

SCADE Display® for the Design of Airborne and Ground-Based Radar Human-Machine Interfaces (HMIs)

Vincent Rossignol

Esterel Technologies, 9 rue Michel Labrousse, 31100, Toulouse, France

vincent.rossignol@esterel-technologies.com

Abstract:

Radar displays have evolved over time from being pure mechanically-driven devices (oscilloscopes) able of displaying raw video showing the operators the detected and amplified target return signal (and the receiver noise) to more evolved, computer-based, interfaces where digital processing allows cleaning up the display by eliminating noise and clutter, and creating precise symbols for each target. These radar interfaces are now becoming more and more interactive, giving radar display operators the ability to not only observe the images, but also interact with them and send commands.

This paper introduces the use of model-based tools, such as SCADE Display®, for the design of radar Human-Machine Interfaces (HMI), which are becoming intrinsically more complex, due to their interactive nature. It shows how the use of a certified tool like the SCADE Display KCG code generator, complying with the most stringent levels of criticality of the DO-178B and DO-178C standards in aerospace, combined with the reliance upon international standards such as OpenGL for the rendering of HMIs, increases productivity and reduces development and verification costs of either airborne or ground-based radar HMIs.

Keywords: radar display, model-based design, certification, DO-178B, DO-178C, SCADE Display

I INTRODUCTION

Radar displays have evolved over time from being pure mechanically-driven devices (oscilloscopes) able of displaying raw video showing the operators the detected and amplified target return signal (and the receiver noise) to more evolved, computer-based, interfaces where digital processing allows cleaning up the display by eliminating noise and clutter, and creating precise symbols for each target. Today, dedicated software application HMIs allow displaying and controlling multiple radar units and associated peripherals, in order to provide users with intuitive and simple display allowing to obtain the best performance from their surveillance system. Typically, these interfaces feature the control and view of one or more radars. Additional

controls for advanced users allow optimizing the radar settings and the HMI display characteristics for specific applications. These radar interfaces are now becoming more and more interactive, giving radar display operators the ability to not only observe the images, but also interact with them and send commands.

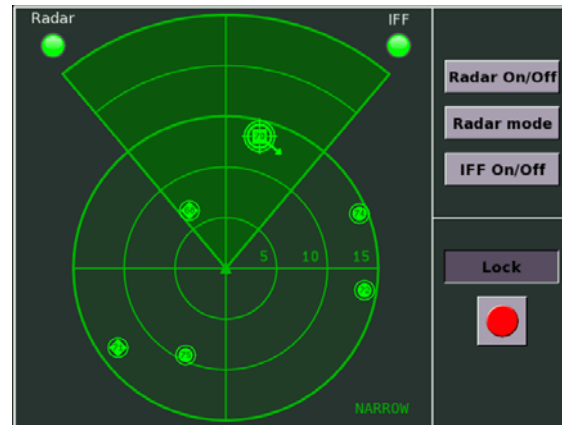


Figure 1: Typical interactive radar HMI (designed with SCADE Display)

In addition, safety and mission-critical software development requires rigorous processes, methods, and tools in order to ensure high quality and reliability of the developed HMI products. In the aeronautics industry, authorities, such as FAA or EASA, certify the produced software according to the DO-178B or DO-178C standard [1,2]. Certification is the process of verifying that the entire development life cycle conforms to the requirements of the standard.

The challenge is not only to master the cost of the development activities, but also the cost of verification activities, which in a certification context represent the highest part of the overall cost, while ensuring a high reliability and quality of the resulting software.

Thus, there is a need for dedicated HMI design tools and processes that are able to support both the required flexibility in the display creation for

various types of interactive radar displays and the increasing demand for safety in avionics displays.

In this paper, we show how the SCAD Display model-based approach promoted by Esterel Technologies [3] provides solutions for this challenge when applied to mission-critical embedded software for radar HMIs.

The paper is organized as follows:

- Section II presents the model-based design approach supported by SCAD Display, showing its benefits regarding the challenge.
- Section III focuses on the SCAD Display capabilities for abstracting HMI “interactivity” concepts at model level.
- Section IV presents the certified automatic OpenGL code generation focusing on the portability of generated code over a variety of radar display targets.

II MODEL-BASED HMI DESIGN WITH SCAD DISPLAY

The SCAD® model-based HMI development environment, including the SCAD Suite® and SCAD Display COTS tools, intends to be as intuitive as possible in order to provide both developers (i.e. programmers) and designers (i.e. non-programmers) a single and integrated tool suite concurrently supporting the prototyping and the development of embedded radar HMIs as models, without the need to write a single line of code.

From the purely graphical standpoint, SCAD Display offers an enhanced support of most OpenGL standard graphic features (including vector drawings, complex masks, bitmaps and texture-mapping, alpha-blending, haloing and anti-aliasing support), providing display system engineers with a high level of graphics quality and performance while benefiting from the inherent openness of OpenGL solutions, implemented on a large number of desktop, mobile and embedded targets. The SCAD structured formalism not only allows increasing reuse of specification parts, but also supports work share and concurrent development of graphical specifications.

The SCAD Display model-based approach supports the following features for the prototyping and specification of high quality embedded radar displays:

- WYSIWYG (“What You See Is What You Get”) editing, with support of OpenGL, OpenGL SC 1.x (Safety Critical) and OpenGL ES 1.1 and 2.0 (Embedded Systems) [4] graphic features (including

vector drawings, complex masks, advanced text primitives, bitmaps and texture-mapping, alpha blending, haloing and anti-aliasing support)

- Dedicated interactivity primitives, for streamlined support of embedded interactive HMIs (described in details in section III)
- Increasing reuse of specification parts as library objects, or widgets, supporting modular design, work share and concurrent development of graphical specifications
- Multi-file storage
- Extensive library of domain specific and interactive HMI widgets
- Rapid animation of the HMI specification through a simple and intuitive GUI, without any need to write complex scenarios
- Automatic generation of portable standalone executable

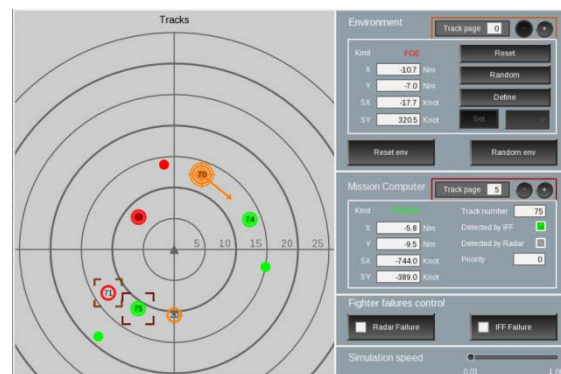


Figure 2: Complex radar HMI standalone executable

Not only such a model-based approach is the solid foundation for certified software code generation (as described in section IV), but it also allows for supporting the whole HMI system design life cycle, as required in the most critical radar software design processes. The requirements traceability, configuration management or documentation generation services, are greatly facilitated by such a model-based tool suite: all actors of the design process can easily keep the models synchronized, exchange automatically generated software documentation, trace modifications all along the life of the designed HMI system.

III INTERACTIVITY MANAGEMENT AT MODEL LEVEL

As far as interactive HMIs are concerned, SCAD Display provides capabilities for abstracting “interactivity” concepts at model level, so that no hardware architecture or low-level drivers are constraining the early prototyping or design stages.

The design formalism lays its foundations on three kinds of interactive primitives, all independent from the target platform: active areas, event listeners and behaviors. These specific modeling “bricks” (or primitives) for handling interactivity are called Interactors. Interactors allow specifying several kinds of user interaction with graphical objects, or zones, within a HMI model, with a great level of freedom.

The general workflow for defining user interaction in SCADE is the following:

- Mark active area(s) around the object(s), in order to detect presence of pointer(s) within the areas,
- Specify pointing device event listener(s), in order to detect when the user selects, clicks and/or drags the object(s),
- If needed, attach a behavior of the object(s) – with a SCADE Suite model using a behavior reference, or with imported code.

Three different kinds of interactors are available in the SCADE Display modeling language:

- Active areas
- Event listeners
- Cursor position request

Active Areas

Interactivity management in the displays is first provided through active areas: states and values are returned by taking into account the specified pointing device coordinates and the definition of the active area. For instance, a status inside/outside for the pointing device coordinate is returned. A ratio between the pointing devices coordinates and the geometric characteristics of the interactive area can also be returned.

An active area is defined (actually “drawn”) in the model as a dotted outline around an area in the tool visualization area. It allows detecting the inputs from the HMI user, i.e. when the pointer has entered or exited the area, and when the pointer is clicked and/or dragged over it.

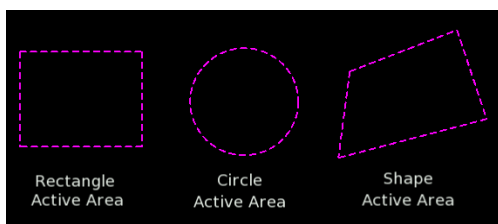


Figure 3: Active area primitives in SCADE Display

One can define three kinds of active areas in the models: rectangle areas, circle areas and arbitrary shape areas.

Event Listeners

Event listeners allow handling discrete run-time events from one or several keyboards and/or pointing devices. There are two kinds of event listener primitives: pointer and keyboard event listeners. In the SCADE Display generated code, those primitives are not tied to one particular operating system or windowing system implementation, which make these constructions easily portable and adaptable to any target device. A same single model will thus be easily deployed on various targets. This is one of the key assets of the SCADE model-based solutions for interactive HMIs.

A Pointer Event Listener has no graphical representation. At run-time and at each execution cycle, the pointer event listener records each pointer’s position, clicks of buttons, and other pointer events (i.e., wheel up and wheel down if actually tied to a mouse). A value is associated to each one of these events.

A Keyboard Event Listener has no graphical representation. At run-time and at each execution cycle, the keyboard event listener records a press of any key. A value is associated to each one of these events, corresponding to the code of the pressed key. Again, depending on the HMI target environment, the key code will either be a traditional ASCII code, or a custom code.

Cursor Position Request

The Cursor Position Request, kind of counterpart of the pointing device event listener, allows forcing at run-time the position of a given pointing device to a predefined position. This could be useful to indicate the crew in the cockpit that an (inter)action is required in the HMI in a given zone or place.

References, Behaviors & Imported Code

References are specific primitives in the SCADE Display modeling language that allow referencing external object/data/behavior in the specification. Amongst the various reference primitives, the Behavior allows specifying a SCADE Suite model in which is defined the behavior that one wants to associate with the interactors. Practically, the behavior primitive is dedicated to the behavior management of graphical objects with control logic. The SCADE Suite design formalism for control logic is especially handy to design state-based algorithms, such as the various states of a button (up/down/hover/etc.). Alternatively, the Imported Code reference primitive can be used to define within the graphical model the interfaces of an external function that is provided by the user as external C code.

Combining SCADE Display Interactors for Natural Interfaces

The arbitrary combination of those basic “interactivity” bricks by designers, completed by the power of expressivity – and safety – of a dedicated data and control flow language like SCADE Suite for the behavioral logic, not only allows supporting a traditional “widget-based” approach for interactive radar displays. It also paves the way for more “service-based” - or Natural – HMIs, like the ones which have already made their way to the consumer industry (e.g. in tactile music players, smart phones or tablets). Indeed, this new way of designing HMIs shall basically allow the airborne or ground operators interacting in a natural way with the radar systems.

As an example, with SCADE, in order to implement a now “classical” map zooming function with two fingers on a tactile screen, two *pointer event listeners* (one per finger) and one *behavior* construct for the zooming algorithm will be used. This design pattern can be stored as a SCADE library element – a kind of micro-model – for latter reuse in any another HMI model.

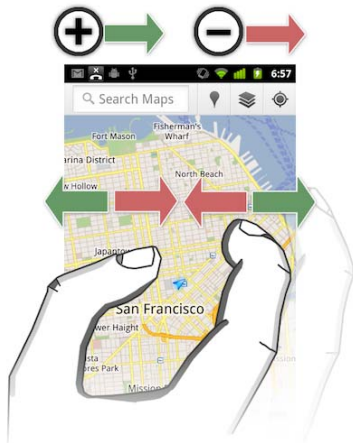


Figure 4: Map zooming function with two fingers

Now imagine – the radar screen is bigger, typically on the ground station, and the tactile technology allows for better precision – that you want to implement a more complex, complete or accurate zooming function that would rely on the five fingers of the same hand: three more *pointer event listeners* (five in total, one per finger) will be added in conjunction with a bit more elaborate *behavior* construct for this complex zooming algorithm. Here again, this new design pattern can be stored as an additional library SCADE element that will populate and grow the library of natural HMI capabilities.

The SCADE development environment already comes with a set of interactive widget examples

that are useful to quickly get started in HMI design. These widgets – buttons, sliders, lists, menus – can easily be tuned, being SCADE models themselves, by end-users for example to adapt to the specific needs of tactile or multi-touch interactivity. As the knowledge and deployment of natural user interfaces grow in the embedded displays domain, the SCADE library of widget and behavior examples will also expand.

IV CERTIFIED CODE GENERATION FOR A WIDE VARIETY OF DISPLAY PLATFORMS

The SCADE model-based design paradigm for cockpit HMIs not only allows managing and abstracting interactivity concepts at model level. Indeed, the strong integration between SCADE Suite and SCADE Display features the capability to automatically generate coupling code between the functional logic of the HMI and its interactive graphical representation. This allows minimizing the costly and error-prone task of re-writing the coupling code between both parts, often the most changing part of the system because interfacing various elements.

The SCADE model-based tool suite also features the automatic, one-click, generation of the HMI executable application for a variety of targets. As a matter of fact, the code generated out of SCADE is fundamentally independent from the target platform – whether it is the hardware and associated drivers, or the operating system – as no system calls are being performed in this generated code. The portability of HMI models as executable applications is thus greatly facilitated, as the only needs for adaptation often only reside in the main execution and interaction loops, or in the windowing system management, even though the always wider adoption of international standards like OpenGL – for the drawings – or EGL – as the associated windowing system – also facilitates this task.

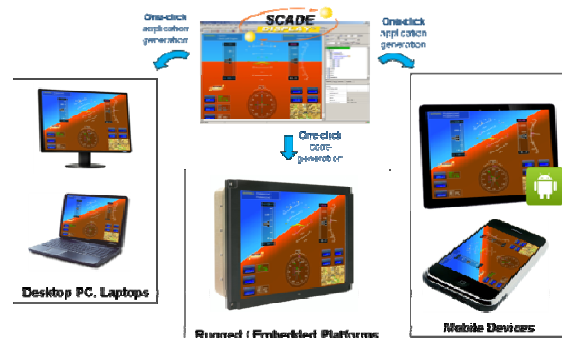


Figure 5: A wide range of code and application generation targets

As of today, SCADE natively supports the automatic generation of interactive HMI models for both Windows desktop PCs / laptops (all capable of managing OpenGL applications, either through dedicated graphics chips, or through motherboard built-in graphics adapters) and Apple iOS or Android compatible mobile platforms, such as smart phones or tactile tablets, through the native support of OpenGL ES 1.1 and 2.0 by SCADE Display. 'ES' stands for Embedded Systems, and is the OpenGL "flavor" now deployed on the majority of these targets, and making its way through the embedded radar displays industry, being very efficient for fast processing of a large number of graphical items, such as tracks on a radar display.

The tool suite is completed by the SCADE Suite KCG and SCADE Display KCG code generators, which allow the automatic generation of compact, readable, and traceable C code that is easily portable on any embedded or non-embedded target. Indeed, the generated code shows a complete independence from the hardware and (real time) operating systems.

SCADE Display, like SCADE Suite, has been designed from the ground up to meet DO-178B/C requirements. SCADE Display KCG has already been qualified a number of times as a development tool under DO-178B for level A software, and is also qualifiable for use in DO-178C level A embedded software. SCADE Display KCG qualification eliminates coding errors that could traditionally appear in the coding phases, as well as the need for low-level testing, enabling fast and safe late changes in the HMI designs. A dedicated handbook [5] addresses in details the issues of cost and productivity in the development of safe embedded software for avionics display applications. Such projects, driven by the DO-178B guidelines, traditionally require very difficult and precise development tasks, incurring high verification efforts.

CONCLUSION

When it comes to the newest kinds of embedded display platforms in the radar industry – large tactile screens, mobile devices – the unique combination of SCADE modeling and abstraction capabilities for various interactivity concepts, through basic "bricks" – the Interactors – with its intrinsic independency from the target platform in the generated code, provide cockpit and HMI designers with virtually infinite possibilities to create natural interfaces that can be deployed quickly on target. Behaviors, interactions, modalities are not "built-in" or "hidden" in the tool, as everything is a designed and managed as a model. This brings both HMI developers and

designers in the radar displays domain the necessary flexibility and advanced capabilities to create their own natural interfaces, based on models, without the need to know or manipulate low-level concepts, thus concentrating on their core task and providing them the needed freedom for a maximum of creativity.

The joint use of SCADE Suite and SCADE Display at the mock-up and prototyping phase of airborne or ground-based radar HMIs naturally supports the required continuity in the design process. This SCADE-centric process also includes the development and simulation of these HMIs, including within the framework of critical interfaces design – typically the ones you find in radars – as the SCADE code generation tool chain benefits from the DO-178B/C certifications credits.

REFERENCES

1. DO-178B Software Considerations in Airborne Systems and Equipment Certification, RCTA and EUROCAE, 1992
2. DO-178C Software Considerations in Airborne Systems and Equipment Certification, RCTA and EUROCAE, 2011
3. Esterel Technologies Web Site, <http://www.esterel-technologies.com>
4. Khronos Web Site, <http://www.khronos.org/opengl/>
5. "Efficient Development of Safe Avionics Display Software with DO-178B Objectives Using Esterel SCADE®" Methodological Handbook, *Esterel Technologies Web Site*

BIO DATA OF AUTHOR



Vincent Rossignol is SCADE Display and ARINC 661 Product Marketing Manager at Esterel Technologies. He has 10 years of experience as a Product Marketing Manager. He is the co-founder of the ARINC 661 product line at Esterel Technologies, and active member of the ARINC Cockpit Display Systems Subcommittee. He graduated in 2001 as engineer in computer science from SUPELEC (Ecole Supérieure d'Electricité, France) and UPV (Universidad Politecnica de Valencia, Spain).