A Distributed Real-time Embedded Application for Surveillance, Detection, and Tracking of Time Critical Targets

Joseph Loyall, Richard Schantz
BBN Technologies, Cambridge, MA
{jloyall,schantz}@bbn.com

David Corman, James Paunicka
The Boeing Company, St. Louis, MO
{david.e.corman, james.l.paunicka}@boeing.com

Sylvester Fernandez
Lockheed Martin, Eagan, MN
sylvester.j.fernandez@lmco.com

Abstract

As computer systems become increasingly internetworked, there is a growing class of distributed real-time embedded (DRE) applications that have characteristics and present challenges beyond those of traditional embedded systems. They involve many heterogeneous nodes and links, shared and constrained resources, and are deployed in dynamic environments with changing participants. In this paper, we present a representative DRE application of medium scale that we are developing for the DARPA PCES program. This application consists of several unmanned aerial vehicles, command and control centers, and ground based combat vehicles to perform surveillance, detection, and tracking of time critical targets, an ever increasing threat in today's world. We describe the application, the scenario in which the application is being demonstrated, and issues and challenges associated with developing a DRE application of this complexity.

1. Introduction

Embedded applications, typically embedded on single processors or a few processors connected by a backplane, are becoming more internetworked as part of larger distributed real-time embedded (DRE) systems, such as industrial processing, military, and telemedicine systems. Although each embedded node might be a homogenous, closed, and static system, a DRE application has the following characteristics:

- Heterogeneous, shared, and constrained resources
- Changing modes, participants, and environmental conditions
- End-to-end requirements, e.g., requirements on the quality of information, are typically dictated by the user of the information, which can be remote from the provider of the information.

As part of DARPA's Program Composition for Embedded Systems (PCES) program [18], BBN, Boeing, and Lockheed Martin are developing a capstone flight demonstration of advanced capabilities for time critical target surveillance, tracking, and engagement. The PCES capstone demonstration is a medium scale DRE application, consisting of several communicating airborne and ground-based heterogeneous nodes in a dynamic environment with changing mission modes, requirements, and conditions.

The PCES capstone demonstration embodies a real-world context, scenario, and problem space illustrating the challenges and issues associated with the development and execution of DRE applications. Section 2 describes the context and scenario of the PCES capstone demonstration. Section 3 describes the makeup of the PCES capstone application. Section 4 describes the issues and challenges illustrated in the PCES capstone application. Section 5 describes some related work. Finally, Section 6 provides some concluding remarks.

2. Overview of the PCES Capstone Application

DARPA's PCES program is researching improved software engineering technologies for the construction

---

This work has been supported by DARPA and AFRL under contract numbers F33615-03-C-3317 and F33615-01-C-1847. Approved for Public Release, Distribution Unlimited.

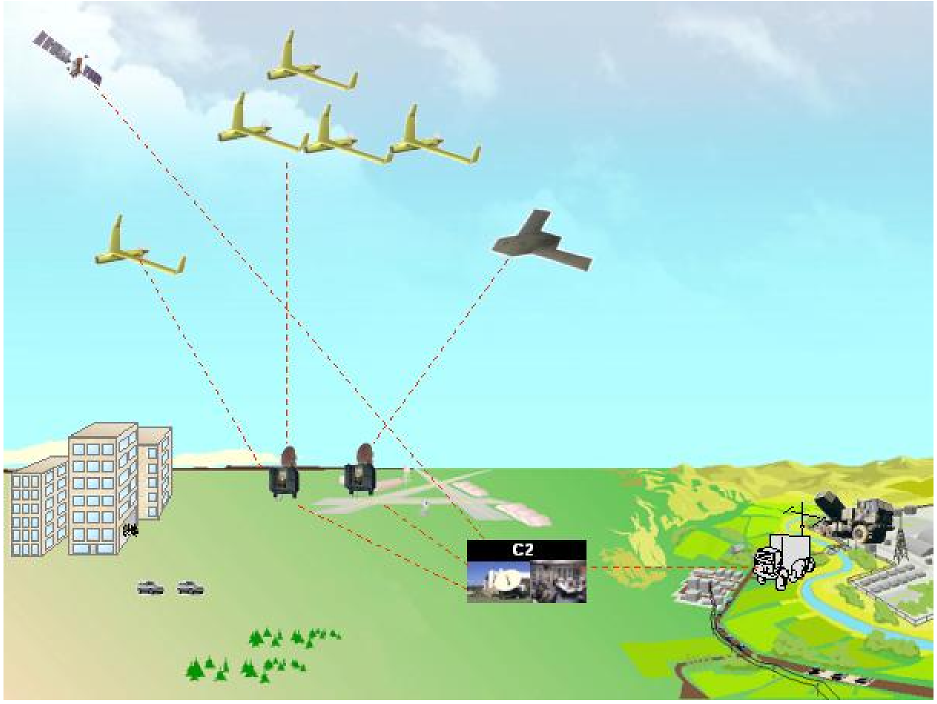

Figure 1. The PCES capstone demonstration concept of operations includes multiple RUAVs, combat vehicles, and command centers

of DRE systems. It is focusing on application to the important areas of time-critical targets (TCTs) [17]. As defined in [17], a TCT is a time sensitive target [28] "with an extremely limited window of vulnerability or attack opportunity." TCTs represent some of the most dangerous and challenging threats in modern theaters of war.

Operationally, TCTs present challenges because the time to detect and respond to a threat is shortened significantly from that of preplanned missions, to minutes instead of hours. Technologically, TCTs present the challenge of developing a Sensor to Decision Maker to Shooter/Weapon System (SDMS/W) chain of heterogeneous embedded nodes with predictable, real-time quality of service (QoS) end-to-end.

The PCES capstone demonstration is a realistic instantiation of a multiple SDMS/W chain with the real-time requirements of handling TCTs. The PCES capstone demonstration concept of operations is illustrated in Figure 1. It consists of a set of reconnaissance UAVs (RUAVs) performing theater-wide surveillance and target tracking and sending imagery to, and under the control of, a Command and Control (C2) Center. Specific RUAVs can be commanded to concentrate their surveillance on areas of interest (AOI). When a positive identification of a threat has been made, the commander can direct engagement by ground (i.e., US Army) or air (i.e., USAF manned or unmanned fighters) combat units. Specific surveillance units then gather battle damage indication (BDI) imagery and the SDMS/W process is repeated as needed.

Table 1 lists some of the advanced software engi-

Table 1. PCES software engineering technologies used in the capstone demonstration

Technology Area	PCES Technologies
Component middleware	CIAO [2], PRiSm [19]
QoS adaptive middleware	QuO [12], Qoskets [22]
Distribution middleware	TAO [25], JBI [27]
Modeling and assembly tools	DQME [31], CADML [4], PICML [4], GME [11]
Middleware services	Notification service [25], RT event channel [25]
Languages	RT Java [1], QuO QDL [12], IDL [25], CIDL [1]
QoS mechanisms and managers	CPU broker [6], Diffserv [9], RT CORBA [25]

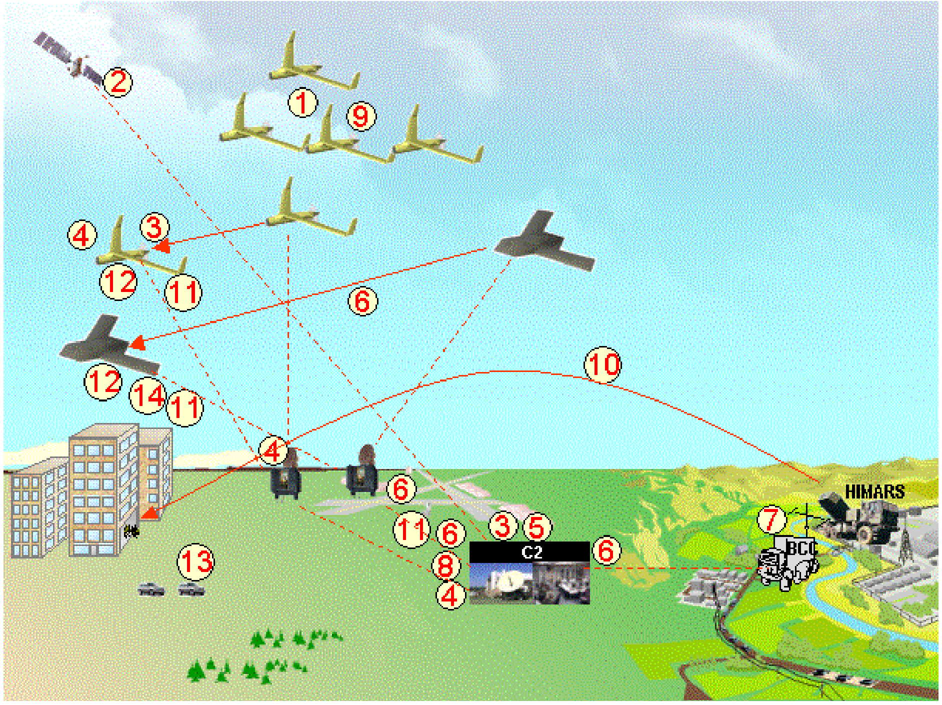

Figure 2. Scenario for the Spring 2005 scheduled PCES capstone demonstration

neering technology being used to construct the PCES capstone application. Many of these technologies were developed under the PCES program. Others were developed under earlier DARPA programs and integrated and enhanced under PCES.

Figure 2 illustrates a concrete scenario for the PCES capstone flight demonstration, to be conducted in Spring 2005, consisting of the following steps:

1. A set of RUAVs are performing surveillance in established patrols.
2. Intelligence assets establish the existence of a likely threat (i.e., a TCT) at a particular location.
3. A C2 node receives the intel alert and directs an RUAV to reroute to the site.
4. The RUAV collects surveillance imagery of the AOI and sends it to the C2 node through the RUAV ground station (GS).
5. The C2 verifies the TCT, determines precise coordinates for its location, and locates weapon systems available to respond to the TCT: a High Mobility Artillery Rocket System (HIMARS) missile launcher within range and a weaponized UAV (WUAV) that is currently out of range.
6. The C2 issues a call for fire to the Battle Command Cell (BCC), i.e., the C2 node for the HIMARS launcher, and sends the latest target image to the BCC; The C2 also directs the WUAV to move into range for subsequent weapon support.
7. The BCC computes the launch corridor, i.e., the path the missile will take, and communicates it to the C2 node.
8. The C2 node orders the RUAVs (and any other airborne or ground vehicles) to clear the launch corridor.
9. The UAVs clear the zone.
10. The C2/BCC issue the full command to fire; the HIMARS launches a missile.
11. After impact, the C2 commands the WUAV to collect BDI imagery and the RUAV to continue surveillance of the AOI.
12. The WUAV forwards BDI imagery to the C2 and to the BCC; the RUAV forwards reconnaissance of the AOI to the C2.
13. The C2 detects mobile targets departing from the scene and directs the WUAV to track and engage.
14. WUAV tracks the moving targets, calculates the weapon deployment parameters and engages the targets before they enter an area with high possibility of collateral damage.

3. Makeup of the PCES Capstone Application

The PCES capstone demonstration application consists of the following elements:

- A ScanEagle UAV [20] and its associated ground station serving as the live flight RUAV.
- Another ScanEagle UAV and its associated ground station serving as the WUAV.
- Multiple additional simulated RUAVs.
- A command and control center consisting of multiple computers executing situational assessment, control, and image receivers and displays.
- Five additional computers running the Battle Command Cell application.
- A HIMARS mobile missile launcher [8].

Figure 3. The ScanEagle is launched from a catapult (the image is from http://www.insitugroup.com/pages/Products/seascanSpecs.html)

The ScanEagle, illustrated in Figure 3, is a low-cost, long-endurance UAV developed jointly by Boeing and the Insitu Group [20]. The ScanEagle is well-suited for lengthy surveillance missions because it

- Is catapult-launched and retrieved using a sky-hook based system so it can be deployed in areas in which there is no runway
- Can operate semi-autonomously, flying a pre-planned flight pattern, and can operate for up to 15 continuous hours
- Can fly at about 16,000 feet making it difficult to detect and less vulnerable to hostile ground fire
- Carries either an inertially stabilized electro-optical or infrared camera, which can produce full motion video imagery as analog television signals to the ScanEagle ground station
- Has an internal avionics bay that can carry additional computing payloads.

The ScanEagle includes a paired vehicle and ground station. We are incorporating this into the PCES capstone demonstration by using some of the vehicle capabilities as is, e.g., the flight computer, and adding additional capabilities, some in an extra processor in the payload of the ScanEagle and others in an extra processor accompanying the ScanEagle ground station.

The design of the RUAV node is illustrated in Figure 4. The payload computer in the ScanEagle flight vehicle will be running PCES developed software to do route planning, threat evaluation, and space deconfliction, written in real-time Java and running on the TimeSys Linux real-time operating system (OS) [26]. The ground station will be extended with an extra processor that will be running PCES developed software (C++) to do image processing and QoS management. This ground station processor will also be running the TimeSys Linux real-time OS, with a PCES-developed CPU Broker [6] for CPU management and Differentiated Services [9] for network management. PCES developed QoS management software will be retrieving the analog imagery from the ScanEagle ground station receiver, digitizing it, and shaping it (e.g., changing its rate, size, and resolution) to fit the need of the C2 node and the capacity of the network and CPU resources available. More information about the end-to-end QoS management research illustrated in the PCES capstone demonstration is described in [14].

The WUAV, illustrated in Figure 5, similarly utilizes the payload computer and an extra processor connected to the ground station. ScanEagle, while not planned to be a weaponized platform, will be used as the WUAV surrogate – a mission nominally to be performed by platforms such as J-UCAS. It will be running either VxWorks or Timesys Linux on the payload computer and will include a Launch Acceptability Region (LAR) processing component. The extra ground

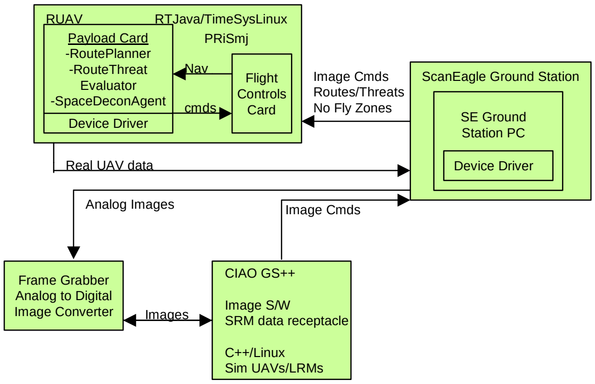

Figure 4. The PCES capstone demonstration adds a payload computer on the ScanEagle and an extra ground station (CIAO GS++) processor for the RUAV, both running PCES software

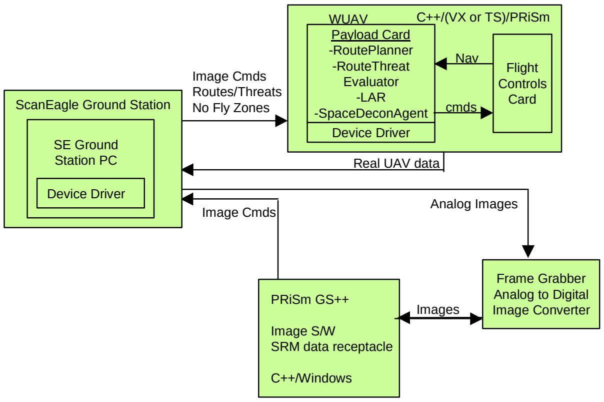

Figure 5. The WUAV ScanEagle is similar to the RUAV ScanEagle except running different software, OS, and infrastructure

station processor will be running PCES developed QoS management software similar to that on the RUAV ground station processor.

The QoS management software on the ground station processors is written using BBN's QuO middleware [12][21]. The WUAV version uses Boeing's PRiSm avionics component middleware [19], while the RUAV uses Vanderbilt's CIAO component middleware [2].

The C2 consists of multiple machines, some running MS Windows and others running Timesys Linux, as illustrated in Figure 6, with some processes written in Java and others in C++. The C2 nodes include situational assessment (SA) displays (one shown in Figure 7), command and control processing, and displays of the reconnaissance and BDI imagery (one shown in Figure 8).

The Battle Command Cell, developed by Lockheed Martin, is a prototype for future capabilities for the HIMARS launcher, a member of the multiple launch rocket system (MLRS) family. The BCC included in the PCES capstone demonstration, illustrated in Figure 9, is developed using PCES technology (components, middleware, and model-driven architecture) and includes situation awareness, embedded C2, and sensor displays of target imagery, as illustrated in Figure 10, that can be used to support Check Fire commands prior to launch and effects assessment after impact. The BCC concept moves some command responsibilities down to a lower level. A BCC, which is embedded within a launcher, is intended to control from 1 to 9 other launchers. It coordinates with the PCES capstone C2 node to issue Call For Fire (CFF) commands, deconflict the launch corridor, prepare the HIMARS for launch, launch the weapon, and then assess the effectiveness of weapon delivery.

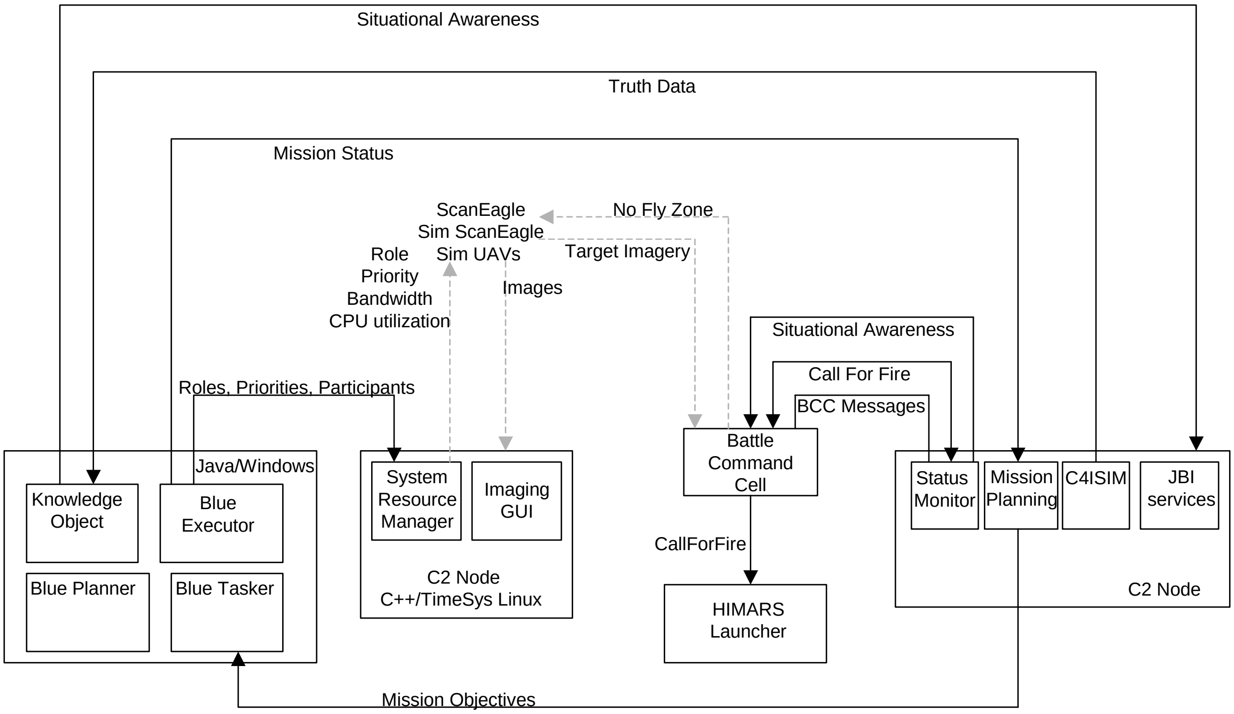

Figure 6. Architecture of the PCES capstone demonstration command and control

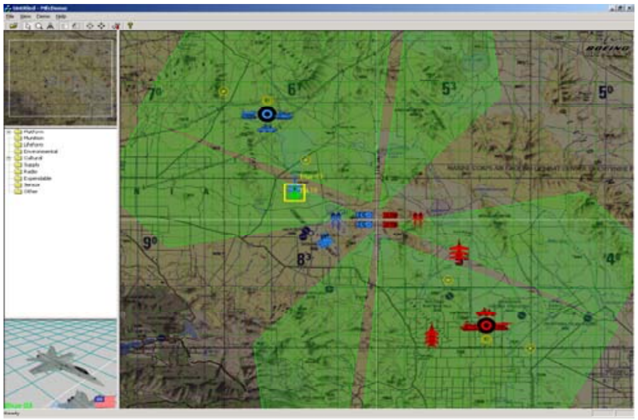

Figure 7. The PCES capstone demonstration includes a situational assessment display

Selected Image: UAV_SENSOR_5

Resource Allocation
Bandwidth Allocated (Mbps) 0.0
CPU Allocated (%): 0.0
DCSP Allocated: 0

Mission Information
Priority: LOW
Role: SURVEILLANCE

Data Management

	min	max
Frame Rate:	0.0	30.0
CPU Reservatior	10.0	10.0
Compression:	1	4
Cropping:	0	1
Scaling:	0	4

Figure 8. The PCES C2 node includes reconnaissance and BDI displays

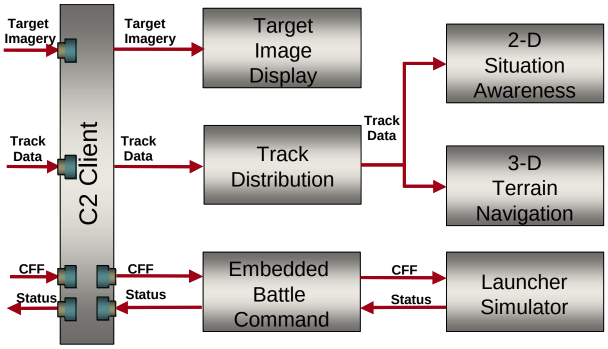

Figure 9. Architecture of the Battle Command Cell

Figure 10. The BCC node includes target image displays

4. DRE Characteristics, Issues, and Challenges Illustrated by the PCES Capstone Demonstration

The PCES capstone demonstration application exhibits challenges and issues that are indicative of medium-scale DRE applications with similar characteristics. The following sections discuss some of these.

4.1. Mission Modes and Roles Drive System Requirements

The way in which information is to be used in the mission determines the requirements on the system for the delivery, format, and content of surveillance and C2 information. In the PCES capstone demonstration, there are three roles that UAVs play

- Surveillance
- Target tracking and engagement
- Battle damage indication

The ScanEagle camera transmits imagery through a 2.4 GHz analog television downlink. We capture the imagery from the ScanEagle ground station receiver and pipe it through an analog to digital converter, so we can control the resolution, rate, and size of the imagery, based on how the imagery is to be used, the amount of resources available to transmit and process the imagery (i.e., network bandwidth and CPU), and the number of UAVs sharing the resources.

For RUAVs performing surveillance, the primary mission is to maximize the surveilled area with sufficient resolution in imagery for a commander to determine an item of interest. This means that imagery from each RUAV must be sent at a sufficient *rate* to ensure there are no gaps in surveillance coverage and at sufficient *size* and *resolution* for a commander to discern command level detail.

The minimum frame rate is computed based on the speed of the RUAV and the scan size of its camera to avoid any gaps in surveillance. For example, at the ScanEagle's cruising speed of 49 knots [10] (approximately 56 mph or 0.025 km/sec) and a scan size of

0.25 km by 0.25 km[‡], the frame rate should be no lower than 0.1 frame per second, or one image every 10 seconds. At a slower speed, higher altitude, or lower camera zoom, the frame rate could be lower, but the detail in each image would be affected.

Likewise, the camera can capture more detail in each image by increasing the zoom or lowering the altitude of the camera, but the minimum frame rate would need to be higher to avoid gaps in surveillance. The quality of each image can also be controlled by changing the scale, cropping, or resolution (e.g., through compression). The minimum acceptable resolution and size must be determined by commander input, by human factors research, or by experimentation.

When an RUAV is directed to reroute in response to a potential TCT, it changes to the target tracking role. The mission requirements of this RUAV are to provide high resolution imagery so that positive target or threat identification can be made. Since the RUAV can hover or circle over the AOI, the minimum rate is no longer determined by the speed of the RUAV, but by the speed of any mobile targets, or more accurately, the difference between their speed and that of the RUAV since the RUAV can pursue. Likewise, with stationary targets and the RUAV centered on the AOI, the image can be cropped or zoomed in to get higher resolution. For the target tracking role in the PCES capstone demonstration, a representative set of requirements is to send imagery at no smaller than 640x480 pixels, no data loss, and greater than 2 frames per second.

After target engagement, the WUAV performs the BDI role. The WUAV needs to provide regular imagery until a human operator determines that he has sufficient detail to discern battle damage. Imagery from the WUAV does not need to start immediately, since dust and smoke will obscure the scene immediately after engagement, but once imagery has started, high resolution imagery must be delivered regularly until a commander decides it is sufficient.

4.2. Challenges of Distribution and Heterogeneity

Standalone embedded systems are traditionally homogeneous, static, and closed. The PCES capstone application consists of embedded nodes – avionics, sensors, weapon systems, command and control – with their strict predictability and control requirements, but these are networked into a larger system of embedded systems, with the corresponding challenges of distributed, heterogeneous, and networked systems. Each embedded node is no longer a closed, controlled environment to the extent that its operation is dependent upon or can affect the operation of other embedded nodes.

The PCES capstone demo consists of subsystems written in different languages and running on different platforms, both software and hardware. It therefore exhibits some of the following characteristics and challenges related to distribution and heterogeneity in DRE systems.

Combining multiple subsystems to work together. This includes

- Defining and implementing interfaces between subsystems
- Managing differences in data formats and information structure
- Managing interdependencies between subsystems and contention for shared resources and services

End-to-end requirements. As mentioned in Section 4.1, the system requirements are based on the mission and emanate from the C2 node(s). However, the means to enforce and satisfy the requirements are distributed throughout the system. There is a challenge to distribute the requirements information and manage the system and local resource managers, including mediating conflicting resource needs. A description of how we provide end-to-end QoS management in the PCES capstone demonstration application is provided in [14].

Distributed spheres of control. There are multiple centers of control in the system. The C2 node is the main center of control, as its name implies. However, there are also control stations associated with each UAV and with the HIMARS missile launcher (i.e., the BCC). Control cannot be centralized, as the latency and complexity of controlling every subsystem, every sensor, and every actuator no matter how distant from a single point would limit the scale and scope of the system. Control and authority must be distributed, with the C2 providing mission requirements, policy, and direction, but with more local control authority enforcing it.

4.3. Joint Services

The PCES capstone demonstration application combines US Air Force and US Army operations, with the associated challenges. Our emphasis is mainly on the technical challenges, such as exchanging information between the subsystems, interfacing the subsys-

---

[‡] The actual scan size of the ScanEagle camera at any time is based upon its zoom and the altitude of the ScanEagle.

tems, and managing the interdependencies and contention for shared resources and services.

An equally challenging set of issues is that dealing with operation and doctrine. We touch upon those in the capstone with the shared authority for engaging a target. Currently, the US Army and US Air Force have different ways of approaching the issuing of fire commands and deconfliction. We simulate a potential approach to shared authority by having the C2 issue the initial call for fire and the BCC determine the launch corridor to be cleared. In our scenario, only when both authorities have verified that the corridor is cleared does the BCC issue the final call for fire, which actually launches the weapon. Actual doctrine determining the command authority and order of actions is beyond the scope of, but should be supported by, the technical solutions we are developing.

4.4. Construction of DRE Systems

The PCES capstone application illustrates challenges in the manner in which medium and large scale DRE systems are, and will be, built. Traditionally, these types of systems are built as one of a kind, stovepiped systems. Within the PCES program, we are developing technologies to improve the software engineering of DRE systems and are applying them to the construction of the PCES capstone demonstration application in the following ways:

- *Construction by composition*. We are developing subsystems, e.g., the BCC, the C2, and the UAVs, separately and composing them together using well-defined, standards-based interfaces, including the Joint Battlespace Infosphere (JBI) [27] and CORBA services (Notification Service and Event Channels).
- *Separation of concerns*. The functionality of the application is developed separately from the quality of service concerns. The QoS concerns are added to the functionality using aspect-oriented and QoS encapsulation technology [5][21].
- *Component-based technology*. The functional and QoS elements of the architecture are standards-based components (in two different component models), supporting the automated assembly and deployment of the application [2][19][22].
- *Product line separation of development roles*. We follow the process of separating the roles of developers that create generic, reusable components; developers that create application specific components; and integrators that assemble the application from components [23].

5. Related Activities

Our development of the PCES capstone demonstration builds upon previous flight demonstrations we have conducted or been involved in.

In the Weapon System Open Architecture (WSOA) program, we developed and prototyped capabilities for aircraft to do mission replanning *while airborne en route to a target* (as opposed to on the ground prior to a mission). It involved establishing a NetMeeting-like interface and browser capability between a C2 aircraft and a fighter aircraft, to enable the fighter's weapons officer to browse the C2 intelligence database and to exchange map and intelligence information, in addition to the radio based voice exchanges previously possible. We conducted a flight demonstration involving an AWACS and F-15 fighter in December 2002 that illustrated the ability to exchange imagery in real-time between the aircraft while they are in the air [7][13][30].

In another recent flight demonstration, we flew UAVs with advanced control systems for the DARPA Software Enabled Control program in June 2004 [24].

The PCES capstone application exemplifies the challenges and capabilities in DRE systems, which are at the core of a number of current military programs, such as the following:

- Future Combat Systems (FCS), the US Army's next generation of systems for network centric combat capabilities [29].
- DD(X), the US Navy's next generation surface combat ship, with network centric computer control as a key architectural element [3].
- Single Integrated Air Picture (SIAP), a joint services initiative to increase the interoperability and information exchange between weapons systems [16].
- FORCEnet, a US Navy initiative to link warfighters, sensors, C2, platforms, and weapons into a network-centric, distributed system [15].

6. Concluding Remarks

This paper described the PCES capstone demonstration as a representative application for DRE systems, in terms of its characteristics, the challenges it presents, and the way it is being constructed.

The main conclusion we draw is that the complexity and capabilities of DRE systems are increasing, enabled by the abilities to network embedded systems and the need to increase the capabilities and scope, and reduce the timeline, of military systems. This is exemplified by the number of current military programs that have network centric, DRE systems as their bases. Software engineering technology must evolve to support this trend, to manage the complexity, and to maintain our ability to construct these systems with confidence, as is being done under programs like PCES. Applications such as the PCES capstone demonstration are important to demonstrate and evaluate emerging software engineering support.

The PCES capstone demonstration is scheduled for flight demonstration in Spring 2005. As part of this, we plan to conduct experiments evaluating the PCES technology contribution to easing the burden of constructing DRE systems and increasing their capabilities.

Acknowledgements

We are grateful to all the program managers involved with the PCES project, including Dr. Joe Cross, Dr. Gary Koob, Dr. Doug Schmidt, Daniel Schreiter, Ron Szkody, Ray Bortner, and Marvin Soraya.

References

[1] Gregory Bollella, James Gosling, Benjamin Brosgol, James Gosling, Peter Dibble, Steve Furr, and Mark Turnbull. *The Real-Time Specification for Java,* Addison-Wesley Pub Co, 2000.

[2] Component Integrated ACE ORB (CIAO), http://www.cs.wustl.edu/~schmidt/CIAO.html

[3] DD(X), http://www.globalsecurity.org/military/systems/ship/dd-x.htm

[4] CADML (Component Assembly and Deployment Modeling Language) and PICML (Platform Independent Component Modeling Language), http://www.dre.vanderbilt.edu/cosmic/

[5] Gary Duzan, Joseph Loyall, Richard Schantz, Richard Shapiro, and John Zinky. "Building Adaptive Distributed Applications with Middleware and Aspects." International Conference on Aspect-Oriented Software Development (AOSD '04), Lancaster, UK, March 22-26, 2004.

[6] E. Eide, T. Stack, J. Regehr, and J. Lepreau. Dynamic CPU Management for Real-Time, Middleware-Based Systems. Proceedings of the Tenth IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2004), Toronto, ON, May 2004.

[7] Christopher D. Gill, Joseph P. Loyall, Richard E. Schantz, Michael Atighetchi, Jeanna M. Gossett, David Corman, "Integrated Adaptive QoS Management in Middleware: A Case Study", 10th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'04), May 25 - 28, 2004, Toronto, Canada.

[8] High Mobility Artillery Rocket System, HIMARS, http://www.army-technology.com/projects/himars/

[9] IETF, An Architecture for Differentiated Services, http://www.ietf.org/rfc/rfc2475.txt

[10] Insitu Group, ScanEagle A Specification Sheet, http://www.insitugroup.net/Downloads/ScanEagle_A.pdf.

[11] A. Ledeczi, M. Maroti, A. Bakay, G. Karsai, J. Garrett, C. Thomason, G. Nordstrom, J. Sprinkle, and P. Volgyesi. "The Generic Modeling Environment," *WISP'2001*, Budapest, Hungary, May 24-25 2001.

[12] Joseph Loyall. "Emerging Trends in Adaptive Middleware and its Application to Distributed Real-time Embedded Systems," *Third International Conference on Embedded Software (EMSOFT 2003)*, Philadelphia, Pennsylvania, October 13-15, 2003.

[13] J. Loyall, J. Gossett, C. Gill, R. Schantz, J. Zinky, P. Pal, R. Shapiro, C. Rodrigues, M. Atighetchi, and D. Karr. "Comparing and Contrasting Adaptive Middleware Support in Wide-Area and Embedded Distributed Object Applications," *Proceedings of the 21st IEEE International Conference on Distributed Computing Systems*, Phoenix, Az., April 2001.

[14] P. Manghwani, J. Loyall, P. Sharma, M. Gillen, and J. Ye. "End-to-end Quality of Service Management for Distributed Real-time Embedded Applications," submitted for publication.

[15] Richard W. Mayo. "ForceNet: Turning Information into Power." *Proceedings Magazine*, February 2003. http://www.usni.org/Proceedings/Articles03/PROmayo02.htm

[16] Louis S. Metzger and Donald R. Erbschloe. "Who's Got the Big Pict1ure? – Single Integrated Air Picture, integration of air reconnaissance systems," *Aerospace Power Journal*, Fall 2001.

[17] BGEN Jim Morehouse. "Time Critical Targeting," 2002 DoD Interoperability Conference, 25-28 March 2002. http://www.dtic.mil/ndia/2002interop/morehouse.pdf

[18] Program Composition for Embedded Systems, http://dtsn.darpa.mil/ixo/programdetail.asp?progid=69

[19] Wendy Roll. Towards Model-Based and CCM-Based Applications for Real-Time Systems, Proceedings Sixth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC '03), Hokkaido, Japan, May 14-16, 2003.

[20] ScanEagle, http://www.boeing.com/defense-space/military/unmanned/scaneagle.html.

[21] Richard Schantz, Joseph Loyall, Michael Atighetchi, and Partha Pal. "Packaging Quality of Service Control Behaviors for Reuse," *ISORC 2002, The 5th IEEE International Symposium on Object-Oriented Real-time distributed Computing*, April 29 - May 1, 2002, Washington, DC.

[22] Praveen K. Sharma, Joseph P. Loyall, George T. Heineman, Richard E. Schantz, Richard Shapiro, Gary Duzan. "Component-Based Dynamic QoS Adaptations in Distributed Real-Time and Embedded Systems," *In-*

ternational Symposium on Distributed Objects and Applications (DOA), Agia Napa, Cyprus, October 25-29, 2004.

[23] D. Sharp, "Reducing Avionics Software Cost Through Component Based Product Line Development," *Proceedings of the 10th Annual Software Technology Conference*, 1998.

[24] Software Enabled Control, http://dtsn.darpa.mil/ixo/programdetail.asp?progid=39

[25] TAO, The ACE ORB, http://www.cs.wustl.edu/~schmidt/TAO.html

[26] TimeSys Corporation, "TimeSys Linux GPL: Performance advantages for embedded systems," 2003, white paper, version 1.1. http://www.timesys.com/index.cfm?bdy=linux_bdy.cfm

[27] United States Air Force Scientific Advisory Board Report on Building the Joint Battlespace Infosphere, Volume 1: Summary, SAB-TR-99-02, December 17, 1999, Cleared for Open Publication February 2000.

[28] US Joint forces Command Joint Warfighting Center and OSD Joint Warfighters Joint Test and Evaluation. "Commander's Handbook for Joint Time-Sensitive Targeting," March 22, 2002. http://www.jwf.jte.osd.mil/pdf/tsthndbk.pdf

[29] Frank Vizard. Future Combat. Scientific American.Com, January 13, 2003. http://www.sciam.com/article.cfm?articleID=000C031A-12A4-1E1E-8B3B809EC588EEDF

[30] World News Roundup, Aviation Week & Space Technology, January 13, 2003, p. 394.

[31] Jianming Ye, Joseph Loyall, Richard Shapiro, Richard Schantz, Sandeep Neema, Sherif Abdelwahed, Nagabhushan Mahadevan, Michael Koets, Denise Varner. "A Model-Based Approach to Designing QoS Adaptive Applications," *The 25th IEEE International Real-Time Systems Symposium*, Lisbon, Portugal, December 5-8, 2004.