Fábio Henrique de Assis

Checagem de Arquiteturas de Controle de Veículos Submarinos: uma abordagem baseada em especificações formais

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção de Título de Mestre em Engenharia.

São Paulo
2009

Fábio Henrique de Assis

Checagem de Arquiteturas de Controle de Veículos Submarinos: uma abordagem baseada em especificações formais

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção de Título de Mestre em Engenharia.

Área de concentração:
Controle e Automação

orientador: Newton Maruyama

São Paulo
2009

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, de julho de 2009.

Assinatura do autor _____

Assinatura do orientador_____

FICHA CATALOGRÁFICA

Assis, Fábio Henrique de
Checagem de arquiteturas de controle de veículos submarinos: uma abordagem baseada em especificações formais / F.H. de Assis. -- ed.rev. -- São Paulo, 2009.
144 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

1. Submersíveis não tripulados 2. Arquitetura de software (Especificação) I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecâ – nicos. II. t.

Dedicatória

à minha família.

Agradecimentos

Gostaria de agradecer primeiramente aos meus pais e ao meu irmão, que sempre me apoiaram em todas as escolhas que tomei em relação à minha carreira, principalmente nos momentos mais difíceis.

Agradeço também à minha namorada por estar sempre comigo, e muitas vezes por compreender minhas ausências que se fizeram necessárias ao longo deste trabalho.

Não poderiam ficar de fora as pessoas que contribuíram de maneira efetiva para que esse trabalho fosse concluído, que são meu orientador e meus colegas de laboratório, além de todos os membros da XMobots, opinando, discutindo e muitas vezes ajudando a desenvolver certas partes do trabalho.

"O único lugar onde sucesso vem antes do trabalho é no dicionário"

Albert Einstein

Resumo

O desenvolvimento de arquiteturas de controle para veículos submarinos é uma tarefa complexa. Estas podem ser caracterizadas pelos seguintes atributos: tempo real, multitarefa, concorrência e comunicações distribuídas em rede. Neste cenário, existem múltiplos processos sendo executados em paralelo, possivelmente distribuídos, e se comunicando uns com os outros. Neste contexto, o modelo comportamental pode levar a fenômenos como *deadlocks*, *livelocks*, disputa por recursos, entre outros. A fim de se tentar minimizar os efeitos de tais dificuldades, neste trabalho será apresentado um método para checagem de modelos de arquiteturas de controle de veículos submarinos baseado em Especificações Formais. A linguagem de especificação formal escolhida foi CSP-OZ, uma combinação de CSP e Object-Z. Object-Z é uma extensão orientada a objetos da linguagem Z para a especificação de predicados, tipicamente pré e pós condições, além de invariantes de dados. CSP (*Communicating Sequential Process*) é uma álgebra de processos desenvolvida para descrever modelos comportamentais de processos paralelos. A checagem de modelos especificados formalmente consiste na análise das especificações para verificar se um sistema possui certas propriedades através de uma busca exaustiva em todos os estados em que este pode entrar durante sua execução. Neste contexto, é possível checar corretude, *livelocks*, *deadlocks*, etc. Além disso, pode-se relacionar duas especificações diferentes a fim de se checar relações de refinamento. Para as especificações, o verificador de modelos FDR da Formal Systems Ltd. será utilizado. A implementação é desenvolvida utilizando um perfil da linguagem Ada denominado RavenSPARK, uma junção do perfil Ravenscar (desenvolvido na Universidade de York) com a linguagem SPARK (um subconjunto da linguagem Ada desenvolvido pela Praxis, Inc.). O Ravenscar é um perfil para desenvolvimento de processos, e portanto os processos de CSP, incluindo seus canais de comunicação, podem ser facilmente criados. Por outro lado, SPARK é uma linguagem onde podem ser inseridos predicados para os dados (originalmente especificados em Object-Z) utilizando anotações da própria linguagem. A linguagem SPARK possui uma ferramenta, o Examinador, que pode checar códigos de modelos baseado nestas anotações. Em resumo, o método proposto permite tanto a checagem de modelos em CSP quanto a checagem no nível de código. Para isso, as especificações em Object-Z devem inicialmente ser convertidas em um código na linguagem SPARK juntamente com suas respectivas anotações, para que então a checagem do modelo possa ser realizada no código. O desenvolvimento de uma arquitetura de controle reativa para um ROV denominado VSOR (Veículo Submarino Operado Remotamente) é utilizado como exemplo de uso do método proposto. Toda a arquitetura de controle é codificada utilizando a linguagem Ada com o perfil RavenSPARK e embarcada em um computador do tipo PC104 com o sistema operacional de tempo real VxWorks, da Windriver, Inc.

Abstract

The development of control architectures for Underwater Vehicles is a complex task. These control architectures might be chracterised by the following attributes: real-time, multitasking, concurrency, and distributed over communication networks. In this scenario, we have multiple processes running in parallel, possibly distributed, and engaging in communication between each other. In this context, the behavioural model might lead to phenomena like deadlocks, livelocks, race conditions, among others. In order to try to minimize the effects of such difficulties, in this work a method for model checking control architectures of underwater vehicles based on formal specifications is presented. The chosen formal specification language is CSP-OZ, a combination of CSP and Object-Z. Object-Z is an object-oriented extension of Z for the specification of predicates, typically, data pre, post and invariant conditions. CSP (Communicating Sequential Process) is a process algebra developed to describe behavioural models of parallel process. The model checking of formal specifications is a task of reasoning on specifications in which a system verifies certain properties by means of an exhaustive search of all possible states that a system could enter during its execution. In this context, it is possible to check about correctness, liveness, deadlock, etc. Also, one can relate two different specifications in order to check a refinement ordering. For the specifications, the model checker FDR of Formal Systems Ltd. is utilised. The implementation is developed using an ADA language profile called RavenSPARK, a union of the Ravenscar profile (developed at the University of York) and the SPARK language (a subset of the ADA language developed by Praxis, Inc.). The Ravenscar is a profile for developing processes, so CSP processes including their message channels can be easily deployed. On the other hand, SPARK is a language where one can insert data predicates (originally specified in Object-Z) using language annotations. The SPARK language has a tool, the Examiner, that can model check code based on these annotations. In summary, the proposed method allows model checking of CSP processes but does not allow any checking in the code level. On the contrary, Object-Z specifications must first be converted into a SPARK language code, together with proper annotations, and then model checking can be realised in code. The development of a real-time reactive control architecture of an ROV named VSOR (Veiculo Submarino Operado Remotamente) is used as an example of the use of the proposed method. The whole control architecture is coded using the ADA Language with the RavenSPARK profile and deployed into a PC104 cpu system running the Vxworks real-time operating system of Windriver, Inc.

Lista de Figuras

Lista de Tabelas