

Tutorial da ferramenta DevCompatibility

Diego Câmara Sales

01-2020 - versão 01

1 Introdução

A constante evolução tecnológica proporciona o desenvolvimento de novos componentes eletrônicos, dispositivos e plataformas computacionais que aprimoram e habilitam a realização de aplicações que previamente não eram possíveis. Com isso, os projetistas podem trocar os componentes da arquitetura de um sistema prolongando seu ciclo de vida e gerando versões que atendem aos novos requisitos de projeto.

Entretanto, realizar modificações nos componentes da arquiteturas é um desafio. Identificar os componentes que devem ser trocados, selecionar potenciais candidatos e avaliar a compatibilidade requer o uso de uma metodologia sistemática e ferramentas que auxiliem os projetistas a explorar cenários e impactos na arquitetura do sistema.

Para isso, foi desenvolvido a ferramenta *DevCompatibility* que tem por objetivo realizar a avaliação e análise de troca dos dispositivos de sensoriamento e atuação (S&A) da arquitetura do sistema modelado em AADL, gerando automaticamente potenciais cenários de evolução onde o projetista define quais os atributos de qualidade e objetivos que deseja modificar.

1.1 Considerações Iniciais

Inicialmente o projetista deve possuir os arquivos AADL do modelo arquitetural que deseja realizar a troca dos dispositivos de S&A e também a biblioteca de dispositivos candidatos modelados. Para tal, o projetista deve efetuar a declaração das características que deseja avaliar e o conjunto de propriedades do modelo AADL seguindo um conjunto de definições interpretadas pela ferramenta.

As análises do modelo arquitetural e biblioteca de dispositivos candidatos são realizados através da definição de um conjunto de atributos no escopo do modelo AADL para análise dos dispositivos candidatos compatíveis. Para isso serão apresentados a seguir um exemplo da estrutura de um modelo AADL da arquitetura do sistema e biblioteca de dispositivos utilizados no âmbito do projeto de pesquisa de veículos aéreos autônomos PROVANT, demonstrando a exploração de cenários com a troca de um dispositivo do sistema.

A ferramenta contempla a rastreabilidade dos requisitos do sistema que foram modificados e modelo arquitetural AADL. A modelagem dos requisitos da arquitetura é feita com a linguagem *ReqSpec* devido a integração com ambiente de desenvolvimento OSATE2 através do plug-in ALISA.

1.2 Modelo da arquitetura do sistema

O projetista deve indicar na ferramenta o local onde encontra-se o arquivo do modelo AADL da arquitetura do sistema que deseja realizar a exploração de cenários, avaliação e análise da troca de

componentes declarados.

A ferramenta efetua a leitura dos sistemas implementados no modelo AADL e extrai os dados dos subcomponentes, conexões e propriedades declaradas. A Figura 1 apresenta um exemplo de sistema implementado da arquitetura do sistema utilizado na versão 3.0 contendo um conjunto de subcomponentes do tipo processador, processo e dispositivos.

```
1 system implementation ProVant_3_0.impl
2 subcomponents
3   STM32F405:processor STM32F405;
4   nrf51822:processor nrf51822;
5   BEAGLEBONE:processor STM32F405;
6   RF_Firmware:process nRF51822_Firmware;
7   STM32F405_Firmware:process STM32F405_Firmware;
8   BEAGLE_Firmware:process Beagle_Firmware.impl;
9   GY85:device GY85;
10  ... (10 architectural component)
20 connections
21  C0: bus access STM32F405.uart_bus -> UART;
22  ... -- (17 connections)
23 properties
24  Actual_Processor_Binding => (reference (
25    STM32F405)) applies to STM32F405_Firmware;
26  ... -- (12 Allocations and bindings)
27 end ProVant_3_0.impl;
```

Figura 1: Exemplo de figura com figure

A ferramenta **DevCompatibility** tem como foco explorar, avaliar e analisar potenciais cenários a partir da troca dos dispositivos (*device*) da arquitetura do sistema, identificando a compatibilidade de interface física e conexões de hardware, tipo de dados, protocolos de comunicação e ligações de software.

Cada subcomponente da arquitetura possui um conjunto de características e propriedades modelados de acordo com as descrições técnicas do datasheet de cada componente. O ambiente OSATE2 disponibiliza um conjunto de propriedades para análise e uso dos plug-ins nativos, sendo possível criar novas análises a partir da criação do arquivo conjunto de propriedades (**property set**).

1.2.1 Modelagem de dispositivos

Com intuito de facilitar a representação dos conceitos e relacionamentos dos componentes AADL é apresentado o conceito central elaborado em metamodelo, apresentado por [1], para descrever os tipos de componentes e conexões entre as interfaces. A linguagem AADL fornece dois mecanismos para declarar componentes: **ComponentType**, que especifica um componente descrevendo apenas sua interface ou usando a implementação de componentes que especifica um componente, declarando estrutura interna, conforme mostrado na Figura 2.

Os componentes AADL são declarados de acordo com o seu tipo ou implementação de uma determinada categoria. A implementação de um componente pode realizar o refino do tipo do componente especificado adicionando detalhes como subcomponentes, modo de operação, conexões e caminho do fluxo de dados.

Os componentes possuem valores de propriedades associados para definir as propriedades do componente que devem ser avaliados e analisados. Na secção de características (**features**) são definidas as especificações de interface, como portas e barramentos, assim como os tipos de dados utilizado pelo componente.

A Figura 3 apresenta um exemplo de modelagem do dispositivo ADIS16480, sendo criado um pacote (**package**) contendo um conjunto de modelos que provem suporte a definição dos tipos de

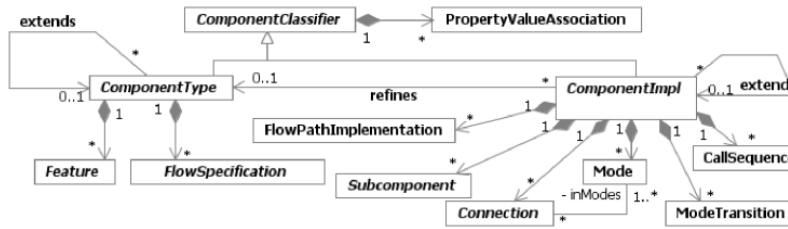


Figura 2: Metamodelo de componentes AADL[1].

dados, barramentos, propriedades físicas e elétricas utilizadas na modelagem do dispositivo (linha 3). As características definem o tipo de porta, barramentos e dados (linhas 6 e 7). O fluxo de dados é definido, neste caso o dispositivo fornece dados ao sistema em uma determinada faixa de latência (linha 9). As propriedades que definem o comportamento temporal, elétricos e preço são apresentadas nas linhas 11 a 15.

```

1 package ADIS16480
2 public
3 with provant_data, Buses::SPI, Physical_Properties, Electricity_Properties;
4 device ADIS16480
5 features
6   DOF9: out data port provant_data::Nine_Axis.impl;
7   spi_bus: requires bus access Buses::SPI::SPI.IMPL;
8 flows
9   f1: flow source DOF9 {latency => 200us .. 500us;};
10 properties
11   Dispatch_Protocol => Periodic;
12   Period => 10ms;
13   Electricity_Properties::Core_Voltage => 3.3V;
14   Electricity_Properties::Run_Current => 254.0mA;
15   SEI::price => 400.0;

```

Figura 3: Modelo AADL do dispositivo ADIS16480.

A ferramenta **DevCompatibility** realiza a leitura das características e propriedades de cada componente da arquitetura

1.2.2 Modelagem de propriedades

Os parâmetros modelados seguem as normas definidas na linguagem AADL onde são definidos no arquivo property set o conjunto de propriedades criadas para serem declaradas na modelagem dos componentes AADL. O padrão de sintaxe utilizado para leitura das propriedades é relacionado diretamente com o tipo de funcionalidade dos dispositivos e sua característica, que são classificadas de natureza física, elétrica, eletrônica e mecânica. Como exemplo, a Figura 1 apresenta as propriedades declaradas no dispositivo GY85 (Accelerometer, Compass e Gyroscope), indicando as funcionalidades e características declaradas no property set. Como regra, são declarados a funcionalidade iniciando a primeira letra em maiúsculo, utilizando “_” para separar mais de uma palavra, na sequência deve ser declarado o atributo de qualidade da mesma forma.

```

1 property set Device_Property is
2 Device_Functionality: list of
3 Physical_Properties::Device_Sensor_Type applies to(system, device);
4 Device_Type: Physical_Properties::Device_Actuator_Type applies to(device); --
5 Device_Sensor_Type : type enumeration (Sensor,Gyroscope, Compass, Accelerometer,
    Barometric_Pressure, Magnetometer, LiDar, Pitot, Camera, Sonar, Radar,
    Transponder, Radio, Hall_Sensor, MEMS);
6 Device_Actuator_Type : type enumeration (Motor, ServoMotor, BrushlessMotor,
    DirectCurrentMotor, AcMotor_2phase, AcMotor_1phase);
7 Device_Application : type enumeration (Real_time, Multimedia, Automotive,
    FlightControl, LowCost, SafeCritical, IoT, AUTOSAR, general_purpose,
    Industrial, Structural, Health, Militar, Aero);

```

Figura 4: Exemplo de figura com figure

```

17 Device_property::Power_Consume => 0.02W;
18 Device_property::Device_Functionality => (Gyroscope, Compass,
    Accelerometer, Barometric_Pressure, Magnetometer);
19 Device_property::Accelerometer_Range => (2.0g,4.0g,8.0g,16.0g);
20 Device_property::Gyroscope_Range => (125.0dps,2000.0dps);
21 Device_property::Magnetic_field_range => (1300.0uT,1300.0uT,2500.0uT);
22 Device_property::Magnetic_field_resolution => (0.3uT);
23 Device_property::Device_Software => (EKF, DigitalMotionProcessing);
24 Device_property::Device_Application => (Militar);
25 Device_property::Device_Bus_Available => (SPI);
26 Device_property::Device_Max_Bus_Bandwidth => (1000.0 Kbitsps);
27 Device_property::Device_Noise_Accelerometer => 100.0 ug_sqrHz;
28 Device_property::Device_Noise_gyroscope => 3.0 mdps_sqrHz;
29 Device_property::ZRO_25C_GYRO => (-5.0 dps, 5.0 dps);

```

Figura 5: Exemplo de figura com figure

```

1 system requirements reqs for ProVant_3_0 [
2   description "These are requirement for UAV"
3 requirement R1 : "UAV weight limit" [
4   val MaximumWeight = 1.2 kg
5   category Quality.Mass
6   description this " shall be within
7   weight of " MaximumWeight
8   value predicate MaximumWeight == #SEI::
9   WeightLimit
10  see goal SCFgoals.ng1]
11 requirement R2 : "UAV inlet power" for UAVSystem::US
12 [
13   val MaximumPowerBudget = 5.0 W
14   category Quality.Performance
15   compute actualvolt: Physical::
16   Voltage_Type
17   value predicate MaximumPowerBudget == #
18   SEI::PowerBudget
19   see goal SCFgoals.g2]

```

Figura 6: Exemplo

1.3 Modelagem dos requisitos

2 Ferramenta DevCompatibility

2.1 Instalação

A ferramenta foi desenvolvida para rodar como plug-in ambiente OSATE2 com intuito do projetista realizar a exploração dos cenários com base nos modelos AADL que possui. Para isso, é necessário instalar o ambiente OSATE2 de desenvolvimento seguindo o passo a passo disponível no site .

Após a instalação do Java SDK 8 e pacote de projetos do OSATE2 o projetista deve copiar o arquivo jar na pasta de plugins da ferramenta e compilar novamente a ferramenta via ECLIPSE versão oxygen.

Com intuito de facilitar a utilização da ferramenta foi criado um arquivo JAVA que pode ser executado como aplicativo para rodar em diferentes tipos de sistema operacional. No sistema operacional Windows o usuário pode rodar o aplicativo *AadlEvaluator.jar* e executar os seguintes comandos: Abrir a pasta onde se encontra o aplicativo *.jar* e executar o comando *java -jar AadlEvaluator-0.9.9.jar*.

```
C:\Users\User>cd C:\Users\User\Downloads\  
C:\Users\User\Downloads>cd AadlEvaluator-0.9.9-V2  
C:\Users\User\Downloads\AadlEvaluator-0.9.9-V2>java -jar AadlEvaluator-0.9.9.jar  
Reading logger configuration file...  
Reading images...  
Starting the GUI...
```

Figura 7: Comandos para rodar o aplicativo no Windows.

2.2 Exploração de Cenários

A primeira tela da ferramenta apresenta duas áreas de visualização das seleções e resultados. A área da esquerda é apresentada os cenários possíveis de arquitetura, de acordo com as definições escolhidas, enquanto a área da direita é apresentada as mudanças realizadas automaticamente pela ferramenta para que os cenários sejam compatíveis com a arquitetura atual, sendo apresentado na Figura 8.

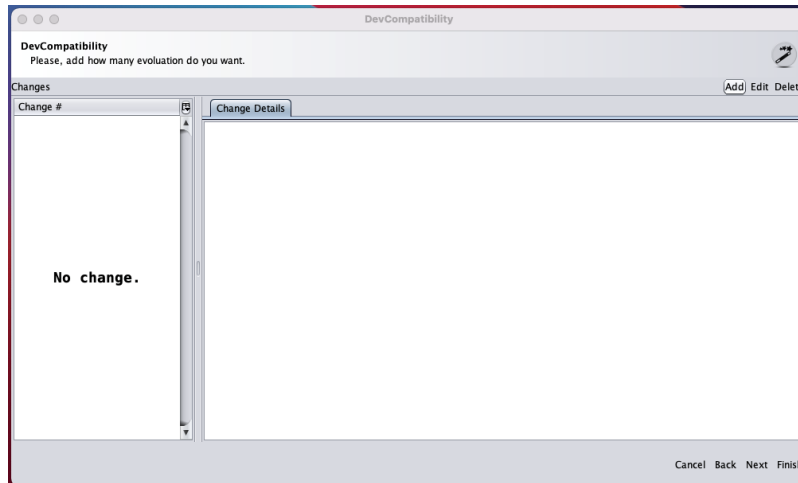


Figura 8: Tela inicial da ferramenta DevCompatibility.

O botão “Add” apresenta os três modos de operação disponíveis na ferramenta: modificações completas no sistema, troca de um componente por outro e geração automática de cenários. que podem ser utilizados para realizar modificações no modelo arquitetural, conforme apresentado na Figura 9.

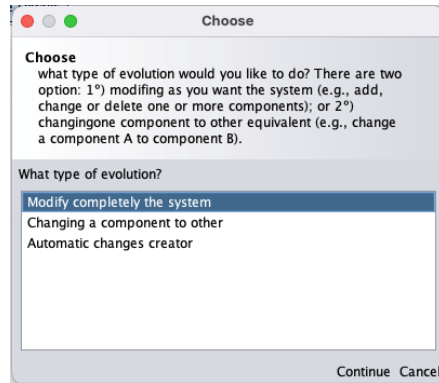


Figura 9: Seleção do tipo de exploração de projeto de arquitetura deseja realizar.

Como exemplo, será apresentado o passo a passo da troca de um dispositivo da arquitetura utilizando o modo automático. Após o usuário definir os arquivos da biblioteca que contêm os modelos AADL dos dispositivos candidatos e arquitetura do sistema que deseja efetuar a troca. A ferramenta apresenta ao projetista o levantamento de todos os componentes declarados na arquitetura selecionada, assim como os candidatos que possuem ao menos uma funcionalidade compatível. A Figura 10 apresenta os componentes encontrados no modelo AADL de uma arquitetura de sistema e os candidatos encontrados.

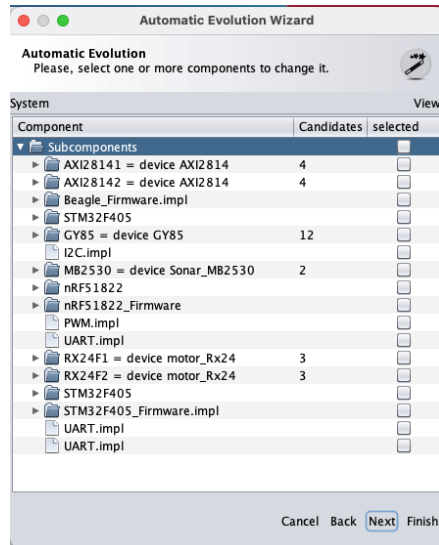


Figura 10: Seleção dos dispositivos que serão trocados.

O projetista deve selecionar quais componentes devem ser trocados e na sequência clicar no botão **“Next”**. Caso o projetista deseje visualizar previamente o modelo do componente que deseja realizar a troca, assim como o modelo de requisitos declarado que está relacionado com o

mesmo, deve selecionar um dos subcomponentes listados na ferramenta e clicar no botão “View”. A visualização dos modelos AADL e Reqspect é apresentada na Figura 11, indicando que o projetista realizou a referência dos requisitos declarados com as propriedades do componente. Caso os requisitos não sejam mostrados o projetista deve observar se o nome dos componentes e suas funcionalidades estão declaradas corretamente.

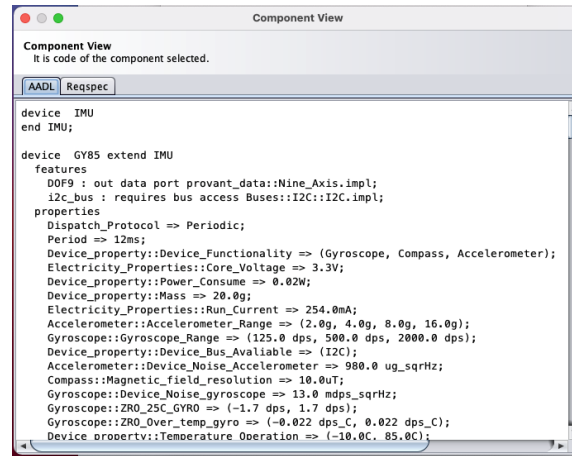


Figura 11: Visualização prévia do dispositivo selecionado em AADL.

Após clicar no botão “Next” o projetista pode definir como solucionar os problemas encontrados na conexão dos componentes com a arquitetura. Dispositivos que possuem mais de uma conexão e interface devem ser selecionados considerando, como sugestão, o uso do barramento de comunicação semelhante ao utilizado anteriormente. Neste caso, dentre os 12 dispositivos candidatos o dispositivo GY955 apresentou 2 tipos de barramentos que podem ser usados.

A ferramenta alertou de que a conexão C4 que é responsável por realizar a interface com o barramento i2c não está disponível no dispositivo candidato, tendo de escolher entre os barramentos disponíveis SPI e UART. Vale ressaltar que os barramentos disponíveis apresentados possuem interface com a arquitetura do sistema selecionado, caso não exista a ferramenta adiciona um componente que tem por função realizar a compatibilidade conforme apresentado na Figura 12.

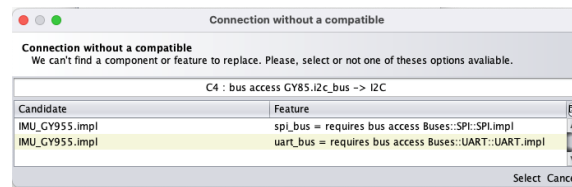


Figura 12: Seleção de barramentos compatíveis.

Na sequência a ferramenta apresenta um conjunto de alternativas de combinações de cenários indicando quais dispositivos e interfaces de hardware e software são compatíveis. Caso seja realizado modificações na interface para habilitar a compatibilidade do dispositivo com a arquitetura uma

mensagem indicando a inclusão de um dispositivo de hardware do tipo conversor e para software a realização de *wrapper*.

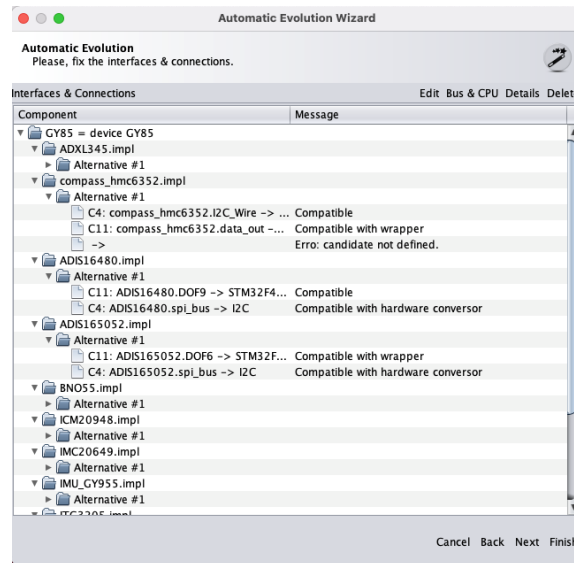


Figura 13: Relatório dos dispositivos candidatos com interface compatível.

Os sistemas que possuem mais de um processador e barramentos disponíveis na arquitetura podem ser selecionados com objetivo de efetuar o remanejamento do novo dispositivo trocado. Para isso, o projetista deve clicar no botão “Bus & CPU” para ter acesso aos componentes disponíveis e selecionar a nova configuração da arquitetura. A Figura 14 apresenta um exemplo dos processadores e barramentos disponíveis na arquitetura do projeto ProVant 3.0.

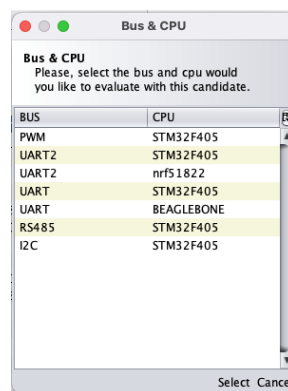


Figura 14: Lista dos barramentos e CPU disponíveis na arquitetura do sistema.

Vale ressaltar que caso não seja realizado modificações quanto aos barramentos e processador a ferramenta considera a utilização dos componentes previamente utilizados pelo componente seleci-

onado para a troca.

Caso o projetista queira visualizar mais detalhes de como a ferramenta realizou a ligação do dispositivo candidato com a arquitetura o projetista pode selecionar o componente e alternativa de cenário desejado e clicar no botão “Details” onde serão apresentados em uma janela o resumo das interfaces e conexões selecionadas automaticamente pela ferramenta.

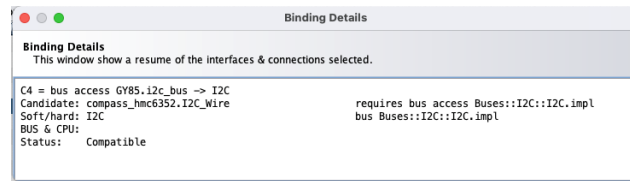


Figura 15: Detalhes de quais conexões e interfaces são compatíveis.

Após a visualização previa da compatibilidade entre a interface dos dispositivos candidatos com a arquitetura do sistema, o projetista clica em “Next” para que os cenários sejam gerados conforme é apresentado na Figura 10 a seguir. As possíveis combinações de cenários são apresentadas na esquerda da janela indicando o número de cenários (evolutions) que foram gerados em conjunto com uma mensagem contendo os tipos de modificações realizados. O projetista pode selecionar um dos cenários criados e obter mais detalhes quanto a mudança realizada na arquitetura para efetuar a compatibilidade com o dispositivo candidato e interface utilizada clicando na aba “Change Details”.

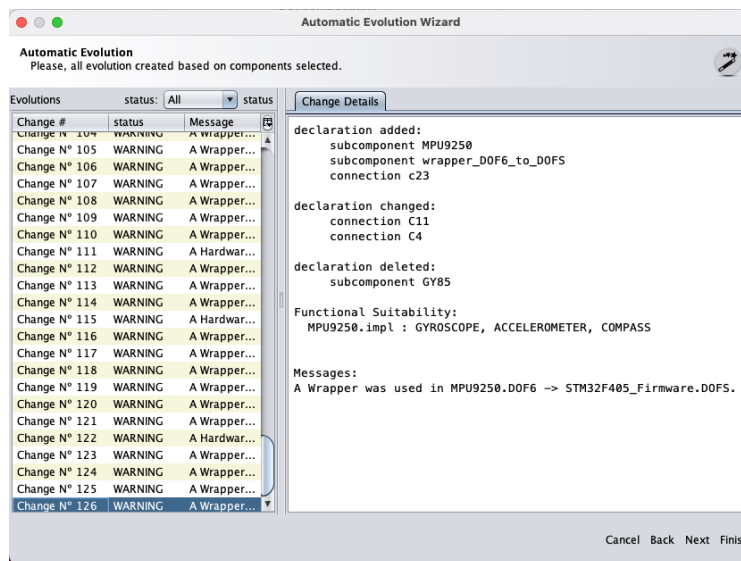
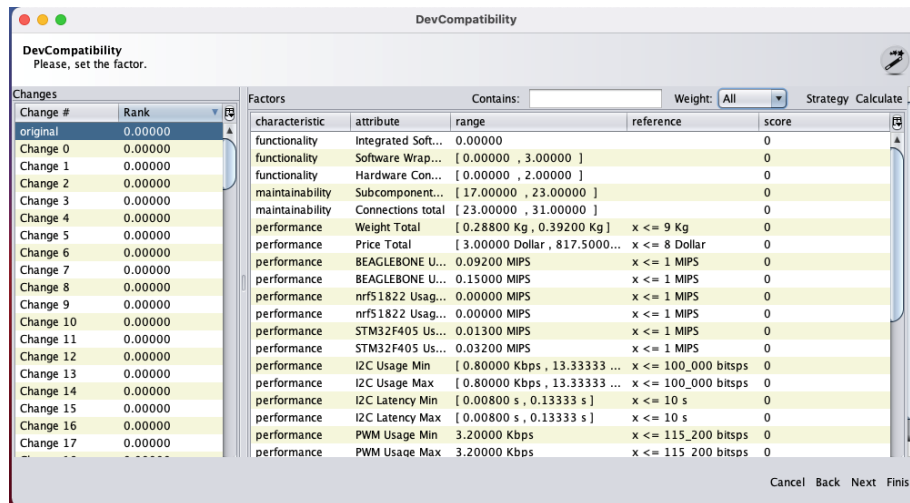


Figura 16: Lista de possíveis cenários compatíveis criados.

As modificações, características e atributos de cada cenário gerado para realizar a compatibilidade do dispositivo candidato são armazenados na ferramenta para posterior avaliação e análise realizadas nas próximas etapas clicando no botão “Next”. Após o projetista visualizar os cenários

gerados ocorre a etapa de otimização da arquitetura do sistema onde são definidos os pesos de acordo com o tipo de características e atributos de qualidade se deseja modificar na arquitetura original.



The screenshot shows the 'DevCompatibility' window with a 'Please, set the factor.' message. It contains two main panels: 'Changes' and 'Factors'.

Changes Panel:

Change #	Rank
original	0.00000
Change 0	0.00000
Change 1	0.00000
Change 2	0.00000
Change 3	0.00000
Change 4	0.00000
Change 5	0.00000
Change 6	0.00000
Change 7	0.00000
Change 8	0.00000
Change 9	0.00000
Change 10	0.00000
Change 11	0.00000
Change 12	0.00000
Change 13	0.00000
Change 14	0.00000
Change 15	0.00000
Change 16	0.00000
Change 17	0.00000

Factors Panel:

characteristic	attribute	range	reference	score
functionality	Integrated Soft...	0.00000		0
functionality	Software Wrap...	[0.00000 , 3.00000]		0
functionality	Hardware Con...	[0.00000 , 2.00000]		0
maintainability	Subcomponent...	[17.00000 , 23.00000]		0
maintainability	Connections total	[23.00000 , 31.00000]		0
performance	Weight Total	[0.28800 Kg , 0.39200 Kg]	x <= 9 Kg	0
performance	Price Total	[3.00000 Dollar , 817.5000...	x <= 8 Dollar	0
performance	BEAGLEBONE U...	0.09200 MIPS	x <= 1 MIPS	0
performance	BEAGLEBONE U...	0.15000 MIPS	x <= 1 MIPS	0
performance	nrfs1822 Usag...	0.00000 MIPS	x <= 1 MIPS	0
performance	nrfs1822 Usag...	0.00000 MIPS	x <= 1 MIPS	0
performance	STM32F405 Us...	0.01300 MIPS	x <= 1 MIPS	0
performance	STM32F405 Us...	0.03200 MIPS	x <= 1 MIPS	0
performance	I2C Usage Min	[0.80000 Kbps , 13.33333 ...	x <= 100_000 bitsps	0
performance	I2C Usage Max	[0.80000 Kbps , 13.33333 ...	x <= 100_000 bitsps	0
performance	I2C Latency Min	[0.00800 s , 0.13333 s]	x <= 10 s	0
performance	I2C Latency Max	[0.00800 s , 0.13333 s]	x <= 10 s	0
performance	PWM Usage Min	3.20000 Kbps	x <= 115_200 bitsps	0
performance	PWM Usage Max	3.20000 Kbps	x <= 115_200 bitsps	0

At the bottom right of the window are buttons: Cancel Back Next Finish.

Figura 17: Atributos de qualidade de cada cenário avaliado.

Para isso o projetista deve selecionar o tipo de atributo de qualidade que deseja definir o peso clicando duas vezes com o mouse na coluna score. O valor do peso deve ser entre 0 e 1, onde o valor maior representa maior importante para o atributo selecionado. Caso o projetista deseje utilizar uma estratégia previamente disponibilizada pela ferramenta para realizar a definição dos pesos deve ser clicado no botão **“Strategies”** e selecionar uma das estratégias conforme apresentado na Figura 18.

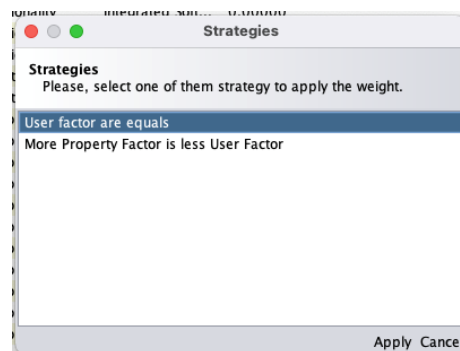


Figura 18: Definição da estratégia utilizada para calcular o ranque dos cenários.

Após a definição dos pesos o projetista clica no botão **“Calculate”** para realizar o cálculo do ranque de cada cenário gerado. Ao clicar na coluna **“Rank”** o projetista pode organizar os cenários por peso e identificar quais cenários possuem o melhor rank quando comparado com a arquitetura

inicial (original) conforme apresentado na Figura 19.

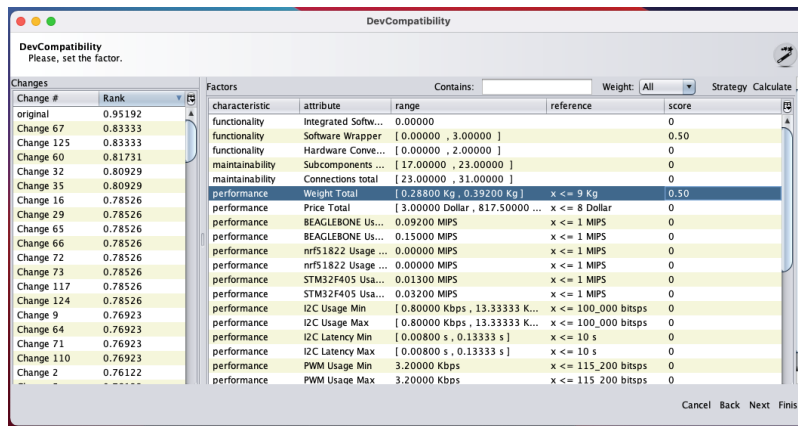


Figura 19: Definição de pesos e cálculo do ranque de cenários.

Após a classificação dos cenários por ranque de pesos o projetista pode visualizar mais detalhes quanto aos cenários que obtiveram melhor resultado clicando no botão “Next”. A Figura 20 apresenta detalhes quanto ao dispositivo candidato utilizado na substituição do dispositivo selecionado, conexões que foram modificadas, funcionalidades que foram alteradas e modificações de interface de software e hardware.

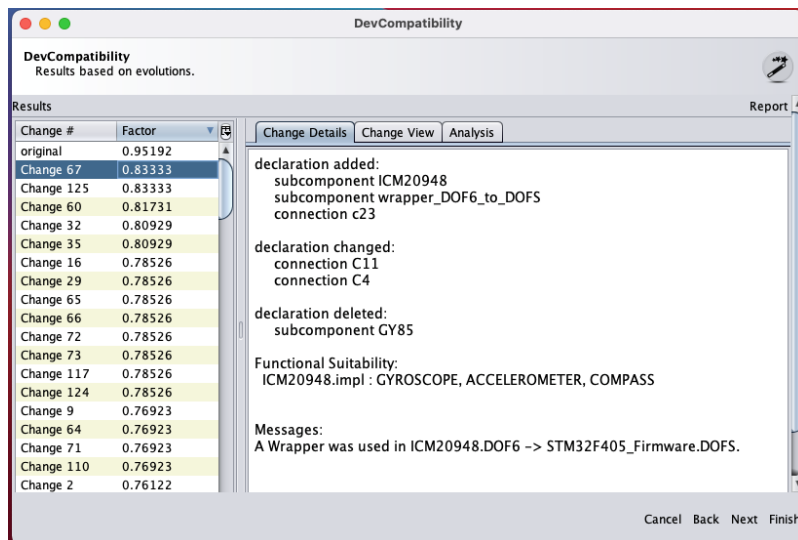


Figura 20: Detalhes das modificações realizadas em cada cenário do ranque.

O projetista pode também acessar detalhes quanto ao modelo arquitetural AADL gerado correspondente ao cenário selecionando clicando na aba “Change View” conforme apresentado na

Figura 21. O cenário selecionado é apresentado previamente em linguagem AADL apresentando as modificações na arquitetura e a troca do dispositivo.

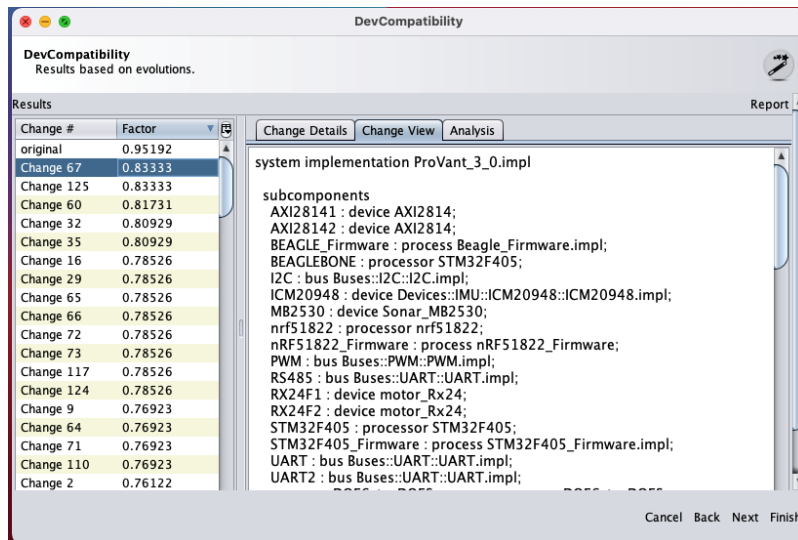


Figura 21: Visualização previa das mudanças de cada cenário gerado.

A ferramenta apresenta também uma análise comparativa do cenário selecionado com a arquitetura inicial (original) indicando as diferenças quantitativas dos atributos de qualidade avaliados indicando em porcentagem qual o impacto das modificações propostas na arquitetura para realizar a troca do dispositivo candidato selecionado. A Figura 22 apresenta o resultado da análise entre a arquitetura original e a selecionada pelo projetista após a definição dos pesos e objetivos de otimização do sistema.

The screenshot shows the 'Analysis' tab in the DevCompatibility tool, displaying a comparative analysis table:

Change #	Factor	Characteristic	Attribute	Original	Evolution	Result
original	0.95192	General	Factor	0.95192	0.83333	The factor has 0,119 less than origi...
Change 67	0.83333	functionality	Hardware Converter	0	0	The Hardware Converter are equals.
Change 125	0.83333	functionality	Integrated Software	0	0	The Integrated Software are equals.
Change 60	0.81731	functionality	Software Wrapper	0	1	The Software Wrapper has 1 more th...
Change 32	0.80929	maintainability	Connections total	23	24	The Connections total has 1 more th...
Change 35	0.80929	maintainability	Subcomponents Total	17	18	The Subcomponents Total has 1 mo...
Change 16	0.78526	performance	Accelerometer - Device Nol...	980.0 ug_sqrHz		The Accelerometer - Device Noise A...
Change 29	0.78526	performance	BEAGLEBONE Usage Max	0.150 MIPS	0.150 MIPS	The BEAGLEBONE Usage Max are eq...
Change 65	0.78526	performance	BEAGLEBONE Usage Min	0.092 MIPS	0.092 MIPS	The BEAGLEBONE Usage Min are eq...
Change 66	0.78526	performance	Compass - Magnetic Field ...	10.0uT		The Compass - Magnetic Field Resol...
Change 72	0.78526	performance	Gyroscope Noise	13.0 mdps_sqrHz		The Gyroscope Noise has 13 more t...
Change 117	0.78526	performance	I2C Latency Max	0.1333333333333333 s	0.016 s	The I2C Latency Max has 0,117 less...
Change 124	0.78526	performance	I2C Latency Min	0.1333333333333333 s	0.016 s	The I2C Latency Min has 0,117 less...
Change 9	0.76923	performance	I2C Usage Max	13.333333333333332 Kbps	1.6 Kbps	The I2C Usage Max has 11,733 less...
Change 64	0.76923	performance	I2C Usage Min	13.333333333333332 Kbps	1.6 Kbps	The I2C Usage Min has 11,733 less...
Change 71	0.76923	performance	PWM Latency Max	0.02782608695652174 s	0.02782608695652174 s	The PWM Latency Max are equals.
Change 110	0.76923	performance	PWM Latency Min	0.02782608695652174 s	0.02782608695652174 s	The PWM Latency Min are equals.
Change 2	0.76122					

Figura 22: Análise comparativa do cenário atual e selecionado.

Os resultados das análises podem ser visualizados em um ambiente interativo através de gráficos onde o projetista seleciona os cenários que deseja avaliar em mais detalhes conjuntos de atributos

e cobertura dos requisitos de projeto. Ao clicar no botão **"Report"** a ferramenta armazena na unidade de disco rígido do computador os dados obtidos das análises e gera um modelo AADL referente a cada arquitetura avaliada.

Referências

- [1] R. Behjati, T. Yue, S. Nejati, L. Briand e B. Selic, "Extending SysML with AADL concepts for comprehensive system architecture modeling," em *European Conference on Modelling Foundations and Applications*, Springer, 2011, pp. 236–252.