A Mission Management Framework for Unmanned Autonomous Vehicles

Eskinder Asmare, Anandha Gopalan, Morris Sloman, Naranker Dulay, and Emil Lupu

Department of Computing, Imperial College London, London SW7 2RH
{e.asmare,a.gopalan,m.sloman,n.dulay,e.c.lupu}@imperial.ac.uk

Abstract. Unmanned Autonomous Vehicles (UAVs) are increasingly deployed for missions that are deemed dangerous or impractical to perform by humans in many military and disaster scenarios. UAVs in a team need to operate in sub-groups or independently to perform specific tasks, but still synchronise state information regularly and cope with intermittent communication failures as well as permanent UAV failures. This paper describes a failure management scheme that copes with failures, which may result in disjoint sub-networks within the team. A communication management protocol is proposed to control UAVs performing disconnected individual operations, while maintaining the team’s structure by trying to ensure that all members of the mission rendezvous to communicate at intermittent intervals. The evaluation of the proposed approaches shows that the schemes are scalable and perform significantly better than similar centralised approaches.

Keywords: Autonomic management, collaborating autonomous vehicles, mission management, communication failure recovery.

1 Introduction

Unmanned Autonomous Vehicles (UAVs) are mobile robots that are often used in civilian disaster-relief missions and military scenarios to reconnoiter in areas which are dangerous or impractical for humans. A challenge in using UAVs for such missions is enabling adaptive self-management so that they can automatically adapt to changes in context and failures without human intervention. Collaborating UAVs form a Self-Managed Cell (SMC) [14], a general architectural pattern for realising self management of individual and teams of UAVs. An SMC team consists of multiple UAVs with at least one commander, which could be a human or another UAV. The commander is provided with a mission specification by its command base and assembles the required UAVs to perform the mission. The mission specification [17] defines how specific roles are assigned to certain UAVs based on their credentials and capabilities.

The mission specification also defines a role management hierarchy and the behaviour of these roles in terms of policies specified using the Ponder2 [17] policy specification language. When a mission is instantiated the commander will

The original version of this chapter was revised. The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-3-642-01802-2_30

© ICST Institute for Computer Science, Social Informatics and Telecommunications Engineering 2009
download its role behaviour specifications (the policies) and start the mission. When new UAVs come into the communication range the commander gives a subset of the mission specification to those UAVs which have satisfied the vetting process with respect to capability and credentials. These UAVs may in turn allocate a subset of the mission roles to other UAVs and the whole process finally results in a formation of a management tree which facilitates control and state information collection. Fig. 1 shows examples of role assignment policies. In the first policy the commander authenticates a newly discovered UAV, and assigns it to a surveyor role if it has the required capability with respect to motion and video camera. In the second policy the commander performs reassignment if the failed role type is a surveyor role.

To ensure that the UAVs comprising the SMC perform their tasks correctly, it is important to cope with different types of failures. Consider a mission scenario that contains the following roles: a Commander (C), which has the initial mission specification, assigns roles and manages the SMC; an Aggregator (A), which receives information from surveyors and builds up a map, a Surveyor (S) containing a video camera, and a Relay (R) which maintains communication by relaying messages in an ad-hoc network. Failures in such missions can occur as a result of intermittent or permanent communication link failures as well as individual node failure. A recent study on UAV failures [4] shows that reliability in field environment is only between 6 and 20 hours.

This paper extends the mission management framework from [3] by evaluating the proposed schemes and elaborating the architecture. This architecture uses a management tree (described in Section 3.1) to define management hierarchies as well as data aggregation hierarchies during execution of the mission. If the periodic state information is not received within a specified timeout period, a failure is considered to have occurred. Various timeouts can differentiate between the types of failures and each is handled accordingly.

In conjunction with failure management, we also actively try to maintain communication between team members using two techniques: i) UAVs adapt their movement to always be within radio range of a neighbour or follow each other (similar to [11,13,14,15]) so as to maintain communication by using UAVs as relays to reach distant nodes; ii) The UAVs gather within a defined rendezvous area at a specified time so as to exchange the requisite state information (this is due to the fact that it may be impractical to restrict motion in some situations and so we take a delay tolerant network approach to cater for UAVs being out of communication range for short periods). In the event that a UAV is unable to

\begin{figure}[h]
\centering
\begin{subfigure}{0.45\linewidth}
\begin{verbatim}
policy event: event/newUAV: 
condition: [summary.uav.credential] 
  root/discovery.uav.uav has: #("motion","video") 
  cap: summary.auth.credential] 
action: [uav 
  root/discovery.fullCapReq.uav role:"surveyor"]
\end{verbatim}
\caption{Initial Role Assignment}
\end{subfigure}
\hspace{0.05\linewidth}
\begin{subfigure}{0.45\linewidth}
\begin{verbatim}
policy event: event/UAVFailure: 
condition: [role=="surveyor"] 
action: [role] 
  root/commander.reassign.role 
  scheme:"default"]
\end{verbatim}
\caption{Role Re-Assignment}
\end{subfigure}
\caption{Sample Ponder2 Policies}
\end{figure}