UxV Software Systems, An Applied Research Perspective

21st March 2005

Greg Broten
Defence Research and Development Canada - Suffield

1 Context

Defence Research and Development Canada has over 20 years of applied experience in developing and implementing remotely operated vehicles. These tele-operated systems ranged included as a boat, a Bobcat, a backhoe excavator, and numerous other wheeled vehicles as shown in Figure 1.

![Figure 1: Cat, Barracuda, Edargo, Kodiak, Rommids, HazMat and an Indoor Robot](image)

The fruits of this research and development have yielded true military vehicles such as the Multi Agent Tactical Systems\(^1\) (MATS) and the ILDP landmine detection vehicle. Examples of tele-operated vehicles developed at DRDC are shown in Figure 2.

\(^1\)A tele-operated platform with onboard nuclear, chemical and biological detection equipment.
Continuing a tradition of innovative development, the Tactical Vehicle Systems Section of Defence R&D Canada-Suffield has been tasked with researching and developing innovative autonomous vehicles that will assist the Canadian Forces in performing their duties in the 21st century. This research will continue for many years, on many different types of unmanned vehicles (UxVs)\(^2\) and thus requires a modular, extensible, flexible and scalable software architecture.

2 Problem

UxV platforms vary widely in application environment, size and capabilities as shown in Figure 3.

These UxV platforms have payload capacities that range from hundreds of kilograms and cubic meters of space to a payload of a few kilograms with only a minimal amount of space available. These platform configurations also have varying sensor compliments that result in unique sensing capabilities for each UxV. This wide range of platform configurations require a modular, flexible and extensible software and hardware architecture that can be scaled to suit the size of each individual platform.

In many situations UxV operate as teams where a group of vehicles coordinate their actions. A software architecture must also inherently support multiple vehicle configurations.

\(^2\)There are five anticipated classes of Unmanned Vehicles (UxV): fixed or rotor wing aircraft Unmanned Air Vehicles (UAV); typically tracked, wheeled, legged Unmanned Ground Vehicles (UGV); stationary monitoring Unattended Ground Sensors (UGS); untethered, propeller or buoyancy driven, Unmanned Underwater Vehicles (UUV); and light propeller driven Unmanned Surface Vehicles (USV).
Figure 3: Hummer, Raptor, Segway RMP, PAW, Extreme Testbed, Yamaha Rmax and the SilverFox

3 Forces

- To create a software architecture that is applicable to a wide variety of platform configurations. To achieve this extensibility and scalability the architecture should inherently support complex and distributed software implementations.

- The software architecture must support both single vehicle configurations and multi-vehicle systems.

- The architecture should be flexible to allow individual researchers to implement solutions that are appropriate to their requirements. Put another way, the architecture should not be rigidly defined and thus force the researcher to implement a solution that is not well suited to his/her problem.

- The developed software should be modular and portable. A sensor driver developed for a given UxV should be usable on a different UxV platform.

- Given that current and future UxV will use multiple processor configurations, a software architecture must inherently support and scale to multiple processor implementations.

- A UxV should use the same mechanisms to receive data from a device located on a remote UxV platform as it uses for a device located on its own platform.
UxVs operate in real world environments and thus the architecture must be responsive to real-time requirements.

4 Solution

A software architecture that is modular, extensible, flexible and scalable is described in [1] and [2]. This architecture uses a strong set of infrastructure services that allows researchers to quickly and easily develop the software that underpins autonomous systems. The key aspects of the infrastructure services are its support of a framework and communications middleware.

This software architecture the TAO communications middleware. TAO is a real-time implementation of CORBA based upon the ACE communications middleware. Miro, which is based CORBA, was determined to be the most appropriate framework. Miro uses the capabilities of CORBA to create a framework which implements the set of components required by a Robosoccer robot, where each component has a clearly defined interface that allows it to share data and methods with other components. While the Miro framework was originally developed for soccer playing robots, its modular, flexible, scalable and extensible design allows it to be easily and quickly extended. DRDC has modified Miro to serve the needs of outdoor UxVs. Extending the capabilities of Miro to support outdoor UGVs involved implementing components that encapsulated the specific functionality required by the target UxV and integrating these new components into the existing Miro structure.

References
