Management Architecture and Mission Specification for Unmanned Autonomous Vehicles

E. Asmare, N. Dulay, H. Kim, E. Lupu, M. Sloman,

Imperial College London, Department of Computing, 180 Queen's Gate, London SW7 2AZ, England

{< e.asmare| n.dulay |h.kim| m.sloman|e.c.lupu >@imperial.ac.uk}

Abstract

This paper proposes the architecture for a Self Managed Cell (SMC) for managing both individual and teams of Unmanned Autonomous Vehicles (UXVs). We outline how to specify a mission in terms of roles for UXVs, the tasks to be performed by a role and the policies for managing tasks. A mission specification also includes role assignment policies for validating credentials and capabilities of a UXV joining the SMC and authorisation policies defining what interactions are permitted between roles or what services a role can access. The concepts are illustrated using a reconnaissance scenario.

Keywords: Collaborative robots, Policy-based robots, Self-management

Introduction

Unmanned Autonomous Vehicles (UXVs) need to adapt their behaviour to current context – location, activity, available resources such as battery power and available services such as quality of communications link. They should be self-managing in that they have to recover or adapt to component failures and optimise performance to best utilise available resources. Additionally, a team of UXVs should cooperate to achieve a particular mission such as surveillance of a specific area or search for specific targets.

We propose using the concept of a Self Managed Cell (SMC) [1] as the general architectural principle for realizing self management of both individual UXVs and teams of UXVs. The SMC is an approach to autonomic computing [2] that emphasises the use of policy to support adaptive self-management. In our approach, collaborating UXVs form a SMC, where the SMC is responsible for self-configuration, as UXVs may join or leave the SMC; self-optimisation in order to achieve the mission with minimum delay or to minimise use of resources; self-healing to recover from failures and self-protecting in that hostile entities may try to subvert the mission by masquerading as members of the cell or mounting denial of service attacks to try to deplete resources or prevent communications.

A UXV may be composed of various sensors for vision, sound, vibration, chemical detection, location as well as supporting communication links. Usually, not all capabilities are available in a single vehicle and so one UXV may provide services to others in the team. Multiple sub-teams of UXVs may also collaborate to achieve an overall mission and so our concept of a SMC permits the composition of SMCs to form larger more complex SMCs and to allow for peer-to-peer relations between SMC. Note that in our architecture, the body-area-network of a soldier, together with his communications facilities, may also form a SMC and permit him to control the overall team of UXVs forming a larger SMC.

A UXV will have a capability specification that describes its resources and the services it can perform. A commander, which could be a human or another UXV, issues a mission

specification to be undertaken by a set of UXVs. Based on the mission specification and the capability specification of the UXV, it will be assigned to perform specific roles within the team which cooperates to perform the assigned mission

In the following section we present the SMC common architectural pattern and its overall functionality. We then present a policy-based mission specification and elaborate on the approach using an example scenario followed by brief discussions of communication and security issues associated with UXV missions. Finally a brief summary of related work is presented.

The Self Managed Cell Concept and Architecture

An SMC is a closed-loop systematic organization of management services which represent a set of hardware and application components in a single device, or multiple devices collaborating to achieve a common goal. The management services interact with each other through an asynchronous event bus. As a minimum an SMC should contain a measurement and event service, and a policy service [1] which are required in order to implement a feed-back loop adaptation mechanism.

The basic architecture of an SMC is shown in Figure 1. We will briefly describe the function of each core service.

The *Policy Service* is the means of specifying the adaptive behaviour of an SMC. Policies [3] are rules governing choices in behaviour and we focus on two main types of policies namely obligation and authorization. Obligations are event-condition-action rules and authorisation policies define what actions a subject can perform on a target resource or service. Policies can be dynamically modified to adapt behaviour.

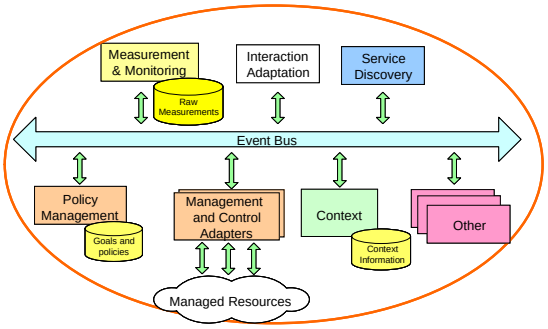

Figure 1: Architecture of a SMC

The *Discovery Service* discovers components which are in range and capable of being members of the SMC. For example, there may be some UXVs from another mission or belonging to an ally that can provide specific services or aid in accomplishing the goals of the SMC team. There are obviously issues relating to authentication when discovering new UXVs as enemy UXVs must not be allowed to join the SMC. We also have to cater for the fact that UXVs may temporarily been out of wireless contact, for example when behind an obstruction, but then come back into range, so have not left the SMC.

The *Event Bus* is responsible for asynchronous notification of events to different management services of the SMC. Event notification is a crucial element of an SMC because adaptation, protection and other self-management actions are specified in terms of obligation policies triggered by events. An obligation policy may perform an action which modifies the behaviour of a single UXV or it may enable or disable other policies to change the overall behaviour of the SMC. The use of an event bus also allows maintaining loose coupling between the various SMC services which can react concurrently and independently to event notifications. However we do not constrain all communication to be over the event bus e.g. video sent from a camera would not make use of the event bus.

The *Measurement and Monitoring Service* is responsible for keeping track of the SMC's operation and generates events when actions need to be taken. Events may indicate degradation or failure of components, detection of specific landmarks or possible 'unidentified' entities etc.

The *Context Service* uses sensors to determine information such as current location, weather conditions, what other SMCs or other entities are in the vicinity as the SMC behaviour will adapt to current context.

Roles provide a means of grouping components in the cell according to their capabilities and correspond to the *role* they play within the cell. For example UXVs with satellite transceivers will be assigned to a long-range-communications role; UXVs with short range transceivers will be assigned to an ad-hoc-network role; UXVs with video cameras would be assigned to a vision role, airborne UXVs would be assigned to an air-surveillance role and officer-SMC would be assigned to a cell controller role.

A SMC is a generic concept which can be used to accomplish specific missions. The mission specification identifies specific required roles, tasks to be performed or services to be provided by the components assigned to the required roles, policies related to the roles etc. These concepts will be elaborated in terms of a specific scenario outlined below.

Reconnaissance Scenario

In this section, we illustrate our approach using a UXV-mission scenario based on the *Urban reconnaissance* scenario specified in [4]. In this scenario there are four UXVs and the objective is to collect data regarding the layout and contents of a house to determine whether it is safe for humans to move in to the house or not. It is suspected that there may be hazardous chemical, biological materials or explosives in the house. There is also a possibility that persons may still be present in the house which may attack or sabotage the UXVs.

We assume the UXVs include functionality such as overland movement and propulsion, autonomous navigation, adaptive mission planning, sensing (e.g. infrared, explosive material detection, radiation or chemical detection), target recognition or identification and situation assessment. The UXVs have a data link among themselves and to a command vehicle which has high-speed long-range communication and more sophisticated processing and data storage capabilities.

Roles and Mission Specification

We can identify the following roles in this scenario.

Commander: this is a manned vehicle with a range of communications equipment.

Surveyors: the surveyors divide the volume of the house among them and produce a map and video of their space. They send the collected data to the aggregator which relays it to the commander.

Aggregator: produces a map of the whole space and distributes it back to the surveyors so that they can use it for the remaining reconnaissance. It will also send all the aggregated information to the command centre.

The roles act as 'placeholders' for the purpose of specifying the required behaviour of the UXVs assigned to roles. This behaviour is specified in the form of policies which define the tasks which the UXV will perform and privileges with respect to accessing services provided by other roles or shared resources within the SMC.

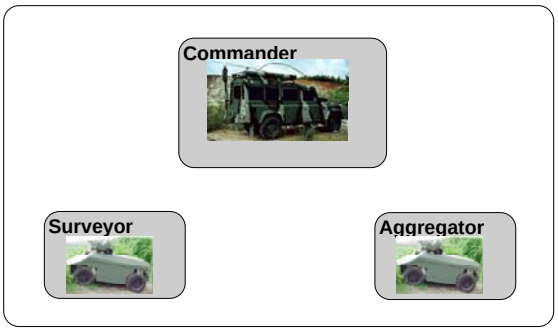

Figure 2: Mission SMC

Figure 2 shows the main roles for the scenario forming a Mission SMC. This SMC can be formed on the commander UXV. This SMC will issue a mission specification which will be interpreted by the involved roles. An outline mission specification defining the roles and tasks is given in Figure 3.

There is only one command vehicle in this mission and as it has the required capabilities, it is assigned to the commander role so it acts as the mission or SMC controller. However the UXVs to be assigned to surveyor and aggregator roles may be discovered later as they may come into radio range only when the commander is near the surveillance area. A discovered UXV would first have to be authenticated and its credentials (certificates) verified in order to ensure that the UXV belongs to an allied force. If it has the required video and sensors it would be assigned to a surveyor role and if it had the required processing and storage it would be assigned to an aggregator role.

1. **oblig on** discovered (uxv, credentials, resources)
   do /smc/roles/commander.assign (uxv)
   when authenticate (credentials) **and**
   resources.comms = “longRange”

2. **oblig on** discovered (uxv, credentials, resources)
   do /smc/roles/surveyor.assign (uxv)
   when authenticate (credentials) **and**
   resources.sensor1 = “video” **and**
   resources.sensor2 = “chemical”

3. **oblig on** discovered (uxv, credentials, resources)
   do /smc/roles/aggregator.assign (uxv)
   when authenticate (credentials) **and**
   resources. process > medium

Mission shared knowledge base
- Certificates and encryption keys needed for security functions
- Preloaded maps
- Overall mission constraints e.g. max. vehicle distance for ad-hoc networking
- Time Constraints

Role types
- **Role** commander
  - **Role assignment policy** - what resources or certificates are needed
  - **Tasks**
    - Report to headquarters
      - **Policies** related to reporting to headquarters
    - Manage aggregators
      - **Policies** related to managing aggregators
    - Manage surveyors
      - **Policies** related to managing surveyors
  - **Authorisation policies** needed for interaction with other roles
- **Role** aggregators
  - **Role assignment policy** - what resources or certificates are needed
  - **Tasks**
    - Movement tasks
      - **Policies** related to movement
    - Collect map data from surveyors
      - **Policies** related to collecting map data
    - Aggregate maps
      - **Policies** related to aggregating maps
    - Send map data to commander
      - **Policies** sending map data to commander
  - **Authorisation policies** needed for interaction with other roles
- **Role** Surveyors
  - **Role assignment policy** - what resources or certificates are needed
  - **Tasks**
    - Movement tasks
      - **Policies** related to movement
    - Setting survey boundaries in cooperation with other surveyors
      - **Policies** related to setting survey boundaries
    - Capture video
      - **Policies** related to managing video camera
    - Draw map
      - **Policies** related to generating maps
    - Detect hazardous chemicals
      - **Policies** related to detecting explosives and indicating on map
    - Transmit map & video to aggregator
      - **Policies** for sending map and video to aggregator
  - **Authorisation policies** needed for interaction with other roles

Figure 3: Mission Specification

The above policies specify how a newly discovered UXV can be assigned to the appropriate role, after the credentials have been successfully verified, based on the UXV’s profile including its resources and capabilities. Note that encoding the role assignment as policies enables us to change the strategy of this assignment during the mission without interrupting its functioning.

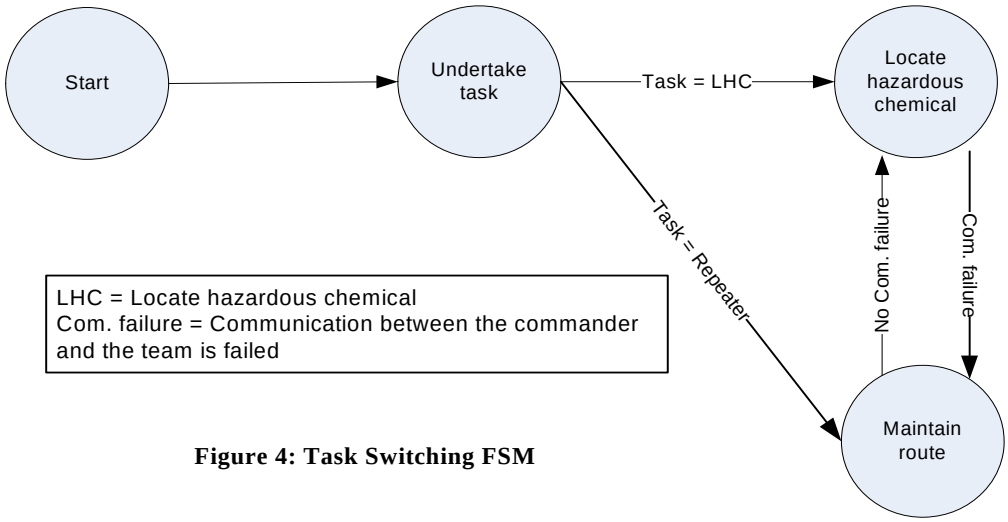

Figure 4: Task Switching FSM

When assigned to the relevant role, the initial tasks and policies relating to that role specified in the mission will be loaded into the relevant UXV. During the course of the mission, the commander may download new tasks and policies into UXVs or just enable or disable some that have been preloaded. This is the means for adapting the behaviour of the UXVs within a role. For example the commander may activate relay-mode communications for the surveyors if some seem to be going out of range.

4. **oblig on** range_failure **do** /smc/roles/surveyor/*.enable(relaymode)

Although the approach to planning depends on the specific type of robot software architecture used in the UXV, assuming a hybrid architecture [5, 6] that has a higher level deliberative part which specifies the sequence of tasks and a lower level reactive [7] part which contains groups of primitive behaviours such as moving to a certain point, we are investigating the possibility of using policies to specify plans (sequences of tasks) and behavioural parameters.

For instance when a UXV assumes a surveyor role, its corresponding plan may be expressed in the Finite State Machine (FSM) shown in Figure 4. The UXV will either perform one of the tasks, i.e., either locating hazardous chemicals or maintaining communication (by serving as a repeater) depending on the task it undertakes. Whichever task it takes it might switch between the two tasks based on events. It is possible to specify this plan using policies as follows:

oblig on task(LHC) **do** /smc/tasks.init(LHC)

oblig on task(Repeater) **do** /smc/tasks.init(Maintain_communication)

oblig on communication_failure **do** /smc/tasks.switchto(Maintain_communication)

oblig on no_communication_failure **do** /smc/tasks.switchto(LHC)

Each task can further be specified using an

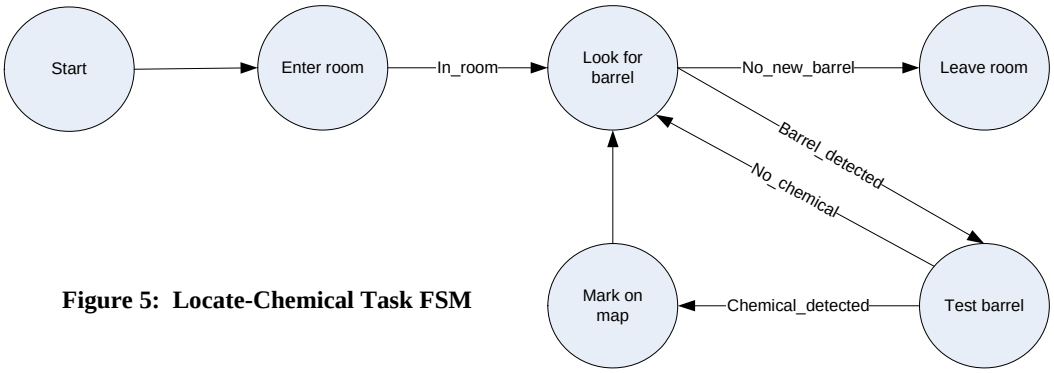

Figure 5: Locate-Chemical Task FSM

FSM; if we consider the task of locating hazardous chemicals, the plan to achieve this might be expressed in the Finite State Machine (FSM) shown in Figure 5.

Again, we can specify this plan using policies using a similar approach as shown before. Finally the subtask in each task, i.e., the behaviours or assemblage of behaviours [7, 8] can be adapted to the current environment by using policies to specify behavioural parameters.

For instance, let us consider one of the assemblage of behaviours - 'entering a room', this assemblage can roughly be treated as a superposition of a set of primitive behaviours such as Moving to a goal, Wandering, Avoiding obstacles, and Biasing a move [9].

Superposing these primitive behaviours to produce a vector, which will control the motor, involves using behavioural parameters. The optimal values for these parameters differ in different environments and context. Policies can be used to adapt primitive behaviours by changing the behavioural parameters based on context.

Authorisation policies specify how the roles are permitted to interact e.g.:

5. **auth+** /smc/roles/surveyor -> /smc/role/aggregator.sendmap()

6. **auth+** /smc/roles/commander -> /smc/role/surveyor.enable()

Policy 5 permits any UXV in the surveyor role to invoke sendmap on a UXV in aggregator role and policy 6 permits the commander to enable operation modes on surveyors.

Communication

We are investigating the network management issues related to autonomous vehicle teams. Autonomous vehicles need to have continuous and quality (based on the mission type) communication through out a mission. Hence network management is an important part of an autonomous vehicle system.

The network management framework should address at least three main issues:

- Network setup among vehicles performing a mission (likely an ad hoc net work with a connection to a backbone). This network will most likely be wireless LAN.

- Network maintenance: this issue is special in autonomous vehicles because the network nodes move and we want the movement of nodes to be in such a way that the nodes are always within a wireless link range. This needs a distributed algorithm to control movement (formation) based on the wireless link signal strength among vehicles (nodes). Because digital signals suffer severe attenuation, the data rate of a wireless link decreases considerably with distance.

- Network performance monitoring.

Security Consideration

We are investigating the security issues related to authentication, authorisation and how to isolate faulty or malicious UXVs which might subvert the mission.

We assume that UXVs are capable of performing asymmetric key based cryptography. We will investigate the use of signed credential certificates for authenticating identity, public keys and organisation membership of UXVs. This requires the ability to preload a set of public keys and some secret keys onto a UXV. These can be used to securely distribute keys for communication encryption and SMC membership keys. It is likely that we will need both SMC keys for encrypting communication between all members of a SMC as well as role specific keys. For example, a SMC belonging to an ally may not be fully trusted so should only be able to decrypt the interactions related to the role it performs. Key management protocols are needed for changing keys when UXVs leave the cell.

Assigning UXVs to roles allows authorisation to be specified in terms of roles – a form of Role based access control. However we may need some sub-roles for UXVs belonging to allies or other organisations and hence are not fully trusted. They would then only have access to a subset of the resources and services provided by the SMC.

We also have to consider that a UXV may be captured or, as a result of a fault, start behaving anomalously. It is necessary to be able to detect anomalous behaviour, take action to isolate the anomalous UXV from the SMC and possibly distribute new keys to other SMC members.

Related Work

Kephert et al. [2] define an Autonomic Element as a component that is responsible for managing its behaviour and interaction in accordance with policies. The SMC concept is thus an autonomic element which realizes policy-based self-management. Although most of the current work on Autonomic Computing focuses on large scale web servers and cluster computing, they also identify policy as the key to adaptive behaviour.

Keoh et al. [11] have proposed an approach to establish and manage mobile ad-hoc-networks using policies. They introduced a community specification called doctrine which defines the roles of the participants in the community, the characteristics that participants must exhibit in order to be eligible to play a role, as well as the policies governing their behaviour within the community. The mission specification we are intending to use is in some ways similar to doctrine. But because our specification also involves tasks, it will noticeably differ from doctrine.

Likhachev et al. [9] have proposed an approach to automatic modification of behavioural assemblage parameters for autonomous navigation tasks. Their approach is based on Artificial Intelligence in that it uses case based reasoning. We try to solve the same problem using a different approach, i.e., policy.

References

[1] N. Dulay, S. Heeps, E. Lupu, R. Mathur, O. Sharma, M. Sloman, J. Sventek, "AMUSE: Autonomic Management of Ubiquitous e-Health Systems", 19 - 22 Sep. 2005, Proceedings of the UK e-Science All Hands Meeting, Nottingham UK.

[2] J. O. Kephart, D. M. Chess, "The Vision of Autonomic Computing", 2003, IEEE Computer, vol. 36, pp. 41-50.

[3] N. Damianou, N. Dulay, E. Lupu, and M. Sloman, "The Ponder Policy Specification Language", 29-31 Jan. 2001, Proceedings of Policy 2001, Workshop on Policies for Distributed Systems and Networks, Bristol, UK, Springer-Verlag LNCS 1995, pp. 18-39.

[4] SEAS DTC – Study Vignettes for Demonstrations, REPORT NO – BAES-ASNB-OA-DTC-RP-00696, Issue – Draft A 27th January 2006

[5] G. A. Bekey, "Autonomous Robots: From Biological Inspiration to Implementation and Control", 2005, First edition: The MIT Press.

[6] R. R. Murphy, "Introduction to AI Robotics", 2000, Cambridge, MA, USA: MIT Press.

[7] R. C. Arkin, "A Behaviour-based Robotics", 1998, Cambridge, MA, USA: MIT Press.

[8] D. C. MacKenzie, R. C. Arkin, and J. M. Cameron, "Multiagent Mission Specification and Execution", 1997, Autonomous Robots, vol. 4, no1., pp. 29-52.

[9] M. Likhachev, M. Kaess, Z. Kira, and R. C. Arkin, "Spatio-Temporal

Case-Based Reasoning for Efficient Reactive Robot Navigation", 2005, http://www-static.cc.gatech.edu/ai/robot-lab/online-publications/LikhachevEtAl2005.pdf

[10] S. R. White, J. E. Hanson, I. Whalley, D. M. chess, and O. K. Jeffrey, "An architectural approach to autonomic computing", 2004, Proceedings of First International Conference on Autonomic Computing (ICAC 04), pp. 2-9

[11] S. L. Keoh, E. Lupu, and M. Sloman, "PEACE: A Policy-Based Establishment of Ad-hoc Communities", 2004, Proceedings 20th Annual Computer Security Applications Conference, ACSAC '04 Tucson Arizona USA, pp386-395

Acknowledgements

The work reported in this paper was funded by the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre established by the UK Ministry of Defence.