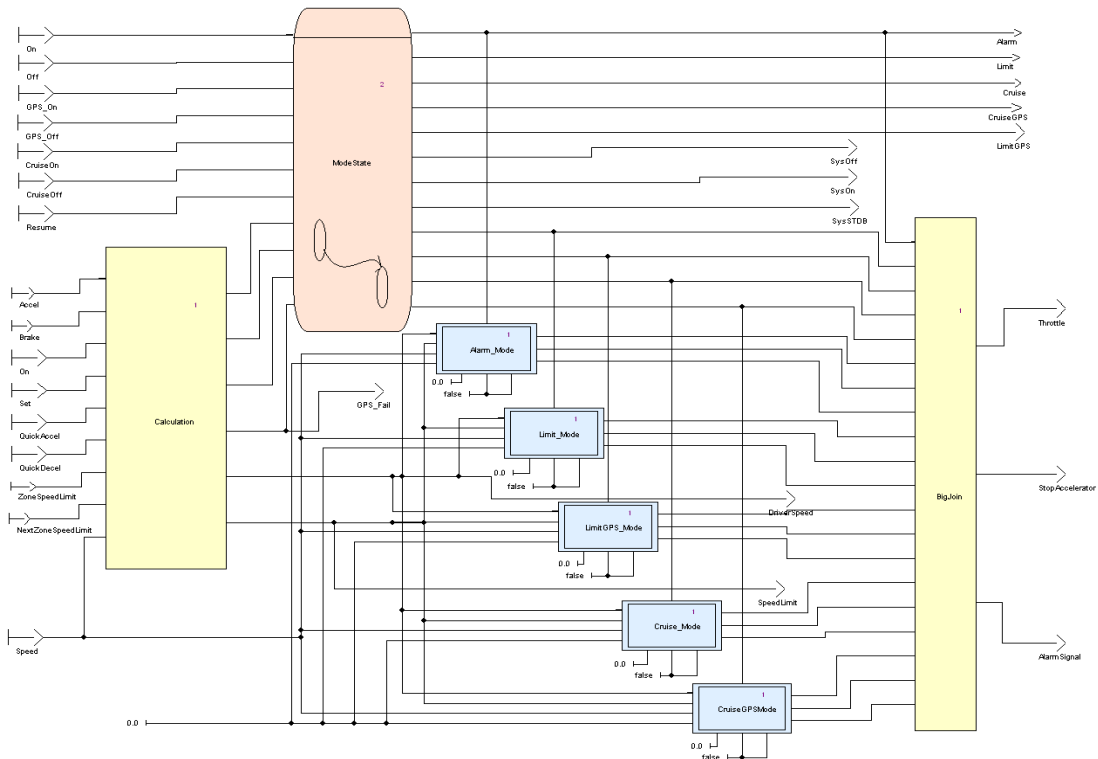


Fig. 7. Internal structure of ICCG model according to control/data flow separation methodology



Fig. 8. Real test

We have fixed the speed requested by the driver at 140km/h. In this case, the authorized speed limit will be always that of the current zone of the car. This test gave satisfactory results and showed that the detection of the different changes of zones is precise. For example, the passage from a 90km/h zone to a 70km/h zone was very quickly announced by the system and the alarm signal was activated until the speed of the car went below 70Km/h.

VII. CONCLUSION

We have studied the application of a control/data flow separation methodology to an automotive ICCG system, showing that the control/data flow mixture representation can make difficult the understanding of the system and the re-use of existing applications. We have proposed a design methodology

allowing to separate clearly control and data parts, and consequently give a good modular development. We illustrated the advantages of this methodology on the ICCG system, making it more readable and easier to maintain and to re-use. Also, this model allows a modular formal verification since the different modules of the system are easily localizable.

REFERENCES

- [1] Ouassila Labbani, Jean-Luc Dekeyser and Pierre Boulet, *Mode-automata based methodology for Scade*, In Springer, Hybrid Systems: Computation and Control, 8th International Workshop, LNCS series, pages 386-401, Zurich, Switzerland, March 2005
- [2] Florence Maraninchi and Yann Rémond, *Mode-automata: About modes and states for reactive systems*, European Symposium On Programming, Springer verlag, LNCS 1381, Lisbon, Portugal, March, 1998
- [3] Michel Marchi, Jacques Ehrlich and Laurent Salesse, *LAVIA: the French ISA project, main issues and first results on technical tests*, Proceedings of the 10th ITS Congress, Madrid, Spain, 2003
- [4] Torbjörn Biding and Vägverket, *Intelligent Speed Adaptation (ISA), Results of large-scale trials in Borlänge, Lidköping, Lund and Umeå during the period 1999-2002*. Technical Report, 2002
- [5] Veerle Beyst, *PROSPER: Project for Research On Speed adaptation Policies on European Roads, Final Report on Stakeholder Analysis*. Technical Report, version 1.4, 2004
- [6] Jean-Manuel Page, *A final technical report on the Belgian Intelligent Speed Adaptation (ISA) trial*. Technical Report, Project and research engineer, Belgian Institute for Road Safety, 2004
- [7] C. S. Jensen, H. Lahrman, S. Pakalnis and J. Runge, *The INFATI Data*. TimeCenter Technical Report, 2004
- [8] Nicolas Halbwachs, *Synchronous Programming of Reactive Systems*. Kluwer Academic Publishers, ISBN 0-7923-9311-2, 1993
- [9] Ouassila Labbani, Jean-Luc Dekeyser and Éric Rutten, *Separating Control and Data Flow: Methodology and Automotive System Case Study*. INRIA Technical report, RR-5832, February 2006