



Architecture de control pour la gestion contextuelle de tâches sur un AUV

Abdellah El Jalaoui, David Andreu, Bruno Jouvencel

► **To cite this version:**

Abdellah El Jalaoui, David Andreu, Bruno Jouvencel. Architecture de control pour la gestion contextuelle de tâches sur un AUV. 2005, pp.287-288. <lirmm-00125754>

HAL Id: lirmm-00125754

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00125754>

Submitted on 22 Jan 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Architecture de contrôle pour la gestion Contextuelle de tâches sur les AUV

A. El Jalaoui, D. Andreu, B. Jouvencel
LIRMM, 161 rue Ada, 34392 Montpellier cedex 5
email: {eljalaoui, andreu, jouvencel}@lirmm.fr

Résumé

Un AUV se compose principalement d'un système informatique embarqué et d'un ensemble de capteurs et d'actionneurs. La nature des tâches à effectuer diffère d'une mission à une autre, voire d'une phase de mission à une autre. Une grande importance est alors accordée au pouvoir d'adaptation, face aux évolutions technologiques et applicatives. On présentera une méthode de conception d'une architecture de commande pour l'AUV Taipan en cours de développement au LIRMM. Cette architecture mixte comporte trois niveaux, un superviseur global, des superviseurs locaux et enfin un ensemble de modules de gestion des moyens d'action et de perception. Cette architecture tente de satisfaire certains critères de conception tels que la modularité, l'évolutivité et la réactivité.

Mots clef

AUV, Architecture de contrôle, Approche contextuelle, systèmes embarqués.

1 Introduction

Le besoin d'opérer dans des eaux de plus en plus profondes et de réduire les coûts, amène les recherches à se concentrer sur l'élaboration de véhicules autonomes capables de se déplacer seuls et de mener à bien des tâches qui nécessitent encore récemment l'assistance de l'opérateur humain.

Ce besoin d'autonomie dans un milieu en constante évolution requiert de la part du véhicule une certaine capacité à pouvoir, à chaque instant, évaluer son état et l'état de son environnement, les combiner avec la mission qu'il lui a été confié et prendre une décision cohérente.

Les architectures de contrôle développées pour ces véhicules deviennent vite complexes et requièrent une méthodologie de conception. On trouve trois principales catégories d'architectures [2]:

- les architectures réactives sont composées de plusieurs modules. Chaque module génère une commande dépendante directement d'un vecteur d'entrées fournies par un ensemble de capteurs. Les différentes commande générées par ces modules sont alors sommée après pondération. La complexité de ce type d'architecture réside dans le réglage des coefficients de pondération permettant de laisser s'exprimer chacune des fonctionnalités tout en s'assurant que le système converge globalement vers l'objectif désiré.
- les architectures hiérarchisées sont construites sur plusieurs niveaux, généralement trois: le niveau haut où sont effectués les prises de décision, le

niveau intermédiaire qui s'occupe du contrôle et de la supervision et le niveau bas qui génère les commandes à appliquer. La récupération des informations externes ainsi que la propagation des décisions du niveau haut nécessite de traverser les niveaux inférieurs ce qui confère à ce type d'architecture une réactivité réduite.

- les architectures mixtes sont à la fois hiérarchisées et réactives. Cela leurs permet alors d'accomplir les objectifs désirés tout en réagissant rapidement aux situations imprévues qui peuvent survenir au cours de la mission.

Trois principaux critères sont généralement utilisés pour évaluer la qualité d'une architecture:

- Modularité: Un logiciel complexe doit pouvoir être subdivisé en plusieurs modules qui pourront être conçus, implémentés et testés séparément.
- Evolutivité: Le logiciel doit pouvoir être facilement adapté lorsque les applications ou l'équipement embarqué (capteurs par exemple) sont amenés à changer.
- Robustesse: Le véhicule doit être capable de mener à bien une séquence d'objectifs dans un milieu de grande incertitude et de pouvoir rapidement répondre à un événement imprévu.

Ce travail va essayer de présenter une nouvelle méthodologie de conception des architectures de contrôle des AUV. L'étude se basera sur le cas de l'AUV Taipan. De taille réduite et conçu pour opérer dans des eaux peu profondes, Taipan mesure 1.8 m de long, 20 cm de diamètre et pèse environ 51 Kg. (voir [3] pour plus de détail). L'architecture issue de cette étude est une architecture mixte composée de trois niveaux. Un superviseur global, plusieurs superviseurs locaux (un pour chaque mode: autonome, téléopération, coopération) et un ensemble de modules gérant l'instrumentation. Chaque niveau manipule des objets (au sens objets informatiques) correspondant à son niveau d'abstraction. Trois types d'objets sont utilisés dans cette architecture. Le superviseur global reçoit de la part de l'opérateur un ensemble d'objets *objectif* définissant la mission à exécuter. Ensuite il transmet au superviseur local concerné une séquence de *sous-objectifs*. Ce dernier envoie des *ordres* aux modules de gestion de l'instrumentation afin de réaliser les *sous-objectifs* en cours.

2 Architecture de contrôle

2.1 Superviseur global

Le superviseur global (SG) reçoit de la part de l'opérateur (ou d'un gestionnaire de mission) un fichier

contenant une mission à exécuter. Une mission est une succession d'*objectifs* que le système doit atteindre durant la navigation. Les objectifs peuvent être un déplacement ou d'autres actions à effectuer à un endroit géographique précis (bathymétrie par exemple). Ces objectifs peuvent être atteints successivement ou en parallèle selon leur nature. Les informations relatives à leur ordonnancement sont spécifiées par l'utilisateur avant le départ de la mission.

Cela permet au superviseur global d'ordonner (séquencer) correctement les objectifs qui vont constituer au final une mission.

Le superviseur global s'assure que les objectifs qui vont être exécutés en même temps ne vont pas utiliser la même ressource. Par exemple le superviseur global empêchera les objectifs *remontée en surface* et *inspecter pipeline* d'être lancés simultanément parce qu'ils utilisent le même ensemble de ressources {propulseur, gouvernes}. Plus précisément cet ensemble de ressources est appelé macro-ressource.

D'autre part, l'objectif *inspection pipeline* et l'objectif *cartographie* utilisent respectivement la macro-ressource {propulseur, gouvernes} et la ressource {sonars latéraux} ce qui les autorise à s'exécuter en même temps. La gestion des ressources au niveau du superviseur global évite que des ordres contradictoires ne soient envoyés aux actionneurs (par le biais des modules d'action).

Pour accomplir un objectif, l'AUV passe par plusieurs étapes. Ces différentes étapes requièrent l'utilisation de différentes lois de commande comme par exemple dans les étapes: *plonger*, *inspecter pipeline*, Ainsi nous sommes amenés à décomposer un objectif en plusieurs entités plus simples. Ces entités que l'on a appelées *sous-objectifs* ne requièrent, durant tout le temps de leur exécution, qu'une seule et même loi de commande. Par exemple l'objectif *inspecter pipeline* sera décomposé en la séquence: {*plonger*, *aller à*, *chercher pipeline*, *suivre pipeline*}.

Finalement, le superviseur global envoie au superviseur local concerné les sous-objectifs à exécuter. Ce dernier, après la réalisation de chaque sous-objectif, lui retournera un rapport d'exécution.

2.2 Superviseur local

Un superviseur local est dédié au contrôle d'une ressource dans un mode donné. En ce qui concerne Taipan, nous avons seulement une ressource (le véhicule lui-même) qui a trois modes de fonctionnement:

- mode autonome: exécute les sous-objectifs issus du superviseur global.
- mode téléopération: téléopération bas niveau du véhicule par l'opérateur (utilisable à l'heure actuelle seulement lorsque l'AUV est en surface).
- mode coopération: commande l'AUV pour le positionner dans une flottille de véhicule.

Nous nous intéresserons ici qu'au mode autonome.

Comme cela avait été évoqué précédemment, le superviseur local reçoit des sous-objectifs à exécuter

(par exemple *plonger*, *aller à*). Pour exécuter un sous-objectif, nous avons besoin de collecter un ensemble de données capteurs et d'envoyer un ensemble de commandes aux actionneurs. Le dialogue avec les capteurs et les actionneurs est assuré par des modules bas niveau que le superviseur local active selon le sous-objectif à exécuter.

2.3 Modules de gestion de l'instrumentation

On trouve deux sortes d'instruments sur un AUV, les capteurs et les actionneurs. Les premiers sont gérés par les Modules de Perception (MP) et les seconds par les Modules d'Action (MA). Tous ces modules ainsi que les différents instruments, utilisent un bus commun pour échanger leurs données.

Un MP est construit pour chaque type de données (appelé variable) requise dans l'architecture (par exemple la position $x, y, z, \theta, \phi, \psi$). Il est souvent nécessaire d'utiliser des données provenant de plusieurs capteurs afin de calculer ou estimer précisément une variable. Ainsi un MP peut gérer plusieurs capteurs.

Deux sortes de traitement sont à la charge d'un MP; le premier est relatif à la configuration des capteurs (mise sous/hors tension, changement de mode), le second est le traitement des données du ou des capteur(s).

Les MA, qui contiennent dans la plupart des cas des lois de commandes, permettent de calculer les commandes à envoyer aux actionneurs.

Tous ces modules sont activés et configurés par le superviseur local. Ainsi pour récupérer la position du véhicule, le superviseur local envoie un ordre au MP correspondant et pour atteindre une position, il envoie un ordre au MA gérant le propulseur et les gouvernes. Les MP et les MA fonctionnent de façon périodique.

Les variables issues des MP (estimées ou mesurées) sont échangées entre les modules bas niveau par l'intermédiaire d'un bus commun.

3 Conclusion

Le travail présenté, s'intègre dans le domaine des architectures logicielles de contrôle pour la robotique. Une architecture de contrôle, développée pour les AUV (Autonomous Underwater Vehicle) est proposée et tente de répondre à certains critères de conception, tels que la modularité, l'évolutivité et la robustesse.

Références

- [1] J.D. Carbou, D. Andreu, P. Fraisse, "Events as a key of an autonomous robot controller", *15th IFAC World Congress (IFAC'02)*, Barcelone, Espagne, 21-26, juillet 2002.
- [2] P. Ridaou, M. Carreras, J. Batlle, J. Ama, "O²ca²: A new hybrid control for a low cost auv", *Proceedings of the Control Application in Marine Systems*, 2001.
- [3] J. Vaganay, B. Jouvencel, P. Lepinay and R. Zapata, "Taipan an AUV for very shallow water applications", *World Automation Congress*, 1998.