

Eric Senn, Lucie W. J. Bourdon, Dominique Blouin

Multi-Paradigm Modeling for early Analysis of ROS-based Robotic Applications using a Library of AADL Models

2022

Ana Barbosa

Resumo

O artigo mostra como fazer análise de arquitetura usando o AADL para aplicações ROS.

Apresenta como foi feita a implementação de uma biblioteca de componentes AADL para o ROS com suporte muitas plataformas.

Mostra quais análises é possível observar nesse primeiro momento de implementação do ROS em uma plataforma robótica.

1. Introdução

Aplicações robóticas são um conjunto de diferentes serviços, *Hardware e Software*.

Para confiar nesses serviços, análises iniciais são cruciais.

Com AADL é possível analisar ambos os serviços, e desenvolvedores de aplicações ROS podem se beneficiar disso.

Robôs perfomam multiplas tarefas.

O ROS está asscodiado com todas essas tarefas, transformando mensagens em *publishers* e *subscribers*.

ROS tem desvantagens.

Sistemas embarcados são limitados e, por exemplo, rodar um middleware em cima do outro (ROS e DDS) é caro energeticamente.

Para saber de qual serviço vem o problema é preciso entender:

- **Software** como conjunto de Nós ROS.
- **Hardware** são sensores, atuadores e placas embarcadas.
- Vínculo entre serviços de software e recursos de hardware.

Para análise dos processos de CPU, barramento, tempo, latência, entre outros, a AADL é uma mão na roda.

2. A Biblioteca

Robôs mobile geralmente possuem duas placas de circuito.

- Um SoC para processar o ROS e seus Nós.
- Uma microcontroladora para gerenciar os atuadores.

Nós específicos são usados para troca de mensagens entre elas.



A biblioteca é organizada em quatro partes:

- **Pacote ROS principal** - Contém definição dos tipos de componentes e suas implementações, separando interfaces de entrada e saída.
- **Pacote de Nós** - Cada Nó possui um pacote próprio onde são definidos entrada, saída, os *theards* e desempenhos medidos.
- **Pacote de Mensagens** - Definição de tipos de mensagens e implementações usadas na comunicação de Nós.
- **Pacote de Hardware** - Rôbos, placas, SoC's, CPU, etc.

A ideia é representar os Nós como *process* da AADL, e seus *publishers* e *subscribers* como *threads*.

Usando a herança da OO da AADL, é possível implementar componentes filhos com valores de desempenho associado a uma plataforma.

Exemplo de Implementação

Nó ROS

```
process usb_cam_nd extends ros::node
features
  rgb_stream_in : in event data port ros_data::video_stream.rgb;
  rgb_image_raw_out : out event data port ros_data::Image.rgb;
end usb_cam_nd;

process implementation usb_cam_nd.impl
subcomponents
  image_broadcaster : thread imagePublisher.impl;
  usbSpinner : thread usb_cam_spinner.impl;
connections
  con1 : port image_broadcaster.pub_msg → rgb_image_raw_out;
  con2 : port rgb_stream_in → usbSpinner.rgb_stream_in;
end usb_cam_nd.impl;
```

Publiser

```
thread imagePublisher extends ros::publisher
features
  pub_msg : refined to out event data port ros_data::Image.rgb;
end imagePublisher;

thread implementation imagePublisher.impl
properties
  Period ⇒ 33333 us; -- @ 30 images/s
end imagePublisher.impl;
```

Implementação - Plataforma Odroid XU4

```
process implementation usb_cam_nd.xu4_a15 extends usb_cam_nd.impl
subcomponents
  image_broadcaster : refined to thread imagePublisher.xu4_a15;
properties
  SEI::MIPSBudget ⇒ 141.0 MIPS; -- (197 MIPS) / (1.4 IPC)
end usb_cam_nd.xu4_a15;

thread implementation imagePublisher.xu4_a15
  extends imagePublisher.impl
properties
  Compute_Execution_Time ⇒ 2319 us .. 2319 us;
  Queue_Size ⇒ 512 applies to pub_msg;
end imagePublisher.xu4_a15;
```

A ideia é representar os Nós como *process* da AADL, e seus *publishers* e *subscribers* como *threads*.

Usando a herança da OO da AADL, é possível implementar componentes filhos com valores de desempenho associado a uma plataforma.

3. Análise dos Modelos AADL

A partir das implementações de modelos do sistema podemos fazer as seguintes análises.

- **Análise de Alocação de Recursos** - Calculo de carga de CPU com base no tempo de execução e threads.
- **Análise de Orcamento de Recursos** - Verifica se a soma das demandas de todas as threads de um processo respeita o MIPSBudget

- **Análise de Carga de Barramento** - Uso do barramento com base no tamanho das mensagens e frequência.
- **Escalonamento de Threads** - Verifica estaticamente se as threads conseguem cumprir as tarefas no tempo esperado.
- **Análise em Tempo Real dos Threads** - Simula a execução real das tarefas, podendo revelar perda de prazo, mesmo quando a análise estática diz que é escalonável.

Comentários

O artigo apresenta uma forma bem eficiente de se analisar diferentes plataformas robóticas utilizando-se de modelos da biblioteca criada.

Com muitos testes e implementações de diferentes plataformas, a biblioteca se mostra bastante extensa.

Também é mostrado quais análises é possível fazer a partir dos modelos criados.

Referência

ERIC SENN; LUCIE W. J. BOURDON; DOMINIQUE BLOUIN. **Multi-Paradigm Modeling for early Analysis of ROS-based Robotic Applications using a Library of AADL Models.** Outubro 2022.