

# INFRAESTRUCTURA PARA ESTUDIAR ADAPTABILIDAD Y TRANSPARENCIA EN EL CENTRO DE CONTROL VERSÁTIL

Juan A. Bonache Seco, Jose A. López Orozco, Eva Besada Portas, Jesús M. de la Cruz  
jabonache@ucm.es, jalo@ucm.es, ebesada@ucm.es, jmcruz@ucm.es  
Universidad Complutense de Madrid

## Resumen

*En la actualidad, la proliferación de vehículos autónomos o tripulados de forma remota involucrados en misiones complejas que requieren cooperación entre ellos y, en algunos casos, la inclusión de vehículos heterogéneos en una misión, está ocasionando una evolución en las estaciones de control de tierra. Éstas, además de mostrar los datos de forma clara, deben contar con una serie de mecanismos que faciliten la labor del operador mitigando su carga de trabajo y nivel de estrés.*

*En este trabajo, se describe un Centro de Control Versátil (CCV) que cuenta con una arquitectura distribuida y la infraestructura necesaria para implementar y experimentar los conceptos de Adaptabilidad (que permite al interfaz gráfico reubicar y alterar el aspecto de los elementos gráficos que se muestran en pantalla) y Transparencia (que permite reducir el nivel de estrés del operador disminuyendo el número de elementos en los que tiene que centrar su atención).*

**Palabras clave:** Estación de Control de Tierra (Ground Control Station - GCS), Centro de Control Versátil, Monitorización y Supervisión, Adaptabilidad, Transparencia, Vehículos Autónomos (UAV, USV, UGV)

## 1. INTRODUCCIÓN

El ámbito de aplicación de los vehículos autónomos o no tripulados crece diariamente, realizando cada vez tareas más diversas. En algunas ocasiones, dichas tareas pueden ser realizadas por un sólo vehículo [1, 2, 3]. En otras alcanzan un nivel de complejidad que obliga a dividir las sub-tareas que se asignan a un equipo cooperativo de vehículos que puede ser: homogéneo, en el caso de que todos los vehículos implicados sean de la misma naturaleza (aérea, marítima o terrestre) o heterogéneo [4, 5]. Este tipo de misiones suele implicar el desarrollo de un software para la estación de control de tierra (Ground Control Station - GCS) muy específico, en el que su modificación para otros tipos de misión o un cambio en el equi-

po de vehículos puede resultar muy costosa.

Podemos encontrar múltiples ejemplos de centros de control tanto en ámbito militar como civil, por ejemplo la Estación de Control de Tierra para Múltiples Entidades de Combate [6], el entorno ASMAC para múltiples vehículos submarinos [7], el sistema de vigilancia AMFIS capaz de fusionar y presentar datos de sensores de vehículos aéreos y terrestres [8], el Centro de Control para entornos marinos diseñado en el proyecto europeo MUNIN [9] y el Centro de Control inmersivo que incorpora nuevas tecnologías como la realidad virtual [10].

Además, si nos centramos en la monitorización y el control, cuanto mayor es la complejidad de la misión y el número de vehículos que participan, más alto es el número de datos que el operador debe visualizar, lo que acarrea un aumento del nivel de estrés y un descenso en la eficiencia del operador. Este problema se ha mitigado, por un lado, aplicando el concepto de Transparencia [11, 12], que consiste en ocultar elementos visuales que se consideran de menor importancia con el objetivo de liberar parte de la carga del usuario. Por otro, para potenciar la ergonomía de los elementos gráficos (posición, tamaño, color, etc.) se utiliza el concepto de Adaptabilidad [13, 14, 15, 16], consistente en modificar (adaptar) los elementos gráficos en tiempo de ejecución para su mejor comprensión por parte del usuario. Mejorar estos aspectos puede ser clave, ya que en los casos que exigen una rápida respuesta (p.e. emergencias) o a la hora de definir las misiones a realizar, la intervención de un humano cerrando el lazo de control [17] puede mejorar el experimento hasta un 50 %.

En la sección 2 de este artículo describiremos un Centro de Control Versátil (CCV) [18] con una arquitectura distribuida que permitirá: 1) reconfigurar la información que se muestra y cómo mostrarla (para cada vehículo y operador), 2) definir alarmas que se activen ante determinados eventos liberando de carga de trabajo al operador mientras dichos eventos no ocurran y 3) reasignar en tiempo real la gestión de los vehículos, pudiendo transferir el control de un operador a otro si se dan las condiciones adecuadas. Además se ha diseñado la infraestructura necesaria implemen-

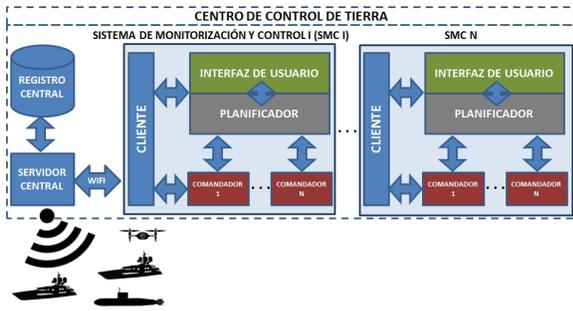


Figura 1: Arquitectura Distribuida del CCV

tar dos mecanismos adicionales: Adaptabilidad y Transparencia[19] que serán detallados en la sección 3. Finalizaremos con las conclusiones y trabajo futuro en la sección 4.

## 2. ARQUITECTURA DEL CENTRO DE CONTROL VERSÁTIL

Durante el desarrollo del proyecto SALACOM (Sistema Autónomo de Localización y Actuación ante Contaminantes en el Mar), se ha desarrollado un Centro de Control Versátil (CCV) que debe tener unas características muy concretas. Ha de ser fácilmente configurable para diferente tipo y número de vehículos (independientemente de su hardware y protocolos de comunicaciones), permitir que los vehículos trabajen de forma autónoma e incluir al operador humano en el lazo de control para tareas de supervisión. También debe ser capaz de modificar en tiempo real algunas de sus características como la visualización de los datos en base a criterios ergonómicos, preferencias del operador o tipo de misión, coordinación/control de vehículos y formaciones de los mismos, definición de maniobras y reorganización de recursos.

Estas características se logran mediante una arquitectura distribuida [18] que puede verse en la Figura 1 y que consta de cuatro módulos:

*Servicio de Comunicaciones.* Es el módulo encargado de gestionar todas las comunicaciones (datos de telemetría, consignas, operaciones de alto nivel, etc.) entre cualquiera de los módulos del centro de control y hacia los vehículos, por lo que es el único de los módulos con comunicación directa con los mismos. Se divide en tres componentes: Servidor Central, Cliente/Clientes y Registro Central.

*Planificador.* Es el módulo encargado de generar comandos de alto nivel para que los vehículos realicen trayectorias o misiones, que podrán diseñarse desde el entorno gráfico propio del Planificador o

desde el Interfaz de Usuario mediante la definición de puntos de paso o rutas predefinidas. Posteriormente estas trayectorias serán enviadas a los vehículos a través del Comandador (que generará las instrucciones de bajo nivel propias de cada vehículo) pasando por el Servicio de Comunicaciones. Además podrá calcular la ruta más adecuada en base a los puntos señalados por el operador en el mapa y generar trayectorias mediante curvas paramétricas (lemniscata, elipse, círculo, seno, etc.) que serán seguidas por el control de los vehículos utilizando diferentes métodos de control y guiado [20, 21, 22, 23, 24]. También hará reaccionar al vehículo ante alarmas recibidas de los vehículos.

*Comandador.* Es el módulo más próximo al vehículo y es propio a él (sólo habrá uno activo por vehículo). Su interfaz es específica para cada vehículo. Mostrará sus alarmas y banderas, que podrán habilitarse y deshabilitarse desde este módulo. También permite maniobras de ingeniería para comprobar el correcto funcionamiento del vehículo. Procesa las órdenes del Planificador o Interfaz de Usuario generando los comandos de bajo nivel y trasladándolas al Servicio de Comunicaciones para que las envíe al vehículo. Además deberá enviar una señal periódica que le indica al vehículo que hay un comandador activo para que no interrumpa su actividad.

*Interfaz de Usuario.* Permite al usuario visualizar el estado, situación en el mapa y datos de telemetría de cada uno de los vehículos involucrados en la misión e interactuar con otros módulos por medio de botones y menús. Permitirá al usuario llevar a cabo acciones como solicitar/renunciar el control de uno o más vehículos que este comandando, establecer rutas y puntos de acceso sobre el mapa (o elegir un fichero desde el que desee cargarlos), previsualizar dichas rutas y realizar algunos cambios sobre la disposición de los elementos para que se ajusten a los criterios ergonómicos del operador. Además permite al usuario definir alarmas tipo notificación.

## 3. ADAPTABILIDAD Y TRANSPARENCIA

Con el objetivo de mitigar la carga de trabajo y el estrés soportado por un operador que debe supervisar/controlar varios vehículos heterogéneos, se han implementado en el CCV dos mecanismos que ayudaran a paliar en la medida de lo posible los efectos negativos del exceso de información y el esfuerzo mental del operador. Por un lado, mediante la Adaptabilidad, se transformará el interfaz para dar prioridad a los elementos gráficos de mayor importancia y mostrarlos de forma más intuitiva y ergonómica. Por otro lado, gracias a la Transpa-

rencia se eliminarán de la interfaz (de forma momentánea o permanente, dependiendo de las necesidades de cada misión) los elementos gráficos menos relevantes, reduciendo así la carga de trabajo del operador.

### 3.1. ADAPTABILIDAD

Un software es Adaptativo cuando es capaz de cambiar en tiempo de ejecución algunas de sus características, tratando de mejorar una o varias facetas del trabajo para el que es utilizado, mejorando su propio funcionamiento u orientando dichos cambios a facilitar a un tercero (un usuario, otro programa con el que deba interactuar, etc.) las herramientas necesarias que le permitan realizar su labor de forma más eficiente.

#### 3.1.1. ANTECEDENTES

Garlan et al. muestran como su framework adaptativo, RAINBOW [13], resulta útil en diferentes ámbitos de aplicación. Para ilustrarlo, describen un servicio web tipo cliente-servidor que debe adaptar los recursos disponibles (número de servidores y ancho de banda) ante la petición de contenido de los clientes para conseguir mejorar la experiencia de los usuarios. Otro ejemplo de que el framework puede aplicarse a problemas diferentes es un sistema de videoconferencia que implementa la misma estructura de módulos o capas pero que varía su comportamiento por medio de estrategias o algoritmos implementados en cada uno de los módulos que gobiernan el software.

Otro ejemplo de aplicación de Adaptabilidad bastante extendido es la distribución de recursos en redes inalámbricas enfocado a la mejora en la calidad del servicio (QoS - Quality of Service). Uno de los ejemplos que podemos encontrar es BAA (Bandwidth Adaptation Algorithm - Algoritmo de Adaptación de Ancho de Banda [25]), que expande o reduce dinámicamente el ancho de banda asignado a un bloque de llamadas entrantes cuando se produce algún cambio que lo invoca (llamada entrante, finalizada, rechazada, etc.). Otro ejemplo es el Sistema Adaptativo de Gestión de Calidad del Servicio (QoS) propuesto por Huang et al. [26] que distribuye dinámicamente los recursos disponibles en un entorno de comunicaciones inalámbrico y provee a los usuarios de servicios multimedia con una calidad que cumple con los requisitos especificados.

#### 3.1.2. IMPLEMENTACIÓN EN EL CCV

En la sección anterior se han mostrado algunos problemas genéricos que han sido resueltos por medio de Adaptabilidad. Veamos a continuación

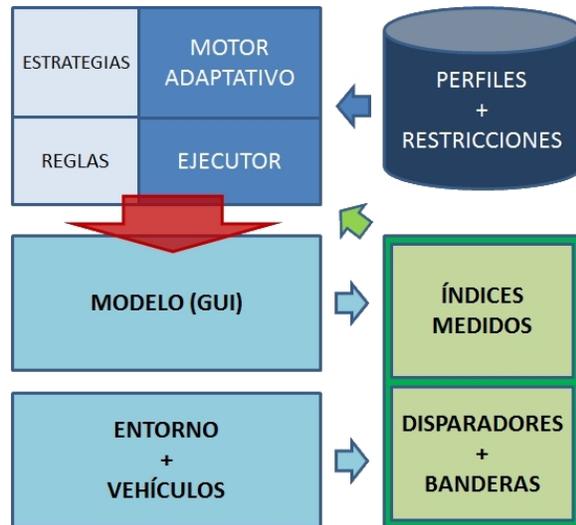


Figura 2: Arquitectura para Adaptabilidad del CCV

cómo podemos aplicarla al CCV para resolver algunos de los problemas planteados previamente.

Para ello se ha diseñado una infraestructura modular que permite a nuestro software modificarse dinámicamente para mejorar la eficiencia del operador. Los cambios se centran en dos vertientes, por un lado en mostrar los elementos necesarios dadas unas restricciones de espacio (como trataremos en la sección posterior dedicada a Transparencia) y por otro en mostrarlos por pantalla con una apariencia que vaya en consonancia con su importancia para el desarrollo de la misión (modificando posición, color y tamaño). Intuitivamente podemos establecer el paralelismo entre nuestras restricciones (espacio e importancia en la misión) y algunas de las mencionadas en ejemplos anteriores (por ejemplo ancho de banda y prioridad).

En muchos de los trabajos estudiados, se cuenta con una estructura similar que difiere en algunos bloques [27, 26], pero para el diseño del software del CCV se ha optado por una infraestructura semejante a la utilizada en RAINBOW [13, 16, 14, 15]. Dicha estructura, como podemos ver en la Figura 2, consta de cuatro bloques:

*La Base de Datos (BD) de Perfiles y Restricciones.* Se trata de un almacenamiento persistente donde se guardarán datos referentes al perfil de usuario del operador, que nos ayudarán a decidir la mejor estrategia para adaptar los elementos del Interfaz Gráfico. También se almacenan ciertas Restricciones del software referentes al Modelo, que hace referencia a la estructura del Interfaz Gráfico y cómo pueden alterarse los elementos en él.

*El Bloque de Parámetros.* Es un módulo que se encarga de tomar medidas sobre algunos índices re-

levantes referentes a la relación entre el Interfaz de Usuario y el Operador (tiempo de reacción, estrés, preferencias gráficas modificables, etc.) y otros disparadores o banderas referentes al entorno (viento, corrientes marítimas, etc.) o a los vehículos (nivel de batería, datos de telemetría, status de la misión, etc.) que serán externos al software del CCV. Estos datos son recibidos por el Motor Adaptativo junto a los datos de perfil y restricciones almacenados en la BD y sirven como condiciones para disparar el mecanismo de Adaptabilidad y como parámetros para elegir la mejor estrategia o algoritmo a aplicar en la adaptación.

El *Modelo*. Es la representación del Interfaz Gráfico y las reglas sobre cómo pueden mostrarse todos los elementos en su interior. Para ello, en el Interfaz de Usuario se cuenta con un bloque para elementos gráficos por cada vehículo dentro del cual se reordenarán y cambiarán de apariencia en base a los datos recibidos del Bloque de Parámetros. También encontramos información sobre los vehículos autónomos que forman parte de la misión y algunos datos relevantes sobre el entorno en el que se va a actuar.

El *Motor Adaptativo*. Es el bloque más importante ya que se encarga de almacenar las Estrategias o algoritmos (pueden estar formadas por subestrategias o Tácticas si éstas son complejas y es conveniente dividir las) y las reglas en base a las cuales el bloque Ejecutor adaptará los elementos gráficos del Modelo.

### 3.1.3. FUNCIONAMIENTO

A continuación describiremos un ejemplo de funcionamiento del mecanismo de Adaptabilidad aplicado a una de las maniobras de seguimiento realizadas por los barcos del proyecto SALACOM, en la que el barco líder (en azul) traza una trayectoria programada y el barco seguidor (en rojo) debe escoltarle a una distancia fijada previamente.

Se parte, en un instante dado de la misión, de la configuración de la Figura 3. Podemos ver los bloques de elementos pertenecientes a cada barco, el líder (izquierda, con borde azul) y el seguidor (derecha, borde rojo). Para este experimento se ha determinado que las posiciones de más peso son las superiores y los elementos se irán colocando hacia abajo cuanto menor sea su prioridad. Las posiciones de mayor prioridad pueden definirse en el software para cada operador, basándonos en sus preferencias personales y criterios ergonómicos almacenados en su perfil dentro de la *BD de Perfiles y Restricciones*.

En la Figura 3 se observa que la velocidad de ambos barcos se sitúa en un segundo plano ya que es

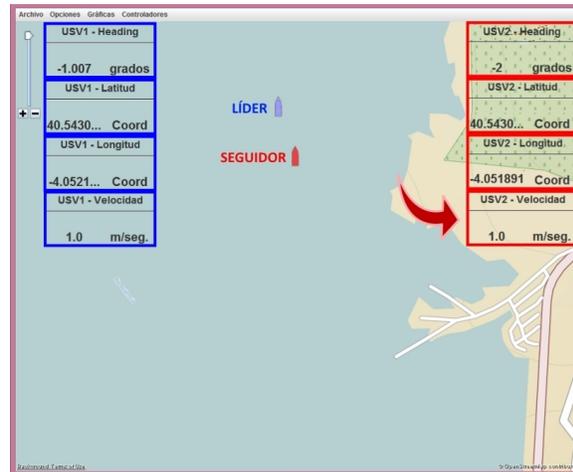


Figura 3: Funcionamiento de Adaptabilidad 1

constante durante este tramo de la maniobra, que dura varios minutos. Al cabo de este tiempo, el barco seguidor aumenta la distancia programada respecto al líder, lo que activa en su controlador un aumento de la velocidad para reestablecer la distancia correcta. El aumento de velocidad, activa un disparador en el *Bloque de Parámetros*, inmediatamente transmitido al *Motor Adaptativo* que, por medio de su bloque *Ejecutor* transmite el cambio al *Modelo* o Interfaz Gráfico.



Figura 4: Funcionamiento de Adaptabilidad 2

En la Figura 4 vemos que el elemento gráfico que representa la velocidad del barco seguidor ha escalado puestos hacia una posición de mayor prioridad. Esto sucede porque el disparador y los datos de telemetría, enviados al *Motor Adaptativo* le han indicado que un dato que anteriormente permanecía constante (la velocidad) ha sufrido un cambio brusco, por lo que se considera importante que el operador sea consciente de dicho cambio y en consecuencia el elemento debe moverse a una posición en la que lo pueda ver mejor. Adicional-

mente, si llegase a un valor considerado crítico, se dispararía también una Alarma visual y/o auditiva para atraer definitivamente la atención del usuario.

### 3.2. TRANSPARENCIA

La Transparencia aplicada al software se basa en la capacidad del programa para mostrar menos información haciendo el proceso más “transparente” al usuario para que su labor sea más rápida y fácil, pero sin que el nivel de confianza que tiene en el programa se vea afectada, ya que lo normal es que se tenga más confianza cuanto más información esté disponible.

#### 3.2.1. ANTECEDENTES

Los Centros de Control de Tierra deben mostrar una buena cantidad de datos pertenecientes a los vehículos monitorizados, a la misión que están realizando y al entorno en el que dicha misión se está desarrollando. Cuando las misiones implican equipos de vehículos, los datos a mostrar crecen prácticamente de forma exponencial y éstos son más complicados de interpretar por parte del operador si los vehículos son de naturaleza heterogénea. Debemos plantearnos, pues, varias cuestiones como: si el espacio en pantalla es suficiente para mostrar todos los datos monitorizados y, suponiendo que lo fuese, si es realmente relevante tener disponibles todos estos datos de forma continua.

El objetivo es, por lo tanto, lograr un equilibrio mostrando los datos suficientes para que el operador conozca todos los detalles relevantes sobre la misión y los vehículos, pero sin que lleguen a provocar un aumento considerable en el nivel de carga de trabajo o de estrés sobre el operador. A esto, debemos añadirle el *nivel de confianza* del usuario en el software, que puede disminuir si se eliminan demasiados datos de su campo de visión, ya que si el usuario no puede determinar rápidamente en qué estado está un vehículo en un momento dado, puede aumentar su nivel de incertidumbre acerca de la correcta realización de la misión. En esta línea, Mercado et al. [12] realizan un estudio sobre un grupo heterogéneo de operadores sobre el que se mide: rendimiento, carga de trabajo y nivel de confianza en el software, alternando entre 3 niveles de transparencia: N1, en el que sólo se muestra información básica sobre los vehículos y el camino que seguirán durante la misión, N2 que mostrará la información del N1 y algunos datos sobre las decisiones de la Inteligencia Artificial (IA) del software y N3 que mostrará la información de los niveles anteriores y algunos datos sobre aspectos de la misión que pueden producir incertidumbre. De este estudio se extrae que el rendimiento del ope-

rador y su nivel de confianza aumentan cuantos más datos se muestran, mientras que el estrés y la carga de trabajo no aumentan significativamente.

Sin embargo, Mercado et al. basan su estudio en un software de planificación que ayuda a tomar decisiones sobre la misión al operador, mientras que nuestro problema se basa en las dificultades que acarrea la monitorización y control de un equipo cooperativo de vehículos autónomos. En este tipo de misiones, por nuestra experiencia en experimentos de campo con los barcos del proyecto SALACOM en el pantano de El Atazar, hemos comprobado que en ocasiones el operador se ve sobrepasado por la cantidad de datos a monitorizar y, en caso de ser necesario el envío de consignas a los vehículos en momentos puntuales, resulta complicado si hay que hacerlo a más de uno simultáneamente.

#### 3.2.2. IMPLEMENTACIÓN EN EL CCV

Debido a las razones descritas anteriormente, se ha decidido implementar en el CCV dos mecanismos con objeto de mitigar el exceso de carga de trabajo y el estrés que puede acarrear para el operador la concurrencia de varios vehículos cooperando en una misión.

El primero consiste en la solución más intuitiva, delegar tareas. Aprovechando la arquitectura distribuida descrita en la sección 2, se dividirán las tareas de supervisión y comando de los vehículos en tiempo de ejecución. De esta forma el software gana una gran flexibilidad permitiendo al operador transferir a otros operadores las tareas que no se sienta capaz de realizar en un momento dado. Para ello el operador debe indicarle al CCV que desea delegar una tarea, por ejemplo el control de uno de los barcos, por medio del menú de opciones del *Interfaz Gráfico*, que le mandará la orden al *Servicio de Comunicaciones*, donde se comprobará que la transferencia puede realizarse sin peligro, revisando la disponibilidad y otras restricciones en el *Servidor Central*. Una vez comprobado que se cumplen todos los requisitos, se enviará una notificación al operador que se desea transferir las tareas y éste deberá aceptarlas (ya que no tendría sentido transferirle tareas a un Operador que ya está muy ocupado). Si el segundo operador acepta, el sistema realiza la transición de forma que el vehículo no quede incomunicado en ningún momento.

En segundo lugar, se ha implementado entre las estrategias de Adaptabilidad del CCV un sistema de Transparencia que nos permitirá ocultar ciertos componentes gráficos del Interfaz de usuario cuando no sean relevantes. Este mecanismo, como podemos ver en la Figura 5 se modeliza mediante

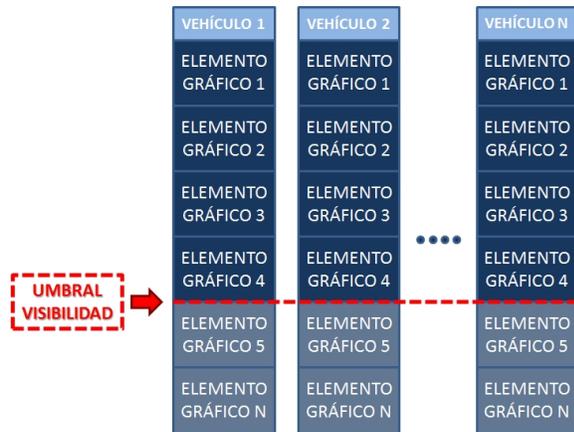


Figura 5: Esquema de Vectores de Visibilidad para Transparencia

dos elementos:

Los *Vectores de Visibilidad*. Se trata de un array de elementos gráficos donde se almacenarán ordenados de mayor a menor prioridad (un elemento gráfico con mayor prioridad tendrá más importancia y por tanto tendrá más probabilidad de ser mostrado en pantalla) todos los elementos que representen datos recibidos de un vehículo. Por lo tanto, cada uno de estos vectores corresponde a uno de los grupos de elementos gráficos donde se muestran los datos de un vehículo. La prioridad de los elementos variará en tiempo de ejecución en base a los datos recibidos por el *Motor de Adaptabilidad*, que decide cuándo se deben reordenar los elementos mediante la estrategia establecida para ello en ese momento o por algún disparador recibido de los datos de una misión, vehículo o bajo demanda del operador.

Los *Umbrales de Visibilidad*. Indican a cada *Vector de Visibilidad* cuántos elementos deben mostrarse por pantalla. Este número puede variar por las razones mencionadas anteriormente. Por ejemplo, si se detecta que el usuario está sobrecargado, se puede optar por disminuir el número de elementos a mostrar para que su carga de trabajo de monitorización disminuya. Por el contrario, si la carga de trabajo o el nivel de estrés del usuario disminuye, el *Umbral de Visibilidad* aumentará permitiendo al *Interfaz Gráfico* mostrar un mayor número de elementos de cada vehículo.

Los valores asignados a cada elemento gráfico, como hemos comentado anteriormente, pueden variar durante el desarrollo de la misión. Algunos ejemplos de estos valores pueden ser: la *prioridad*, que indica la importancia del elemento gráfico para este vehículo y misión; el *coste* de mostrar el elemento en pantalla y el *valor* que el usuario le otorga a ver dicho elemento. Como es posible que

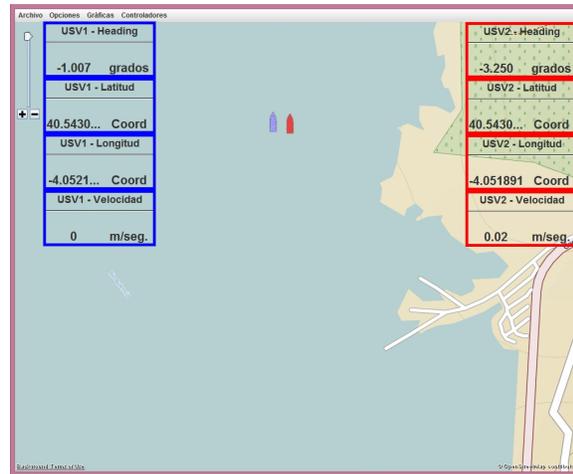


Figura 6: Funcionamiento de Transparencia 1

durante el transcurso de la misión al operador le resulte necesario algún elemento gráfico no visible en ese momento, se podrá elegir cualquiera de ellos mediante un menú desplegable del *Interfaz Gráfico* para mostrarlo instantáneamente.

### 3.2.3. FUNCIONAMIENTO

Un ejemplo de funcionamiento de Transparencia en el CCV puede ilustrarse con una misión de localización en la que los barcos pararán momentáneamente y un cuatrirrotor autónomo (UAV - Unmanned Aerial Vehicle) despegará desde una plataforma en el barco líder para realizar un vuelo de reconocimiento.

Se parte de la configuración mostrada en la Figura 6, en la que los dos barcos han acudido a un punto de ruta que se les ha indicado desde el *Planificador*. Durante la aproximación al punto de ruta, sólo se monitorizan los dos barcos ya que el cuatrirrotor está parado en la plataforma del barco líder, por lo que se pueden mostrar hasta cuatro elementos relevantes referentes a cada uno de los barcos. Al llegar al punto señalado ambos vehículos se paran y el cuatrirrotor despega.

En la Figura 7 vemos la siguiente fase de la maniobra, en la que ambos barcos ya están parados y el cuatrirrotor ha despegado, por lo que aparece su bloque de elementos gráficos (abajo, en negro). Al *Motor Adaptativo* le llegan los datos relativos a la misión, número de vehículos y el usuario desde la *BD de Perfiles y Restricciones* y el *Bloque de Parámetros* y mediante el algoritmo que se haya elegido como estrategia, cambiará el *Umbral de Visibilidad* de los barcos que, al estar parados, tienen menos datos relevantes a monitorizar (la posición). En este caso se fija el Umbral para los barcos en 2 y el del cuatrirrotor en 4, pudiendo monitorizar todos los datos necesarios para cono-

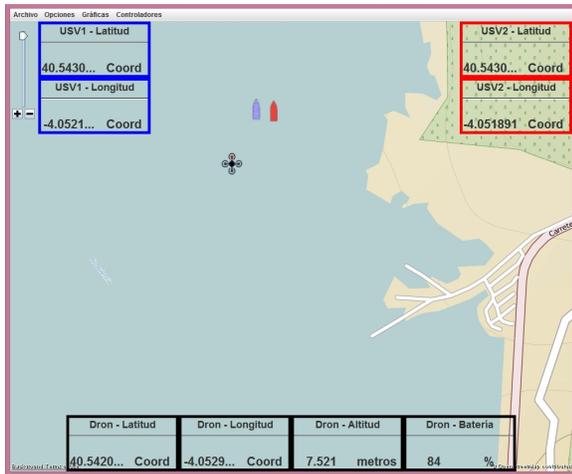


Figura 7: Funcionamiento de Transparencia 2

cer su estado y situación en la misión.

#### 4. CONCLUSIONES Y TRABAJO FUTURO

Se ha diseñado un Centro de Control Versátil reconfigurable que cuenta con una arquitectura distribuida que, además de encapsular algunos aspectos de la codificación de datos y comunicaciones (facilitando la reutilización de cada uno de los módulos que la forman), otorga una mayor flexibilidad a la hora de monitorizar y comandar los vehículos implicados en cada misión (permitiendo la redistribución de recursos entre operadores en tiempo real).

Además se ha dotado al Software de una infraestructura lo suficientemente robusta y flexible para implementar y realizar experimentos sobre los dos conceptos descritos, Adaptabilidad y Transparencia. Esto añade valor a nuestro CCV y tiene el objetivo de facilitar la labor del usuario reubicando y cambiando el aspecto de los elementos gráficos del Interfaz de Usuario o incluso ocultando algunos de menor relevancia momentáneamente, reduciendo así la carga de trabajo que el operador debe soportar.

Actualmente se está trabajando en la implementación de diferentes algoritmos que formen las Estrategias y Tácticas para el *Motor Adaptativo* y de esta forma poder probar cuáles se ajustan mejor a las necesidades de los operadores y las misiones en el proyecto SALACOM. También se está trabajando en fusionar las medidas, tanto de estado [28] como de entorno [29] de los vehículos para obtener una estimación más precisa y completa en la detección de obstáculos potenciales o a la hora de actuar de forma reactiva en el caso de colisión inminente.

#### Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Educación, Cultura y Deporte bajo el proyecto DPI2013-46665-C2.

#### Referencias

- [1] Patterson, M. C., Mulligan, A., and Boiteux, F. (2013) Safety and security applications for micro-unmanned surface vessels. *2013 OCEANS-San Diego*, pp. 1–6, IEEE.
- [2] Ribas, D., Palomeras, N., Ridao, P., Carreras, M., and Mallios, A. (2012) Girona 500 auv: From survey to intervention. *IEEE ASME Transactions on Mechatronics*, **17**, 46–53.
- [3] Sutton, R., Sharma, S., and Xao, T. (2011) Adaptive navigation systems for an unmanned surface vehicle. *Journal of Marine Engineering & Technology*, **10**, 3–20.
- [4] Murphy, R. R., Steimle, E., Griffin, C., Cullins, C., Hall, M., and Pratt, K. (2008) Cooperative use of unmanned sea surface and micro aerial vehicles at hurricane wilma. *Journal of Field Robotics*, **25**, 164–180.
- [5] Lindemuth, M., et al. (2011) Sea robot-assisted inspection. *IEEE robotics & automation magazine*, **18**, 96–107.
- [6] Heo, J., Kim, S., and Kwon, Y. (2016) Design of ground control station for operation of multiple combat entities. *Journal of Computer and Communications*, **4**, 66–71.
- [7] Mupparapu, S. S., Chappell, S. G., Komerska, R. J., Blidberg, D. R., Nitzel, R., Benton, C., Popa, D. O., and Sanderson, A. C. (2004) Autonomous systems monitoring and control (asmac)-an auv fleet controller. *Autonomous Underwater Vehicles, 2004 IEEE/OES*, pp. 119–126, IEEE.
- [8] Bürkle, A., Segor, F., Kollmann, M., and Schönbein, R. (2011) Universal ground control station for heterogeneous sensors. *Journal On Advances in Telecommunications, IA-RIA*, **3**, 152–161.
- [9] Burmeister, H.-C., Bruhn, W., Rødseth, Ø. J., and Porathe, T. (2014) Autonomous unmanned merchant vessel and its contribution towards the e-navigation implementation: The munin perspective. *International Journal of e-Navigation and Maritime Economy*, **1**, 1–13.

- [10] Walter, B. E., Knutzon, J. S., Sannier, A. V., and Oliver, J. H. (2004) Virtual uav ground control station. *AIAA 3rd Unmanned Unlimited Technical Conference, Workshop and Exhibit*.
- [11] Chen, J., Procci, K., Boyce, M., Wright, J., Garcia, A., and Barnes, M. (2014) Situation awareness-based agent transparency(no. arl-tr-6905).
- [12] Mercado, J. E., Rupp, M. A., Chen, J. Y. C., Barnes, M. J., Barber, D., and Procci, K. (2016) Intelligent agent transparency in human-agent teaming for multi-UxV management. *Human Factors*, **58**, 401–415.
- [13] Garlan, D., Cheng, S.-W., Huang, A.-C., Schmerl, B., and Steenkiste, P. (2004) Rainbow: Architecture-based self-adaptation with reusable infrastructure. *Computer (IEEE Computer Society)*, **7 ( Issue: 10 )**, 48–54.
- [14] Cheng, S.-W., Garlan, D., and Schmerl, B. (2006) Architecture-based self-adaptation in the presence of multiple objectives. *SEAMS 06 Proceedings of the 2006 international workshop on Self-adaptation and self-managing systems*, pp. 2–8.
- [15] Cheng, S.-W. and Garlan, D. (2007) Handling uncertainty in autonomic systems. *Automated Software Engineering - ASE 07*.
- [16] Cheng, S.-W., Huang, A.-C., Garlan, D., Schmerl, B., and Steenkiste, P. (2004) Rainbow: Architecture-based self-adaptation with reusable infrastructure. *Proceedings of the International Conference on Autonomic Computing (ICAC 04)*, pp. 276 – 277.
- [17] Cummings, M. L., How, J. P., Whitten, A., and Toupet, O. (2012) The impact of human–automation collaboration in decentralized multiple unmanned vehicle control. *Proceedings of the IEEE*, **100**, 660–671.
- [18] Bonache Seco, D. C. J. M. M. L.-O. J. B. P. E. d. l. C. G. J., J.A. (2017) Centro de control de tierra para colaboración de vehículos autónomos marinos. *Revista Iberoamericana de Automática e Informática Industrial (RIAI)*.
- [19] Bonache Seco, J. A., López Orozco, J. A., Besada Portas, E., and de la Cruz, J. M. (2016) Centro de control versátil: Estado actual y evolución hacia la adaptabilidad. pp. 979–986, CEA.
- [20] Lapierre, L. and Soetanto, D. (2007) Nonlinear path-following control of an auv. *Ocean engineering*, **34**, 1734–1744.
- [21] Park, S., Deyst, J., and How, J. P. (2007) Performance and lyapunov stability of a nonlinear path following guidance method. *Journal of Guidance, Control, and Dynamics*, **30**, 1718–1728.
- [22] de la Cruz, J. M., Lopez-Orozco, A, J., Besada Portas, E., and Aranda Almansa, J. (2016) Control de formaciones de vehículos marinos de superficie con restricciones de entrada. pp. 1044–1051, CEA.
- [23] de la Cruz, J. M., Lopez-Orozco, A, J., Besada Portas, E., Moreno Salinas, D., and Aranda Almansa, J. (2014) Seguimiento de caminos para formaciones de vehículos marinos de superficie.
- [24] de la Cruz, J. M., Lopez-Orozco, J. A., Besada-Portas, E., and Aranda-Almansa, J. (2015) A streamlined nonlinear path following kinematic controller. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6394–6401, IEEE.
- [25] Naghshineh, T. K. . S. K. . Y. C. . M. (2000) Threshold-type call admission control in wireless/mobile multimedia networks using prioritised adaptive framework. *Electronics Letters IEEE 2000*, **36**, 852–854.
- [26] Kuo, L. H. S. K. C.-C. (2004) Adaptive resource allocation for multimedia qos management in wireless networks. *IEEE Transactions on Vehicular Technology*, **53**, 547–558.
- [27] Oreizy, P., Gorlick, M. M., Taylor, R. N., Heimbigner, D., Johnson, G., Medvidovic, N., Quilici, A., Rosenblum, D. S., and Wolf, A. L. (1999) An architecture-based approach to self-adaptive software.
- [28] Besada-Portas, E., Lopez-Orozco, J. A., Besada, J., and Jesus, M. (2011) Multisensor fusion for linear control systems with asynchronous, out-of-sequence and erroneous data. *Automatica*, **47**, 1399–1408.
- [29] Besada-Portas, E., Lopez-Orozco, J. A., and de la Cruz, J. (2002) Unified fusion system based on bayesian networks for autonomous mobile robots. *Information Fusion, 2002. Proceedings of the Fifth International Conference on*, vol. 2, pp. 873–880, IEEE.