Military
EMBEDDED SYSTEMS

Radar processing for naval upgrades: Software architecture is key to flexibility

DR. DAVID G. JOHNSON CAMBRIDGE PIXEL

0 Comments - Leave a Comment

Advances in naval radar capabilities need to be matched with flexible software architectures that can exploit the radar sensor to identify threats and provide robust software solutions, including multihypothesis, multimodel tracking, that can evolve over extended military program life cycles.

The latest generation of naval radars using coherent pulse Doppler technology provide enhanced detection of targets, including detection of asymmetric threats such as rigid-hulled inflatable boats (RHIBs) and jet skis. Such radars are optimized to detect small targets, even in clutter and in high-sea-state conditions to deliver enhanced situational awareness.

However, such advances in the sensor hardware must be matched by corresponding developments in the software processing of the data to automatically extract target information and to present the radar imagery on a naval radar display for operator interpretation. The software architecture must be flexible enough to accommodate evolutions in the capabilities of the radar, the operational requirements of the display, and the underlying technology that supports the implementation.

Modular software architecture is key

It is well understood that computer-processing hardware has a finite lifetime, beyond which it becomes uneconomic or just impossible to maintain. It is less well-known that software also has a lifetime and eventually needs replacing. The reasons are different, but the end result is the same: Unlike hardware, software is generally subjected to a process of continuous advancement to incorporate new capabilities or meet changing requirements. These changes beyond the original implementation make the software progressively more difficult to maintain and manage.

The software architecture to support radar processing and display in a naval console can be logically structured into modules for radar display, target tracking, and data fusion. Within these major subsystems, individual modules are required to provide the data-processing functions. The modular interaction of the processing functions ensures that modifications to the processing chain can make the software progressively more difficult to maintain and manage.

The software architecture to support radar processing and display in a naval console can be logically structured into modules for radar display, target tracking, and data fusion. Within these major subsystems, individual modules are required to provide the data-processing functions. The modular interaction of the processing functions ensures that modifications to the processing chain can be accomplished with minimal impact of other components in the system.

Take, for example, the target-tracking function. A software-based track extractor processes detections from the radar to identify targets of interest. These targets include other large vessels in range of the ship but also include small targets that may be potential threats, including wooden boats, RHIBs, periscopes, jet skis, and icebergs. Detecting these small targets is the challenge, with the goal being to detect a real target as soon as possible while minimizing the false-alarm rate.

The reliable detection of small, weak targets against a background of clutter takes time. It's a statistical process, which means that repeated detection of a radar return around the same location builds confidence that the radar echo is derived from a real target, rather than from a random process associated with sea or weather. The time taken to distinguish clutter from noise depends on the degree of clutter – a small target in an otherwise flat, calm sea is considerably easier to detect than the same target surrounded by clutter.

Multihypothesis, multimodel tracking

A modern implementation of a target tracker is built around the principles of multiple hypotheses and multiple models. The multiple hypotheses permit the tracker to consider different interpretations of the radar data. Is it a highly maneuverable target? Is it a stationary target seen in different locations because of measurement noise, or is it clutter? Then consider the different possibilities in parallel until the evidence for one hypothesis dominates. The multiple models permit the tracker to consider different target types in the same data, looking for different behaviors that conform to specific rules for the model.

Detecting very small targets, for example, requires some assumptions to be made to limit the search space and avoid clutter detections being incorrectly interpreted as highly maneuvering targets. Without intelligence in the processing, any number of “targets” can be observed by joining together the positions of random clutter detections.

The application of a multihypothesis, multimodel tracker enables the same radar data to be analyzed to detect both the obvious targets representing other ships in the coverage and also the smallest targets, potentially representing threats of interest. Target models exist as a package of target parameters that are chosen to reflect the types and behaviors of targets being searched for. This can include targets moving towards the radar operator’s own ship, for example.

The ability to create new tracking models that can drop in to the tracker architecture with minimal changes is a significant advantage for future upgrades and enhancements of a deployed radar processor. It means that the capabilities of the radar processor can evolve both as the capabilities of the sensor evolve, and also as the nature of the targets change. The pattern of behavior of a target can be represented by a model and searched for in the input. It's a case of knowing what to look for, then building a model that detects that pattern.

Cambridge Pixel's SPx software provides components for key radar-processing functions, including receipt from network, enhancement, scan conversion, tracking, fusion, and recording (Figure 1).

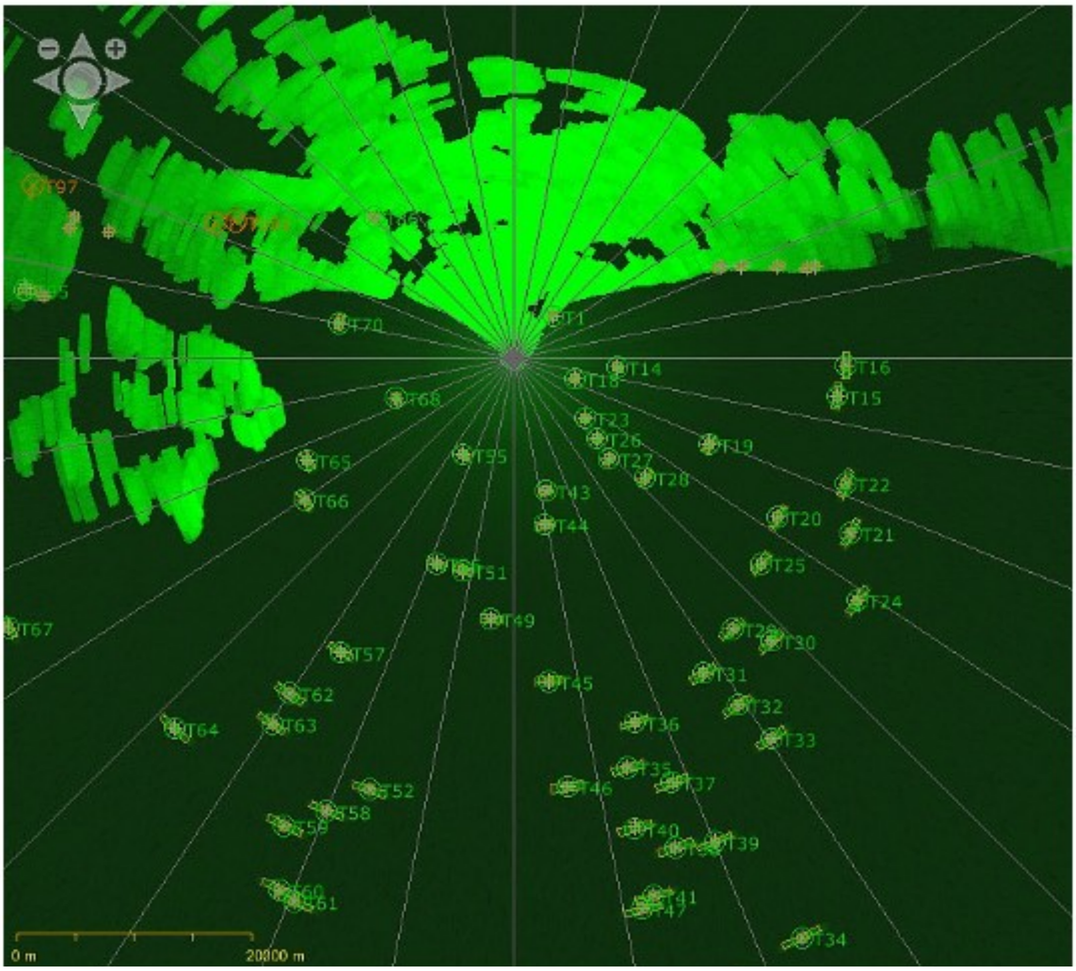

Figure 1: The target-tracking processor provides a local display enabling the visualization of radar video, plots, and track. This interface supports configuration and maintenance of the server.

However, in the context of delivering a scalable, adaptable solution for evolving requirements, the architecture of the software is as important as its function. The expectation of change, whether through enhancements to the radar, the need to detect specific types of target, or changes in the underlying operating system or graphics libraries, are fundamentally built into the SPx software architecture.

Naval Vigilance Radar system

Lockheed Martin UK-Integrated Systems' new Naval Vigilance Radar system is a working example of such an approach, combining an advanced naval radar with a modular processing and display architecture to deliver enhanced radar surveillance now and provide a clear route to adding extra functionality in the future.

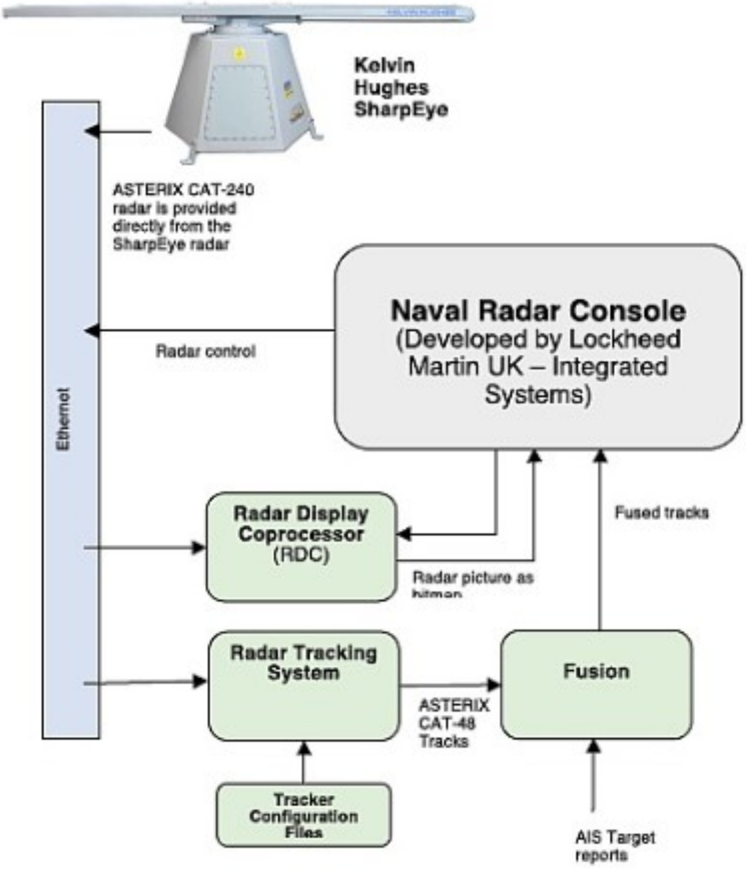
Kelvin Hughes SharpEye
ASTERIX CAT-240 radar is provided directly from the SharpEye radar
Ethernet
Naval Radar Console (Developed by Lockheed Martin UK – Integrated Systems)
Radar control
Radar Display Coprocessor (RDC)
Radar picture as bitmap
Fused tracks
Radar Tracking System
ASTERIX CAT-48 Tracks
Fusion
Tracker Configuration Files
AIS Target reports

Figure 2: The modular software architecture for Lockheed Martin UK-Integrated Systems' Naval Vigilance Radar system has the radar scan converter and target tracking running in separate software applications with a control interface to the main console application.

Lockheed Martin UK-Integrated Systems is under contract to the UK Ministry of Defence to install upgraded navigation radars on more than 60 Royal Navy platforms including on board type 23 frigates (as shown in lead photo). The contract will replace existing radars with the solid-state SharpEye radars from Kelvin Hughes. The software solution being developed by Lockheed Martin UK-

Integrated Systems uses core radar processing and display modules from Cambridge Pixel to handle the radar acquisition from the radar, the scan conversion, target tracking, and radar fusion.

As shown in Figure 2, in the case of the radar scan conversion, for example, a separate software application called the radar display coprocessor (RDC) handles the radar receipt, processing, and display. The RDC is loosely coupled to the main software applications through a network interface that supports the messages to control the radar view. The radar data itself flows only through the RDC software, not through the main application. The output of the RDC is a radar image that is made available to the main application to composite with the application graphics. Similarly, the radar-tracking capability is handled by a separate software application that interfaces to the radar video and provides track data into the main application.

Such modular software architecture for naval radar upgrades enables system integrators to implement enhanced radar systems today that exploit all the features of solid-state Doppler radars but also provide the flexibility for ongoing technology refresh throughout the ten- to 15-year program life cycle.

Dr. David G Johnson is technical director at Cambridge Pixel. He holds a BSc electronic engineering degree and a PhD in sensor technology from the U.K.'s University of Hull. He has worked in radar processing and display for 20 years and led teams developing software solutions for military radar tracking and radar scan conversion. Dave can be reached at dave@cambridgepixel.com.

Cambridge Pixel • www.cambridgepixel.com

THIS ARTICLE WAS PUBLISHED ON JANUARY 18th, 2017.