Rapid Prototyping of a TeleGeomatic Application

Charles ANDRE [1], Tullio Joseph TANZI [2]
(1) Laboratoire I3S, UNSA/CNRS
41 Boulevard Napoléon III,
06041 NICE cédex, France
(2) Ecole Nationale Supérieure des Mines de Paris, Pôle Cindyniques
B.P.207 06904, Sophia Antipolis - France,
andre@unice.fr , Tullio.Tanzi@cindy.cma.fr

1 Abstract

In safety-critical applications, a solution must be validated. Testing unproven solutions on the actual system is forbidden for safety reasons. The management of a motorway, especially when an accident happens, is such a safety-critical system. This paper proposes an enhanced safety management system for the motorways. The behaviour of the system is expressed with a synchronous model. This model is mathematically well-founded, which allows some formal proofs. Moreover, it can be efficiently compiled into a executable prototype on which extensive tests can be applied.

2 Introduction

There exist many ways to ensure safety on a motorway. Most of them are enforced by the law or regulations. In this paper we propose *an enhanced safety system* that should improve the use of the existing infrastructure and facilitate information exchange in case of accident.
In the field of motorway management, safety is of the utmost importance. Any attempt at modifying safety systems must be carefully screened. Candidate solutions must be validated, and if possible, in a formal way. So, when tackling the issue of improving a safety system, we have in mind several harsh constraints:
1 – All the services previously offered must be still offered, and with an at least equal quality of service;
2 – The solution must be *validated at the model level* for two reasons: first, the actual implementation is complex and expansive, second and above all, experimentation on the real system is risky and forbidden by the regulations;
3 – The solution must be described with an unambiguous, precise and rigorous model. This model must be understood by the experts in safety problems who have to assess the solution and who are not necessarily experts in formal models.

On a motorway, the prevention of accidents and the collection of information, when an accident occurs, involve several active (patrol vehicle, cameraman, control centre, …) and passive agents (e.g., emergency call terminals). These agents are part of a telegeomatic system: they interact in real-time in a distributed system. Interactions can be described in terms of discrete events. In what follows we use a graphical discrete event model for expressing the behaviour of the various agents. The model we choose: “syncCharts” [1] is a synchronous model, endowed with a mathematical semantics and suitable for expressing complex behaviours.

3 Specifications

On each side of the motorway there are “Emergency Call Terminals” (ECT) about 1 km away from one another. In case of trouble, a user can call the “Control Centre” from an ECT. The call is initiated by depressing a button. While the button is pressed, the user can speak to the distant operator through a microphone. Conversely, the operator gives instructions through the terminal’s loud-speaker. In a “district” (i.e., a well-delimited part of the motorway), there are about N=30 ETCs. ETCs are connected to the control centre by a heterogeneous physical line made of optical fibres, twisted-pair wires, … The line supports up to N simultaneous calls (one from each ECT). This worst case is, of course, very unlikely. Hence, the idea of enhanced ECT. An enhanced ECT has extended capabilities which are, in a decreasing order of priority:
1 – transfer of emergency calls (their main mission);
2 – transfer of images taken by a technician sent to the scene of an accident;
3 – tracking of the patrol vehicle and dialog between the control centre and the vehicle;
4 – checking of the line continuity, a background task.
Other functionalities could be added in the future.
In order to ensure this additional services, each ECT is equipped with a short-range, low-power radio link (frequency = MHz, power = mW) which can establish contact with its immediate neighbour ECTs. Figure 1 shows a simplified view of the system. “Cuts” are Boolean variables standing for possible line breaking. For convenience, we assign an integer identifier to each ECT. ECT_{k-1} (ECT_{k+1}, resp.) is said to be upstream (downstream, resp.) with respect to ECT_k. The role of each partner (patrol vehicle, control centre, cameraman, …) is beyond the scope of this short paper. Their mission is described in another paper [6].

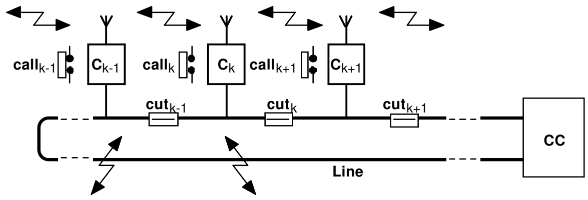

Figure 1: Overview of the system.

4 Expected behaviour

This application is distributed, reactive and subject to real-time constraints. The model we choose must capture:
- concurrency (many actors, with their own activity);
- communication (exchanges of signals and messages between actors);
- synchronisation,
- mutual exclusion,
- and, *pre-emption*.

The first four items are classical ingredients of co-operating systems. Pre-emption deserves a special attention. Pre-emption is the right given to an agent to forbid execution of other agents, either temporarily (suspension), or definitively (abortion). According to this definition pre-emption is a general means of

synchronisation and exclusion (see the G. Berry's paper about pre-emption [2]). Pre-emption may be very useful in expressing reactive behaviours.
D. Harel [5] introduced a graphical model called "statecharts" to specify reactive behaviours. This model is a state-based description of behaviours. Transitions (i.e., changes of states) are triggered by event occurrences. A transition may generate new signals, which in turn may cause triggering of other transitions. Statecharts also support hierarchy (states including state graphs) and parallelism (orthogonal component graphs with concurrent evolutions). Signals are instantaneously broadcast to all the components of the model. This is a *synchronous point of view*.
We have adopted a model derived from statecharts: the "syncCharts" [1], short for "synchronous charts". This model differs from statecharts in two main points:

- a richer and more convenient expression of pre-emption,
- a cleaner semantics, fully compatible with the semantics of the Esterel language [4].

Thanks to this full compatibility with Esterel, a syncCharts description can mix graphical and textual components. Graphical components describe behaviour in a visual way, generally appreciated by engineers. Textual components (i.e., Esterel statements) are useful for detailed or intricate behaviours.
Synchronous languages and related synchronous formalisms have been created to cope with reactive system programming. An enhanced ECT, a patrol vehicle … are such systems. That is the first reason why we choose the synchronous approach.
Advantages of the synchronous approach are many-fold:

1- The designer may focus on the "logic" of the communicating sub-systems. He/she expresses the behaviour of each components which interact through signals. There is no need for adding special mechanisms like semaphores, monitors, …
2- Synchronous modules compose very well, their parallel composition is deterministic (contrary to the classical asynchronous composition).
3- Sequence, parallelism, communication and pre-emption are orthogonal concepts, that is, they can be composed at any level of depth (hierarchy) and in any order. A consequence is that the expression of the behaviour of a component has not to be revisited when a pre-emption is applied to this component (modularity).
4- The semantics of the synchronous models is mathematically defined, so that there is no room for divergent interpretations. Due to their formal background, synchronous programs lend themselves to formal verifications, which is a must in safety-critical systems.

Of course, there exist limitations to the use of the synchronous approach. The instantaneous broadcast of information and the 0-duration of reactions are not realistic for many real-world applications.
Describing the behaviour of an agent with a synchronous model leads to a more coherent and efficient control: intra-component parallelism, synchronisation and communication are compiled out. Usually, remote interactions are not amenable to synchronous modelling. And yet, the distributed nature of the system at hand, is no impediment to its synchronous modelling. The reason is that our communication media guarantees message delivery delays. For inter-component communication, we use "weak synchrony": a signal emitted by a component is perceived by the other components at the next instant.

5 Example of modelling

In this short communication we have to select parts of the model.

5.1 Connection management

Central to the safety problem is the management of the medium access. A cell (the smart micro-controller we plan to implement in an enhanced ECT) receives and transmits messages from and to both the line and the radio.
Since, the line was first dedicated to vocal calls made by users in trouble, this medium should be the privileged medium for "urgent user's call" (denoted by URG in what follows). When not used, the line can be used for the various previously mentioned services (patrol vehicle tracking, image transmission from accident place, …) between the cells and the control centre. The radio is the natural support for communication between the cells and the patrol vehicle or the camera. Moreover, in the case of line breaking, the wireless medium can be substituted for the line, leading to a more dependable system.
Our solution must deal with concurrent communication and ensure:

- the highest priority to URG messages,
- then, image transmission,
- and, when possible, patrol vehicle tracking.

We consider 4 different modes:

- the "Normal connection" mode: it is the nominal running mode,
- three degraded modes: the "backward connection" and the "forward connection" modes entered when a break is detected on the line, in one direction. The "radio connection" mode is the mode in use when the ECT is isolated from the line in both sides.

Note that for simplicity, we assume that the radio is still operational. The syncChart below shows the transition between these modes. A macro-state is pre-empted as soon as the triggering event occurs.

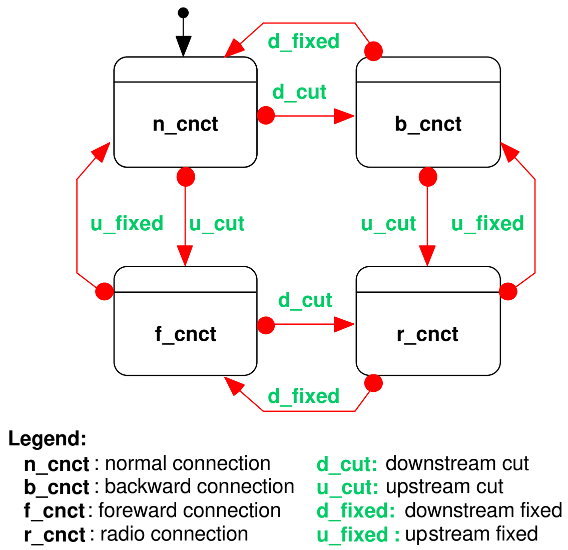

Figure 2: Modes of connection for a cell.

The control in a macro state must reflect the above mentioned priorities. The behaviour must make explicit the choice of what to transmit and on which medium. The choice must be deterministic (predictable behaviour). Because of the combinatorial nature of the selection, a textual representation has been preferred (an Esterel module).

5.2 Behaviour of the patrol vehicle

The patrol vehicle has also several functioning modes. Figure 3 shows its behaviour in the "normal patrol" mode. Every 30 tu (Time Unit), which corresponds to a distance of about 6 km, the vehicle sends a radio message to signal its identity and its position (signal pos). This repeated behaviour is expressed by a transition labelled with 30 tu, at the outmost level: each occurrence of this signal will cause a re-initialisation. When a cell receives this signal, the cell sends back an acknowledgement (signal ack). Messages can be lost. So, every 2 tu, the vehicle keeps sending its message up to the receipt of the acknowledgement signal. If 12 tu have elapsed since the first attempt at transmitting its position (i.e., 6 trials), a message of potential failure is recorded (signal fail). This information will be analysed by the maintenance service. Now, suppose you want to append a new service: giving the position on a request from the control centre (signal req). Using the orthogonality of the pre-emption, it is sufficient to add a new transition, outside the macro-state. This pre-emption forces a message sending by re-initialisation. Nothing has been changed inside the macro-state.

A last comment about determinism. Outer pre-emptions have priority over inner pre-emptions. Suppose that 2 tu occurs at the very same time than req. In this case, only req will act. When pre-emptions are at the same level (e.g., 30 tu and req), the priority must be explicitly given (integer value by the end of the arrow; the smaller the higher priority).

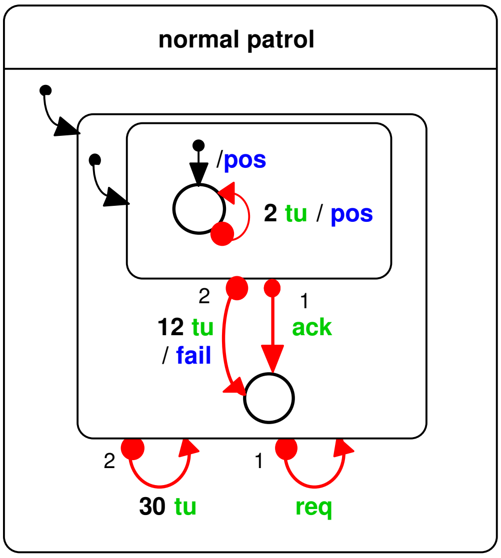

Figure 3: Behaviour of the patrol vehicle (normal mode).

6 Validation

6.1 Simulation

A syncChart can be compiled into an equivalent Esterel program (equivalent means that the behaviour of the syncChart and the behaviour of the associated Esterel program are strictly the same, this is guaranteed by the formal semantics of the model). The Esterel program is then compiled into a set of equations that expresses the control.

The Xes (X Esterel Simulator) tool, available in the Esterel distribution, allows the user to interactively simulate his/her program. Xes automatically constructs a dedicated dialog panel on which input signals can be set by the user, and which displays output signals resulting from the reaction. User's defined panels can also be used. A panel that represents the motorway system is under construction.

The simulator is mainly used for testing scenarios. Scenarios can be recorded and re-played. So, the designer may detect malfunctions, change the specification of the expected behaviour, re-compile and simulate again on previous scenarios and new ones.

This prototype animation is also an excellent tool for learning. A non specialist in reactive programming can easily experiment reactions of the system and suggest improvements. A library of critical situations and catastrophic scenarios can be developed by experts.

When then simulation gets through all the scenarios, it may still remain hidden bugs. The tools presented now, leads to a higher level of confidence.

6.2 Symbolic simulation

Xeve [3] is another tool for reactive system analysis. Xeve applies model-checking techniques and makes an exhaustive search for reachable states. When the designer wants to know if a safety property is satisfied, he/she has to compose the program with an *observer* of this property. The observer is usually expressed in Esterel, but a syncCharts observer can be used, as well.

Xeve is, in fact, an excellent "bug-finder". Due to its exhaustive search, Xeve can even test very unlikely scenarios. When a property is violated, Xeve returns a sequence leading to the counter-example. The analyse of this sequence often reveals deeply hidden bugs.

It seems that Xeve is a solution to safety-critical systems. Unfortunately, there exist restrictions. First, Xeve works on Boolean equations. When the syncChart uses "valued signals" (signals that convey values of a given type), the compilation generates a set of Boolean equations for the control, but also general data processing functions triggered by the Boolean control. In such a situation, Xeve cannot cover all the states, even for finite data types. The second limitation is that even for Boolean equations, the number of states can explode.

One can cope with the first limitation by "model cleaning". Only an expert can do this job consisting in discarding a valued signal or replacing it by, possibly many, pure signals. This is surely the weakest point of the approach.

Face to the explosion we verify sub-systems separately and then consider to what extent the property is still satisfied for the system.

7 Conclusion

In this paper we have proposed a synchronous approach to the design of a safety-critical system. The expected behaviours are expressed with syncCharts, a graphical synchronous model. Through partial examples we have shown how this model can describe complex reactive behaviours. We then explain how to test the solution, in depth, without making the actual implementation, which will be too risky.

8 References

[1] ANDRE Ch. Representation and Analysis of Reactive Behaviors: A Synchronous Approach, IEEE-SMC, Computational Engineering in System Applications, Lille (F), july, 1996, p 19-29.

[2] BERRY G. Preemption in Concurrent Systems, FSTTCS, LNCS n° 761, 1993, p 72-93.

[3] BOUALI A. Xeve: an Esterel Verification Environment, CAV98, Vancouver (BC), LNCS n°1427, 1998, p 500-504

[4] ESTEREL esterel v5, available at http://www-sop.inria.fr/meije/esterel

[5] HAREL D. A Visual Formalism for Complex Systems. Science of Computer Programming, vol 8, 1987, p 231-274.

[6] TANZI T.J, SERVIGNE S, GUIOL R. Real Time Information system for Risk Management on Motorway. CAISE'98 : the 10th Conference on Advanced Information Systems Engineering, Pisa, Italy, June 8-12 1998. 11p.