



Anais

Volume 02
ISSN 2178-6097

WTDSOft 2014

IV WORKSHOP DE TESES E DISSERTAÇÕES DO CBSOFT

COORDENADOR DO COMITÊ DE PROGRAMA

Eduardo Santana de Almeida - Universidade Federal da Bahia (UFBA)

COORDENAÇÃO DO CBSOFT 2014

Baldoino Fonseca - Universidade Federal de Alagoas (UFAL)

Leandro Dias da Silva - Universidade Federal de Alagoas (UFAL)

Márcio Ribeiro - Universidade Federal de Alagoas (UFAL)

REALIZAÇÃO

Universidade Federal de Alagoas (UFAL)

Instituto de Computação (IC/UFAL)

PROMOÇÃO

Sociedade Brasileira de Computação (SBC)

PATROCÍNIO

CAPES, CNPq, INES, Google

APOIO

Instituto Federal de Alagoas, Aloo Telecom, Springer, Secretaria de Estado do Turismo AL, Maceió Convention & Visitors Bureau, Centro Universitário CESMAC e Mix Cópia

Adaptação de critérios de teste de programas concorrentes para o teste de integração de robôs móveis

Marcos Pereira dos Santos¹, Simone R. S. Souza¹

¹ Instituto de Ciências Matemáticas e de Computação (ICMC) –
Universidade de São Paulo (ICMC/USP)
Caixa Postal 668 – 13.560-970 – São Carlos – SP – Brasil

{mpereira, srocio}@icmc.usp.br

Abstract. *This paper presents a master's project under development whose goal is to adapt concurrent program coverage testing criteria to the context of mobile robots, mapping these coverage criteria to deal with errors related to the communication among the components of the mobile robots systems. The proposed approach should be incorporated into the ROS framework, that has been used to support the development of mobile robots. Another objective of this project is conducting an experimental study to compare the applicability of the coverage testing criteria considering other environments to develop mobile robots.*

Resumo. *Este artigo apresenta um projeto de mestrado em andamento cujo objetivo é a adaptação de critérios de teste de programas concorrentes para o contexto de robôs móveis, adaptando-os para tratar erros relacionados à comunicação entre os componentes de um sistema para robôs móveis. A abordagem proposta deve ser incorporada ao framework ROS, que vem sendo utilizado para apoiar o desenvolvimento de robôs móveis. Outro objetivo deste projeto é realizar estudos comparativos para avaliar a aplicabilidade dos critérios de teste considerando outros ambientes de desenvolvimento de robôs móveis.*

1. Introdução

A atividade de teste de software é uma análise dinâmica do produto de software e tem por objetivo encontrar os defeitos presentes no mesmo. Para isso, técnicas e critérios de teste são propostos, os quais devem ser aplicados nas diferentes fases do processo de desenvolvimento. O teste de integração é uma atividade sistemática aplicada durante a integração das subpartes que compõem o programa com o objetivo de descobrir erros associados às interfaces entre os módulos. O objetivo é, a partir dos elementos testados no nível de unidade, construir a estrutura do programa de acordo com o definido no projeto do software [Delamaro et al. 2007].

A programação concorrente está cada vez mais sendo utilizada no desenvolvimento de diversos tipos de sistemas para áreas como: financeira, científica, comercial, redes de sensores, computação ubíqua, entre outras. Um programa concorrente é formado por dois ou mais processos ou *threads* que trabalham juntos para realizar uma tarefa, sendo que essa interação pode ocorrer de forma sincronizada ou não e os processos podem ou não concorrerem aos mesmos recursos computacionais [Andrews 2000]. Esse paradigma de programação permite o desenvolvimento de algoritmos que quando

executados originam novos processos que podem executar concorrentemente e interagirem entre si [Pacheco 2011]. Diferentes execuções de um programa concorrente com a mesma entrada podem produzir diferentes saídas em razão das diferentes sincronizações executadas. O desafio do teste de software nesse contexto é identificar todas as possíveis sincronizações para verificar se as saídas obtidas são as esperadas [Taylor et al. 1992, Wong et al. 2005, Souza et al. 2012].

Um sistema robótico em razão da sua própria natureza, constituído por diversos componentes interagindo, apresentam um certo nível de similaridade com programas concorrentes. Um robô móvel autônomo deve ser capaz de interagir com o ambiente, aprender e tomar decisões corretas para que suas tarefas sejam executadas com êxito. Para o desenvolvimento de robôs móveis são utilizados ambientes que permitem simular o comportamento dos robôs à medida são sendo desenvolvidos. Exemplos destes ambientes são: ROS, Player, Ária, Carmen, Microsoft Robotics Studio, dentre diversos outros [Wolf et al. 2009]. Dentre estes ambientes de desenvolvimento de robôs móveis pode-se destacar o ROS ¹, o qual oferece bibliotecas e um conjunto de ferramentas para auxiliar os desenvolvedores de aplicações para robôs, possibilitando abstrair o hardware e drivers de dispositivo, de modo a simular o funcionamento da aplicação.

Este projeto de mestrado visa a contribuir nesse cenário, em que o objetivo é a investigação e adaptação dos critérios de teste estruturais de programas concorrentes [Souza et al. 2008] para o contexto de sistemas embarcados robóticos, realizando um estudo comparativo desses critérios, considerando o ambiente ROS e outros utilizados para o desenvolvimento desses sistemas.

2. Caracterização do Problema

No contexto de programas sequenciais foram definidos ao longo dos anos várias técnicas e critérios de teste de software. O teste de aplicações concorrentes comparado ao teste de programas sequenciais é mais complicado, pois além das dificuldades inerentes à atividade de teste, novos desafios são impostos principalmente devido ao comportamento não determinístico em que diferentes sequências de sincronização podem ser obtidas com a utilização de uma mesma entrada de teste, e por isso, faz-se necessário avaliar se o comportamento das saídas é o esperado ou não. Uma das linhas de pesquisa do grupo de Engenharia de Software do ICMC/USP está relacionada ao teste de programas concorrentes e diversas são as contribuições neste sentido [Souza et al. 2005, Hausen et al. 2007, Souza et al. 2008, Sarmanho et al. 2008, Souza et al. 2012].

Partindo para o teste de sistemas para robôs móveis, um dos desafios é testar se as informações vindas dos sensores e componentes do sistema. É de suma importância que o robô preserve a sua integridade, bem como também os elementos presentes em seu ambiente como seres humanos, objetos, utensílios, etc. A ação de um robô deve ocorrer de modo ativo sobre um determinado objeto alvo, se realmente for prevista a sua ação sobre este objeto [Wolf et al. 2009]. Observa-se que muitos aspectos presentes em programas concorrentes são semelhantes aos encontrados em sistemas para robôs móveis, os quais precisam interagir com o meio e com as partes que o compõem. Entre as principais semelhanças encontradas entre programas concorrentes e sistemas embarcados para robôs móveis pode-se destacar a comunicação e sincronização de tarefas. Um sistema para

¹ROS - <http://ros.org/wiki/>

robô móvel é composto por diversos dispositivos como sensores, atuadores e programa de controle central que interagem e executam código de forma paralela. A comunicação e sincronização entre as tarefas desempenhadas por cada um dos dispositivos, muitas vezes, é não determinística, assim como em programas concorrentes, em que há diversos processos/*threads* executando de forma concorrente. Considerando esses aspectos, surgiu a motivação de investigar o teste de integração em sistemas embarcados para robôs móveis, buscando explorar questões que vêm sendo tratadas no teste de programas concorrentes.

Para ilustrar as definições apresentadas, considere o grafo da Figura 1. Este aplicação apresentada representa um exemplo de comunicação entre componentes de um sistema para robôs móveis autônomos, desenvolvida utilizando o ROS. O mesmo é constituído por três nós (componentes executáveis) que são: */sensor*, */actuator* e */gps* e pelos tópicos (mecanismo utilizado para troca de mensagens): */info* e */localization*.

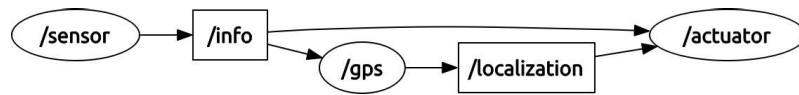


Figura 1. Exemplo de aplicação para sistemas de robôs móveis.

O nó */sensor* publica informações no tópico */info*. Esta mensagem publicada é assinada pelos componentes */actuator* e */gps*. O componente */gps* publica mensagens no tópico */localization* e esta mensagem é subscrita pelo componente */actuator*. Desta maneira ocorre a comunicação e interação entre os componentes deste sistema, assim pretende-se explorar as similaridades destes sistemas com programas concorrentes para adaptação dos critérios de teste.

3. Caracterização da Contribuição

O desenvolvimento deste projeto de mestrado irá contribuir para o estado da arte na área de VV&T (Verificação, Validação e Testes) de robôs móveis. Pretende-se com esse projeto contribuir com o cenário nacional e internacional no que tange a melhoria de qualidade de sistemas para robôs móveis, apresentando critérios de teste de integração para robôs móveis. Assim, espera-se contribuir com a atividade de teste aplicado a sistemas para robôs móveis, por meio da adaptação de critérios de testes de programas concorrentes para robôs móveis. Outra contribuição é apresentar uma proposta de adaptação de critérios de teste para programas concorrentes do nível de unidade para o nível de integração. Desse modo, a adaptação dos critérios de teste ocorre em duas diferentes dimensões: de programas concorrentes para sistemas embarcados robóticos, e do nível de unidade para o nível de integração. Espera-se também gerar resultados sobre a comparação entre diferentes ambientes de desenvolvimento de robótica móvel em relação ao esforço e efetividade da atividade de teste, buscando identificar os desafios impostos ao teste nesses ambientes de desenvolvimento.

4. Estado Atual do Trabalho

Este projeto está vinculado a um doutorado em andamento, o qual explora um modelo para o teste de integração desses sistemas [Brito 2013], contribuindo com a melhoria da qualidade dos sistemas embarcados que vem sendo desenvolvidos no Instituto Nacional de Ciência e Tecnologia (INCT/SEC), com a utilização de técnicas de teste de software.

A etapa atual deste projeto, consiste no estudo sobre o modelo de teste de integração para sistemas embarcados críticos em desenvolvimento pelo grupo de Engenharia de Software do ICMC [Brito 2013]. Também está sendo realizado um estudo dos critérios para teste de programas concorrentes, em que são observados os critérios relacionados à comunicação e sincronização. Outra etapa que está sendo realizada é a adaptação dos critérios definidos para o teste de programas concorrentes para o contexto de sistemas para robôs móveis. Nessa fase estão sendo definidos e implementados os recursos necessários para que esses critérios possam ser utilizados nos ambientes de desenvolvimento de robôs móveis.

5. Trabalhos Relacionados

Na literatura são encontrados trabalhos que propõem critérios de teste de integração para sistemas embarcados [Hossain and Lee 2013, Belli et al. 2009, Lee et al. 2002, Seo et al. 2007, Sung et al. 2011]. A maioria dos estudos apresentam critérios no nível de integração entre unidades que compõe estes sistemas como classes, métodos, funções, etc. A proposta deste projeto possui como objetivo a proposição de critérios de teste de integração para sistemas robóticos móveis considerando características de comunicação entre os componentes. Nos trabalhos relacionados, não foram encontrados estudos que descrevem a utilização dos critérios de teste propostos e sua aplicabilidade para sistemas para robôs móveis ou sistemas mais complexos como sistemas embarcados críticos. Os trabalhos apresentam estudos de caso, no geral, para sistemas embarcados que não são de natureza crítica como celulares, televisão, robôs não autônomos, dentre outros. Algumas características encontradas em sistemas para robôs móveis autônomos devem ser levadas em consideração na aplicação dos critérios de teste para estes sistemas, as quais não são encontradas nos demais sistemas, como em programas concorrentes. Algumas destas características são: capacidade de percepção, capacidade de agir, robustez e inteligência [Wolf et al. 2009]. Desta maneira, requer que os critérios de teste para programas concorrentes sejam estudados e adaptados verificando-se também estas características.

6. Avaliação dos Resultados

Até esta fase do projeto foram obtidos alguns resultados. Foi realizado um mapeamento sistemático sobre teste de integração para sistemas embarcados críticos e está sendo escrito um artigo que será submetido a conferências. Após a adaptação/implementação dos critérios, os mesmos serão validados com a realização de experimentos, os quais deverão englobar como estudos de caso os sistemas desenvolvidos tanto pelo INCT/SEC como CaRINA ², visando avaliar a aplicabilidade da proposta. Os experimentos serão conduzidos com base no processo experimental proposto por Wohlin [Wohlin et al. 2000].

Agradecimento

Os autores gostariam de agradecer a Capes, pelo apoio financeiro.

Referências

Andrews, G. (2000). *Foundations of Multithreaded, Parallel, and Distributed Programming*. Addison-Wesley.

²CaRINA - <http://www.inct-sec.org/br/aplicacoes/carina>

- Belli, F., Hollmann, A., and Padberg, S. (2009). Communication sequence graphs for mutation-oriented integration testing. In *Secure Software Integration and Reliability Improvement. SSIRI. Third IEEE International Conference on*, pages 387–392.
- Brito, M. A. S. (2013). Estudo e definio do teste de integrao de software para o contexto de sistemas embarcados crticos. *ICMC/USP - Projeto de doutorado em andamento*.
- Delamaro, M. E., Maldonado, J., and Jino, M. (2007). *Introdução ao Teste de Software*. Rio de Janeiro: Elsevier (Editora Campus), 1a edition.
- Hausen, A. C., Verglio, S. R., Souza, S. R. S., Souza, P. S. L., and Simão, A. S. (2007). A tool for structural testing of MPI programs. In *Latin-American Test Workshop - LATW*, volume 1, Cuzco.
- Hossain, M. I. and Lee, W. J. (2013). A scalable integration testing approach considering independent usage pattern of global variables. *International Journal of Software Engineering and Its Applications*, pages 195 – 204.
- Lee, N. H., Kim, T. H., and Cha, S. D. (2002). Construction of global finite state machine for testing task interactions written in message sequence charts. In *Proceedings of the 14th international conference on Software engineering and knowledge engineering, SEKE '02*, pages 369–376, New York, NY, USA. ACM.
- Pacheco, P. (2011). *An Introduction to Parallel Programming*. Elsevier.
- Sarmanho, F. S., Souza, P. S. L., Souza, S. R. S., and ao, A. S. S. (2008). Structural testing for semaphore-based multithread programs. In *International Conference on Computational Science*, volume 5101, pages 337–346, Krakow. Springer-Verlag.
- Seo, J., Sung, A., Choi, B., and Kang, S. (2007). Automating embedded software testing on an emulated target board. In *Proceedings of the Second International Workshop on Automation of Software Test, AST '07*, Washington, DC, USA. IEEE Computer Society.
- Souza, P. S. L., Souza, S. R. S., and Zaluska, E. (2012). Structural testing for message-passing concurrent programs: an extended test model. *Concurrency and Computation*.
- Souza, S. R. S., Vergilio, S. R., Souza, P. S. L., Simão, T. G. Bliscosque, A. M. L. A. S., and Hausen, A. C. (2005). Valipar: A testing tool for message-passing parallel programs. pages 386–391.
- Souza, S. R. S., Vergilio, S. R., Souza, P. S. L., Simão, A. S., and Hausen, A. C. (2008). Structural testing criteria for message-passing parallel programs. *Concurr. Comput. : Pract. Exper.*, 20(16):1893–1916.
- Sung, A., Choi, B., Wong, W. E., and Debroy, V. (2011). Mutant generation for embedded systems using kernel-based software and hardware fault simulation. *Information and Software Technology*, 53(10):1153 – 1164.
- Taylor, R. N., Levine, D. L., and Kelly, C. D. (1992). Structural testing of concurrent programs. *IEEE Trans. Softw. Eng.*, 18(3):206–215.
- Wohlin, C., Runeson, P., Hst, M., Ohlsson, M. C., Regnell, B., and Wessln, A. (2000). *Experimentation in Software Engineering: An Introduction*. Kluwer Academic Publishers.
- Wolf, D. F., Osório, F. S., Simões, E., and Trindade Jr., O. (2009). Robótica inteligente: Da simulação às aplicações no mundo real. 1:279–330.
- Wong, W., Lei, Y., and Ma, X. (2005). Effective generation of test sequences for structural testing of concurrent programs. In *Engineering of Complex Computer Systems, 2005. ICECCS 2005. Proceedings. 10th IEEE International Conference on*, pages 539–548.