

Uma Proposta de Controle da Adaptação Dinâmica ao Contexto na Computação Ubíqua

por

Nelsi Warken

Orientador

Prof. Dr. Adenauer Corrêa Yamin

Mestrado em Ciência da Computação

PPGINF - Centro Politécnico - Universidade Católica de Pelotas

SUMÁRIO

- 1 Contexto do Trabalho
- 2 Motivações e Objetivos
- 3 Fundamentos Conceituais
- 4 EXEHDA-DA: Visão Geral
- 5 EXEHDA-DA: Modelagem
- 6 EXEHDA-DA: Estudos de Caso
- 7 Considerações Finais

Contexto do Trabalho

Contexto do Trabalho

- **PPGINF**: Mestrado em Ciência da Computação do Centro Politécnico da UCPEL - Linha de Pesquisa: Processamento Paralelo e Distribuído.
- **G3PD**: Grupo de pesquisa integrante do consórcio de pesquisa formado por diversas universidades do Rio Grande do Sul.
- **EXEHDA**: *Execution Environment for Higly Distributed Applications - middleware* adaptativo ao contexto, baseado em serviços, que tem como objetivo criar e gerenciar um ambiente ubíquo.

Motivações e Objetivos

Motivações

- Disseminação e aumento de demanda da Computação Ubíqua (UBICOMP).
- Importância da Adaptação ao Contexto na UBIComp.
- Contribuição para consolidação de *frameworks* para o desenvolvimento de aplicações adaptativas ao contexto.
- Colaboração com estratégias para a qualificação do controle dinâmico da adaptação, que contemple colaboração do ambiente de execução com a aplicação.

Motivações

- Disseminação e aumento de demanda da Computação Ubíqua (UBICOMP).
- **Importância da Adaptação ao Contexto na UBIComp.**
- Contribuição para consolidação de *frameworks* para o desenvolvimento de aplicações adaptativas ao contexto.
- Colaboração com estratégias para a qualificação do controle dinâmico da adaptação, que contemple colaboração do ambiente de execução com a aplicação.

Motivações

- Disseminação e aumento de demanda da Computação Ubíqua (UBICOMP).
- Importância da Adaptação ao Contexto na UBIComp.
- Contribuição para consolidação de *frameworks* para o desenvolvimento de aplicações adaptativas ao contexto.
- Colaboração com estratégias para a qualificação do controle dinâmico da adaptação, que contemple colaboração do ambiente de execução com a aplicação.

Motivações

- Disseminação e aumento de demanda da Computação Ubíqua (UBICOMP).
- Importância da Adaptação ao Contexto na UBIComp.
- Contribuição para consolidação de *frameworks* para o desenvolvimento de aplicações adaptativas ao contexto.
- Colaboração com estratégias para a qualificação do controle dinâmico da adaptação, que contemple colaboração do ambiente de execução com a aplicação.

Objetivo Central:

Conceber um modelo de controle para adaptação dinâmica de aplicações em um ambiente ubíquo, com base em dados monitorados, informações semânticas e inferências a partir das mesmas.

Objetivos Específicos:

- prover adaptação dinâmica, à nível de componente da aplicação, decidindo a melhor ação adaptativa em função da mudança de contexto;
- caracterizar, definir e instanciar o modelo ontológico a ser utilizado pelo middleware para processamento semântico;
- conceber um modelo cujos artefatos de software possam ser reutilizados e customizados para diferentes aplicações;
- propor um framework para instanciar a política de adaptação da aplicação.

Objetivos Específicos:

- prover adaptação dinâmica, à nível de componente da aplicação, decidindo a melhor ação adaptativa em função da mudança de contexto;
- caracterizar, definir e instanciar o modelo ontológico a ser utilizado pelo middleware para processamento semântico;
- conceber um modelo cujos artefatos de software possam ser reutilizados e customizados para diferentes aplicações;
- propor um framework para instanciar a política de adaptação da aplicação.

Objetivos Específicos:

- prover adaptação dinâmica, à nível de componente da aplicação, decidindo a melhor ação adaptativa em função da mudança de contexto;
- caracterizar, definir e instanciar o modelo ontológico a ser utilizado pelo middleware para processamento semântico;
- conceber um modelo cujos artefatos de software possam ser reutilizados e customizados para diferentes aplicações;
- propor um framework para instanciar a política de adaptação da aplicação.

Objetivos Específicos:

- prover adaptação dinâmica, à nível de componente da aplicação, decidindo a melhor ação adaptativa em função da mudança de contexto;
- caracterizar, definir e instanciar o modelo ontológico a ser utilizado pelo middleware para processamento semântico;
- conceber um modelo cujos artefatos de software possam ser reutilizados e customizados para diferentes aplicações;
- propor um framework para instanciar a política de adaptação da aplicação.

Fundamentos Conceituais

Computação Ubíqua

- **Computação Ubíqua:** trata do acesso ao ambiente computacional do usuário (espaço ubíquo), todo o tempo, de qualquer lugar, com qualquer dispositivo.

Características do espaço ubíquo:

- elevada distribuição, composição dinâmica, com suporte a recursos móveis, heterogêneos e mutáveis;
 - centralização no usuário;
 - tendência à invisibilidade no controle dos estados da aplicação.
-
- As **aplicações** na Computação Ubíqua são adaptativas aos elementos de contexto de seu interesse.

Computação Autônoma

A Computação Autônoma contempla quatro aspectos: auto-configuração, auto-tratamento, auto-otimização e auto-proteção. São tratados através de especificações de políticas, que podem ser:

- **Políticas de ação:** expressas na forma SE <condição> ENTÃO <ação>.
- **Políticas de objetivos:** descrevem o estado final que o sistema deve atingir em determinadas situações.
- **Políticas de função de utilidade:** especificam o quão desejáveis são estados alternativos, um com relação ao outro, através de atribuição numérica ou ordenação parcial ou total dos estados.

Ontologias

Especificação formal, explícita de uma conceituação compartilhada, onde:

- **formal** - ontologia é processável por máquina, permite raciocínio automático e possui semântica lógica formal;
- **explícita** - os conceitos e seus requisitos são definidos explicitamente, definições de conceitos, instâncias, relações, restrições e axiomas;
- **conceituação** - se refere ao modelo abstrato do mundo real;
- **compartilhada** - ontologia captura o conhecimento apresentado não apenas por um único indivíduo, mas por um grupo.

EXEHDA-DA: Visão Geral

Adaptação Multi-nível Colaborativa

O EXEHDA-DA, contempla as seguintes etapas operacionais:

- ❶ No nível da aplicação:
 - em tempo de desenvolvimento:
 - definição e criação das ontologias de Política de Adaptação da Aplicação e Contexto de Interesse da Aplicação (FWADAPT);
 - programação dos comandos adaptativos nos códigos dos componentes;
 - em tempo de execução: ativação dos comandos adaptativos, através da comunicação com o *middleware* EXEHDA;
- ❷ No nível do serviço EXEHDA-DA (*middleware*): inferência e decisão por tipo de adaptação ao contexto, a nível de componente de aplicação.

Tipos de Adaptações Dinâmicas

Adaptações dinâmicas de duas naturezas são consideradas no modelo:

- **Adaptação Não-Funcional:** atua sobre a gerência da execução distribuída, operações de mapeamento, reescalonamento, instanciação remota, migração de um determinado componente de software.
- **Adaptação Funcional:** adaptação do código da aplicação, atua sobre a seleção da implementação do componente de software em função do contexto.

Características Incorporadas no EXEHDA-DA:

- um mecanismo de adaptação utilizado, em tempo de execução, por todas aplicações e pelo próprio *middleware*;
- suporte para adaptações funcionais e não-funcionais;
- personalização dos componentes das aplicações, através de seu contexto de interesse e políticas de adaptação;
- um modelo semântico de política de adaptação da aplicação, com as regras, parâmetros e funções de utilidade para as adaptações;
- inferência da decisão de adaptação (códigos e nodos) a partir da política de adaptação da aplicação, das mudanças do contexto e das preferências do usuário;
- uma evolução incremental das especificações de políticas, regras, parâmetros e ações de adaptação;
- reutilização e customização destas especificações no desenvolvimento de novas aplicações adaptativas.

Características Incorporadas no EXEHDA-DA:

- um mecanismo de adaptação utilizado, em tempo de execução, por todas aplicações e pelo próprio *middleware*;
- **suporte para adaptações funcionais e não-funcionais**;
- personalização dos componentes das aplicações, através de seu contexto de interesse e políticas de adaptação;
- um modelo semântico de política de adaptação da aplicação, com as regras, parâmetros e funções de utilidade para as adaptações;
- inferência da decisão de adaptação (códigos e nodos) a partir da política de adaptação da aplicação, das mudanças do contexto e das preferências do usuário;
- uma evolução incremental das especificações de políticas, regras, parâmetros e ações de adaptação;
- reutilização e customização destas especificações no desenvolvimento de novas aplicações adaptativas.

Características Incorporadas no EXEHDA-DA:

- um mecanismo de adaptação utilizado, em tempo de execução, por todas aplicações e pelo próprio *middleware*;
- suporte para adaptações funcionais e não-funcionais;
- **personalização dos componentes das aplicações, através de seu contexto de interesse e políticas de adaptação;**
- um modelo semântico de política de adaptação da aplicação, com as regras, parâmetros e funções de utilidade para as adaptações;
- inferência da decisão de adaptação (códigos e nodos) a partir da política de adaptação da aplicação, das mudanças do contexto e das preferências do usuário;
- uma evolução incremental das especificações de políticas, regras, parâmetros e ações de adaptação;
- reutilização e customização destas especificações no desenvolvimento de novas aplicações adaptativas.

Características Incorporadas no EXEHDA-DA:

- um mecanismo de adaptação utilizado, em tempo de execução, por todas aplicações e pelo próprio *middleware*;
- suporte para adaptações funcionais e não-funcionais;
- personalização dos componentes das aplicações, através de seu contexto de interesse e políticas de adaptação;
- **um modelo semântico de política de adaptação da aplicação, com as regras, parâmetros e funções de utilidade para as adaptações;**
- inferência da decisão de adaptação (códigos e nodos) a partir da política de adaptação da aplicação, das mudanças do contexto e das preferências do usuário;
- uma evolução incremental das especificações de políticas, regras, parâmetros e ações de adaptação;
- reutilização e customização destas especificações no desenvolvimento de novas aplicações adaptativas.

Características Incorporadas no EXEHDA-DA:

- um mecanismo de adaptação utilizado, em tempo de execução, por todas aplicações e pelo próprio *middleware*;
- suporte para adaptações funcionais e não-funcionais;
- personalização dos componentes das aplicações, através de seu contexto de interesse e políticas de adaptação;
- um modelo semântico de política de adaptação da aplicação, com as regras, parâmetros e funções de utilidade para as adaptações;
- inferência da decisão de adaptação (códigos e nodos) a partir da política de adaptação da aplicação, das mudanças do contexto e das preferências do usuário;
- uma evolução incremental das especificações de políticas, regras, parâmetros e ações de adaptação;
- reutilização e customização destas especificações no desenvolvimento de novas aplicações adaptativas.

Características Incorporadas no EXEHDA-DA:

- um mecanismo de adaptação utilizado, em tempo de execução, por todas aplicações e pelo próprio *middleware*;
- suporte para adaptações funcionais e não-funcionais;
- personalização dos componentes das aplicações, através de seu contexto de interesse e políticas de adaptação;
- um modelo semântico de política de adaptação da aplicação, com as regras, parâmetros e funções de utilidade para as adaptações;
- inferência da decisão de adaptação (códigos e nodos) a partir da política de adaptação da aplicação, das mudanças do contexto e das preferências do usuário;
- **uma evolução incremental das especificações de políticas, regras, parâmetros e ações de adaptação;**
- reutilização e customização destas especificações no desenvolvimento de novas aplicações adaptativas.

Características Incorporadas no EXEHDA-DA:

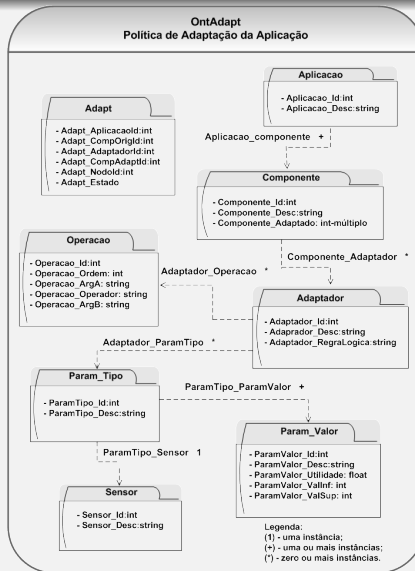
- um mecanismo de adaptação utilizado, em tempo de execução, por todas aplicações e pelo próprio *middleware*;
- suporte para adaptações funcionais e não-funcionais;
- personalização dos componentes das aplicações, através de seu contexto de interesse e políticas de adaptação;
- um modelo semântico de política de adaptação da aplicação, com as regras, parâmetros e funções de utilidade para as adaptações;
- inferência da decisão de adaptação (códigos e nodos) a partir da política de adaptação da aplicação, das mudanças do contexto e das preferências do usuário;
- uma evolução incremental das especificações de políticas, regras, parâmetros e ações de adaptação;
- **reutilização e customização destas especificações no desenvolvimento de novas aplicações adaptativas.**

EXEHDA-DA: Modelagem

Estrutura do Modelo Semântico Proposto

Ontologia de Domínio do EXEHDA-DA

- Geral: **OntUbi** - Ontologia Básica para a UbiComp. Entidades (classes), atributos e relacionamentos possíveis do ambiente ubíquo.
- Específica1: **OntContext** - Ontologia da Situação de contexto: representa os contextos coletados, contextos notificados e os contextos de interesse da aplicação.
- Específica 2: **OntAdapt** - Ontologia de Adaptação: regras, parâmetros, operações e preferências, restrições e ações de adaptação para os componentes das aplicações.
- Específica3: **OntHistAdapt** - ontologia prevista para o registro das decisões de adaptação, histórico das adaptações realizadas.

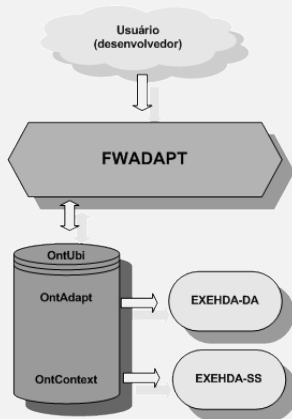


Características Incorporadas no FWADAPT

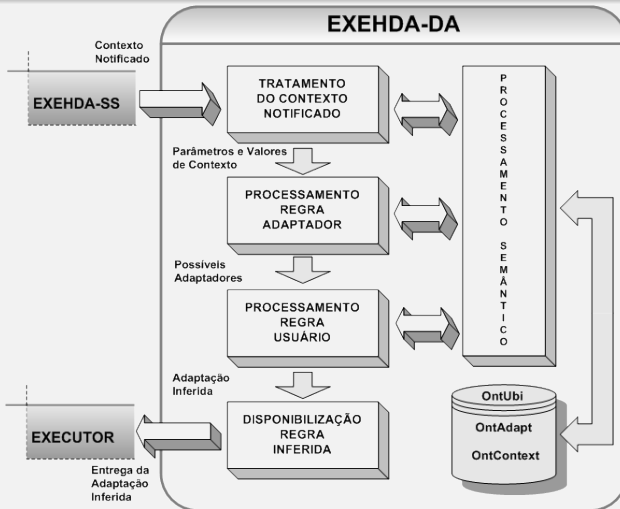
Framework FWADAPT, *framework* proposto para definição da Política de Adaptação Dinâmica dos Componentes da Aplicação:

- ter a possibilidade de ser usado em todas as aplicações que sigam o modelo de programação previsto para o EXEHDA;
- facultar procedimentos de edição e/ou extensão das definições previamente estabelecidas para as aplicações;
- contemplar recurso que facilite a reutilização de especificações já existentes no que diz respeito a adaptações, regras e operações.

FWADAPT - instanciação da OntAdapt e OntContext



Arquitetura de Software do EXEHDA-DA



Comandos Adaptativos

```
ativa (Componente_Id, Nodo_Id | Adaptador_Id);  
move (Componente_Id, Nodo_Id, Adaptador_Id);  
rescalona (Componente_Id, Adaptador_Id);  
desconecta Adaptador_Id;  
reconecta Adaptador_Id;  
seleciona Aplicacao_Id, Componente_Id, Adaptador_Id;  
noContexto Aplicacao_Id, Componente_Id, Adaptador_Id {  
    comando 1  
    comando 2  
    ...  
    comando n }
```

EXEHDA-DA: Estudos de Caso

Principais Tecnologias Envolvidas no EXEHDA-DA

- JAVA - Linguagem de Programação
- OWL - *Web Ontology Language*
- RDF - *Resource Description Framework*
- SPARQL - *Protocol and RDF Query Language*
- API JENA - *Toolkit Java* - criação e manutenção dinâmica de ontologias
- PROTÉGÉ - Editor de ontologias

Estudo de Caso 1

Acompanhamento Ubíquo de Pacientes - AUP

Objetivos da Aplicação AUP

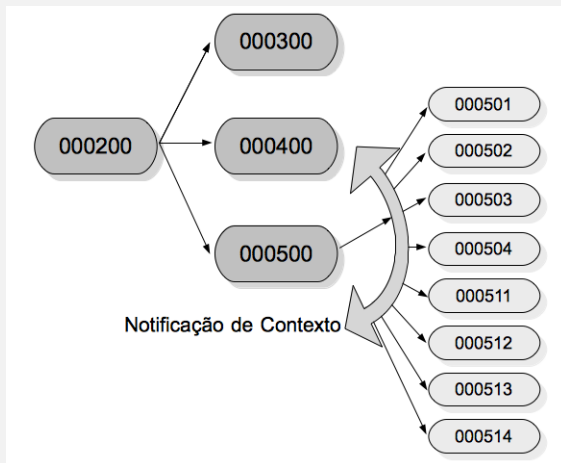
A premissa buscada é qualificar o acompanhamento de pacientes, que não estejam internados em Unidades de Tratamento Intensivo (UTI). Seus objetivos são:

- exibir dados de pacientes adquiridos dinamicamente por mecanismo de sensoreamento de sinais;
- emitir, de forma automatizada, diferentes níveis de alertas, em função dos dados sensoreados, para os agentes de saúde;
- integrar o serviço de alertas da aplicação a rede aberta de comunicação *Google Talk*;
- prover possibilidade de ter acesso, tanto a partir de dispositivo móveis, como de mesa;
- permitir acesso ubíquo ao histórico dos dados sensoreados dos pacientes por agentes de saúde.

Organização da AUP - Classe Componente na OntAdapt

Componente_Id	Componente_Desc	Componente_Adaptado
000200	Seleção de Funcionalidades e de Paciente	0
000300	Envio de Mensagens	0
000400	Histórico de Alertas de Pacientes	0
000500	Monitoramento de Pacientes	000501 000502 000503 000504 000511 000512 000513 000514
000501	Alerta Nível 1 para Interface Desktop	0
000511	Alerta Nível 1 para Interface PDA	0
000502	Alerta Nível 2 para Interface Desktop	0
000512	Alerta Nível 2 para Interface PDA	0
000503	Alerta Nível 3 para Interface Desktop	0
000513	Alerta Nível 3 para Interface PDA	0
000504	Alerta Nível 4 para Interface Desktop	0
000514	Alerta Nível 4 para Interface PDA	0

Organização da AUP - Visão Gráfica dos Componentes



Adaptações na AUP

A AUP foi concebida para explorar adaptações funcionais de duas naturezas. Estas adaptações são especificadas na OntAdapt através do FWADAPT.

As adaptações previstas são:

- 1 Adaptação funcional em função dos dados de contexto senso-reados do paciente (sinais vitais), determinando o componente de software empregado para exibição do nível de alerta.
- 2 Adaptação funcional em função de dispositivo que está sendo utilizado pelo usuário, selecionando o componente com a interface mais adequada ao dispositivo.

Adaptações na AUP - Classe Adaptador na OntAdapt

Propriedade	Valor
Adaptador_Id	001
Adaptador_Desc	Alerta Automático
Adaptador_RegraLogica	$\begin{aligned} & \text{ROUND} \left(\left(\text{CN_ContextoNotificadoSensor.CN_SensorTrad} \left(\text{CN_Sensor.Sensor_Id} = 100 \right) * \text{ParamTipo.ParamValor.ParamValor.Utilidade} \left(\text{ParamTipo.ParamValor.ParamTipo_Id} = 001 \right) \right. \right. \\ & + \text{CN_ContextoNotificadoSensor.CN_SensorTrad} \left(\text{CN_Sensor.Sensor_Id} = 101 \right) * \text{ParamTipo.ParamValor.ParamValor.Utilidade} \left(\text{ParamTipo.ParamValor.ParamTipo_Id} = 002 \right) + \\ & \left. \left. \text{CN_ContextoNotificadoSensor.CN_SensorTrad} \left(\text{CN_Sensor.Sensor_Id} = 102 \right) * \text{ParamTipo.ParamValor.ParamValor.Utilidade} \left(\text{ParamTipo.ParamValor.ParamTipo_Id} = 003 \right) + 0,9 \right) + \text{Contexto_Notificado.CN_ComponentId} \right) \end{aligned}$
Adaptador_Id	002
Adaptador_Desc	Tipo deDispositivo
Adaptador_RegraLogica	$\begin{aligned} & \left(\text{Adapt.Adapt_CompAdaptId} \left(\text{Adapt.Adapt_AplicacaoId} = \text{Contexto_Notificado.CN_AplicacaoId}, \right. \right. \\ & \left. \left. \text{Adapt.Adapt_CompOrigId} = \text{Contexto_Notificado.CN_ComponentId}, \text{Adapt.Adapt_AdaptadorId} = 001 \right) + \right. \\ & \left. \text{CN_ContextoNotificadoSensor.CN_SensorTrad} \left(\text{CN_Sensor.Sensor_Id} = 103 \right) \right) \end{aligned}$

Adaptações na AUP - Parâmetros de Adaptação para o Adaptador de Nível de Alerta

Propriedades	Batimentos Cardíacos	Temperatura	Pressão Arterial
Param_Tipo			
ParamTipo_Id	001	002	003
ParamTipo_Desc	Batimentos Cardíacos	Temperatura	Pressão Arterial
Param_Valor			
ParamValor_Id	001	002	003
ParamValor_Desc	Utilidade BC	Utilidade T	Utilidade PA
ParamValor_Utilidade	0.9	0.8	1.8
ParamTipo_Sensor			
Sensor_Id	100	101	102

Adaptações na AUP - Relacionamentos da OntAdapt

Relacionamento Aplicacao_Componente:

Aplicacao_Id = 100

Componente_Id = 000200, 000300, 000400, 000500, 000501, 000502,
000503, 000504, 000511, 000512, 000513, 000514

Relacionamento Componente_Adaptador:

Componente_Id = 000500 e Adaptador_Id = 001

Componente_Id = 000500 e Adaptador_Id = 002

Relacionamento Adaptador_ParamTipo:

Adaptador_Id = 001 para ParamTipo_Id = 001, 002, 003

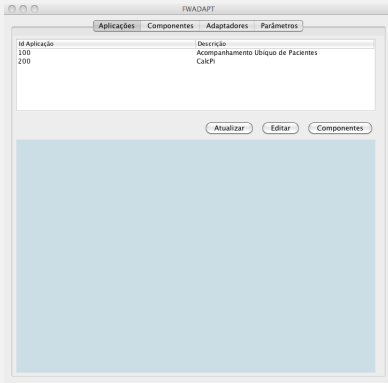
Relacionamento ParamTipo_Sensor:

ParamTipo_Id para Sensor_Id: 001 para 100, 002 para 101, 003 para 102

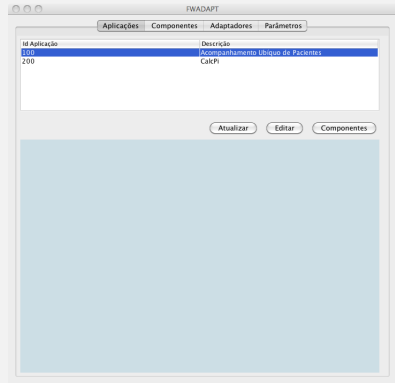
Relacionamento ParamTipo_Paramvalor:

ParamTipo_Id para ParamValor_Id: 001 para 001, 002 para 002, 003 para 003

Adaptações na AUP - FWADAPT na instanciação da OntAdapt - 1

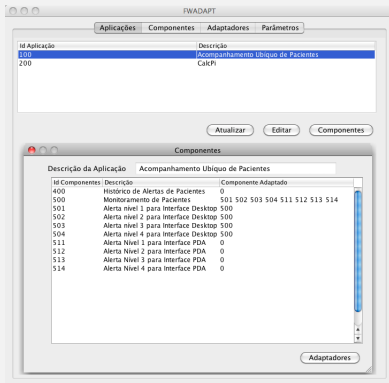


FWADAP - Aplicações

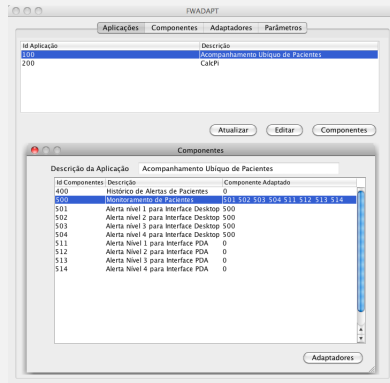


Seleção da Aplicação AUP

Adaptações na AUP - FWADAPT na instanciação da OntAdapt - 2

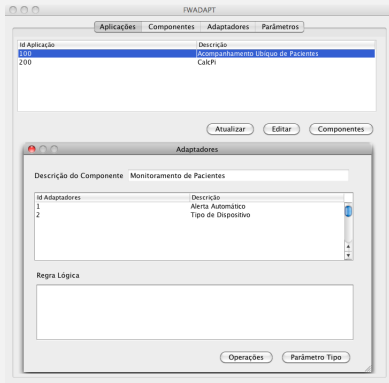


Componentes da Aplicação

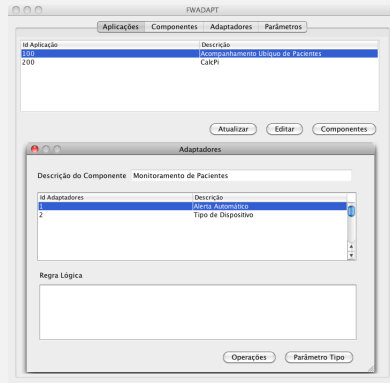


Componente Selecionado

Adaptações na AUP - FWADAPT na instanciação da OntAdapt - 3

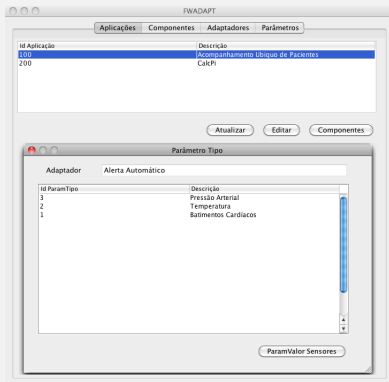


Adaptadores do Componente 000500

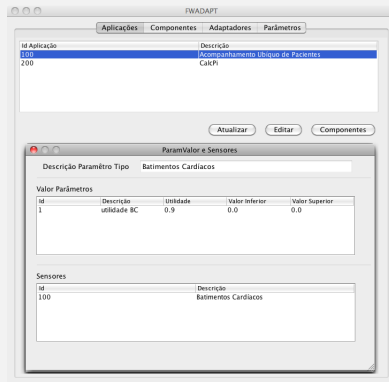


Seleção do Adaptador de Nível Automático

Adaptações na AUP - FWADAPT na instanciação da OntAdapt - 4



Tipos de Parâmetros para o Adaptador Alerta Automático



Valores e Sensores de Parâmetros do tipo Batimentos Cardíacos

Processamento de Regras de Adaptação para AUP - Níveis de Alertas Automáticos

- Nível Alerta = 1: sinais normais. Interface de nível 1 (verde) aplicação com identificação e sinais do paciente: nome, especialidade, BC, T e PA.
- Nível alerta = 2: início de problema. BC ou T fora do normal. Interface de nível 2 (amarelo).
- Nível alerta = 3: problema médio. Apenas PA fora do normal, ou BC e T fora do normal. Interface de nível 3 (laranja) e mensagem gtalk para enfermeira.
- Nível alerta = 4: alerta máximo. PA fora do normal e T e/ou BC fora normal. Interface nível 4 (vermelho) com mensagem gtalk para enfermeira e opção de envio de mensagem para médicos.

Processamento de Regras de Adaptação - Contexto Notificado Nível de Alerta

Classe Contexto_Notificado

Propriedade	Valor
CN_Id	001
CN_AplicacaoId	100
CN_ComponenteId	000500
CN_AdaptadorId	001
Usuario_Id	300

Relacionamento CN.ContextoNotificadoSensor

Propriedades	Batimentos Cardíacos	Temperatura	Pressão Arterial
CN.ContextoSensorId	001	002	003
Sensor_Id	100	101	102
CN.SensorValor	200	37	180,100
CN.SensorTrad	1	0	1
CN.NodoId	1	1	1

Processamento de Regras de Adaptação - Definição do Nível de Alerta Automático

Trad. T	Util. T	Trad. BC	Util. BC	Trad. PA	Util. PA	Total	+ 0.9	Alerta
0	0	0	0	0	0	0	0.9	1
0	0	1	0.9	0	0	0.9	1.8	2
0	0	0	0	1	1.8	1.8	2.7	3
0	0	1	0.9	1	1.8	2.7	3.6	4
1	0.8	0	0	0	0	0.8	1.7	2
1	0.8	1	0.9	0	0	1.7	2.6	3
1	0.8	1	0.9	1	1.8	3.5	4.4	4
1	0.8	0	0	1	1.8	2.6	3.5	4

Processamento da regra do Adaptador 001 - Nível Automático Alerta

Projena - NetBeans IDE 6.7.1

Arquivos | Serviços | ...java | ParmValorSensores.java | Regras.java | **OntAdapt_reg.java** | ImportaB...

Proj... | FWADAPT1.0 | **Projena** | Pacotes de códigos-fonte | app | Acessar.java | Adaptadores.java | Aplicacoes.java | Componentes.java | Fwadapt.java | ImportaBD.java

main - Navegador | Visualização dos membros | OntAdapt_reg | Aplicar_Regras(String Ontology) | ConsultaOntologia(String Ontology) | InferirOntologia(String Ontology) | InferirOntologia2(String Ontology) | listInstancesOntology(String Ontology) | **main(String[] args)**

```

73 String ocQuery = "PREFIX rdf: <http://www.w3.org/1999/02/22-
74 PREFIX ont: <http://www.owl-ontologies.com/Ontology12574439
75 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>" +
76 "PREFIX xsd: <http://www.w3.org/2000/XMLSchema#>" +
77 "SELECT * " +
78 "WHERE " +
79 "{ ?cn rdf:type ont:Contexto_Notificado . " +
80 " ?cn ont:CN_Id " + cnId + " . " +
81 " ?cn ont:CN_AplicacaoId ?vapp . " +
82 " ?cn ont:CN_ComponenteId ?vcomp . " +
83 " ?cn ont:CN_AdaptadorId ?vadapt . " +
84 " ?cn ont:CN_Usuario ?vuser . " +
85 " ?vuser ont:Usuario_Id ?userid . " +

```

Saída - Projena (run)

```

run:
CN_Id: 1
CN_AplicacaoId: 100
CN_ComponenteId: 500
CN_AdaptadorId: 1
CN_UsuarioId: 300
SensorId: 102 Utilidade: 1.8 SensorTrad: 1
SensorId: 101 Utilidade: 0.8 SensorTrad: 0
SensorId: 100 Utilidade: 0.9 SensorTrad: 1
Resultado: 504

```

Processamento de Regras de Adaptação - Contexto Notificado Tipo de Dispositivo

Propriedade	Valor
-------------	-------

Contexto_Notificado

CN_Id	002
CN_Aplicacaoid	100
CN_Componenteld	000500
CN_AdaptadorId	002
Usuario_Id	300

CN_ContextoNotificadoSensor

CN_ContextoSensorId	004
Sensor_Id	103
CN_SensorValor	0
CN_SensorTrad	10
CN_Nodold	1

Processamento da Regra do Adaptador 002 - Tipo de Dispositivo

ProjJena - NetBeans IDE 6.7.1

Arquivos | Serviços | ...java | ParmValorSensores.java | Regras.java | **OntAdapt_reg.java** | ImportaB...

Proj... | FWADAPT1.0 | **ProjJena** | Pacotes de códigos-fonte | app | Acessar.java | Adaptadores.java | Aplicacoes.java | Componentes.java | Fwadapt.java | ImportaBD.java | **OntAdapt_reg.java**

Navegador | Visualização dos membros | OntAdapt_reg | Aplicar_Regras(String Ontology) | ConsultaOntologia(String Ontology) | InferirOntologia(String Ontology) | InferirOntologia2(String Ontology) | listInstancesOntology(String Ontology) | main(String[] args)

```

169
170 String ocQuery = "PREFIX rdf: <http://www.w3.org/1999/02/22-
171 "PREFIX ont: <http://www.owl-ontologies.com/Ontology12574439
172 "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>" +
173 "PREFIX xsd: <http://www.w3.org/2000/XMLSchema#>" +
174 "SELECT * " +
175 "WHERE " +
176 "{ ?cn rdf:type ont:Contexto_Notificado . " +
177 " ?cn ont:CN_Id " + cnId + " . " +
178 " ?cn ont:CN_AplicacaoId ?vapp . " +
179 " ?cn ont:CN_ComponenteId ?vcomp . " +
180 " ?cn ont:CN_AdaptadorId ?vadapt . " +
181 " ?cn ont:CN_Usuario ?vuser . " +
182 " ?vuser ont:Usuario_Id ?userid . " +

```

Saída - ProjJena (run)

```

run:
CN_Id: 2
CN_AplicacaoId: 100
CN_ComponenteId: 500
CN_AdaptadorId: 2
CN_UsuarioId: 300
CN_SensorTrad: 10
Resultado: 514

```

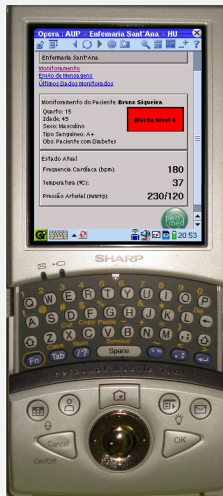
Processamento de Regras de Adaptação - Classe Adapt e Comando Adaptativo

Propriedade	Adaptação 1	Adaptação 2
Adapt_AplicacaoId	100	100
Adapt_CompOrigId	000500	000500
Adapt_AdaptadorId	001	002
Adapt_CompAdaptId	000504	000514
Adapt_NodId	001	001
Adapt_Estado	0	1

Comando Adaptativo no código do Componente 000500

```
sync  
noContexto 100, 000500 {  
  seleciona 001;  
  ativa 002;  
}
```

Processamento de Regras de Adaptação para AUP: Alerta Nível 4 para PDA



AUP - Alerta Nível 1 (Desktop e PDA)

Acompanhamento Ubíquo de Pacientes

Projeto PERTMED - Hospital Universitário

Enfermaria Sant'Ana

10/03/2010

10:30 h

Monitoramento do Paciente: Bruno Siqueira

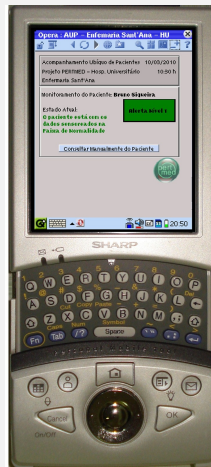
Estado Atual

O paciente está com os dados sensoreados na Faixa de Normalidade

Alerta Nível 1

Consulta Manual do Paciente: Bruno Siqueira

Consultar

AUP - Alerta Nível 2 (Desktop e PDA)

Acompanhamento Ubíquo de Pacientes

Projeto PERTMED - Hospital Universitário

Enfermaria Sant'Ana

10/03/2010

12:30 h

Monitoramento do Paciente: Bruno Siqueira

Dados do Paciente

Quarto: 15

Idade: 45

Sexo: Masculino

Tipo Sanguíneo: A+

Obs: Paciente com Diabetes

Estado Atual

Frequência Cardíaca (bpm)

115

Temperatura (°C)

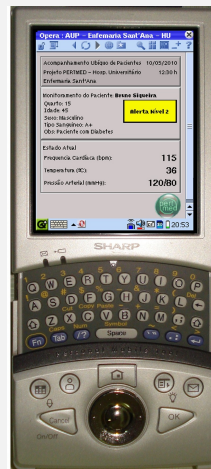
36

Pressão Arterial (mmHg)

120/80

Alerta Nível 2





AUP - Alerta Nível 3 (Desktop e PDA)

Acompanhamento Ubíquo de Pacientes

Projeto PERTMED - Hospital Universitário

Enfermaria Sant'Ana

10/03/2010

14:30 h

Monitoramento do Paciente: **Bruno Siqueira**

Dados do Paciente

Quarto: 15

Idade: 45

Sexo: Masculino

Tipo Sanguíneo: A+

Obs: Paciente com Diabetes

Estado Atual

Frequência Cardíaca (bpm)

130

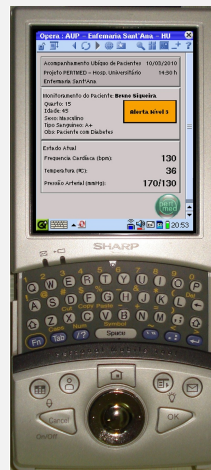
Temperatura (°C)

36

Pressão Arterial (mmHg)

170/130

Alerta Nível 3

AUP - Alerta Nível 4 (Desktop e PDA)

Acompanhamento Ubíquo de Pacientes
 Projeto PERTMED - Hospital Universitário
 Enfermaria Sant'Ana

10/03/2010
 16:30 h

Monitoramento do Paciente: **Bruno Siqueira**

Dados do Paciente
 Quarto: 15
 Idade: 45
 Sexo: Masculino
 Tipo Sanguíneo: A+
 Obs: Paciente com Diabetes

Estado Atual
 Frequência Cardíaca (bpm)
180
 Temperatura (°C)
37
 Pressão Arterial (mmHg)
230/120

Envio de Mensagens
☐ Médico Responsável
☐ Médico Plantonista
 Enviar

Alerta Nível 4

Frequência Cardíaca
 16:50 - 180
 16:45 - 130
 16:40 - 115
 16:35 - 145
 16:30 - 125
 16:25 - 120
 16:20 - 130
 16:15 - 115
 16:10 - 120
 16:05 - 115

Temperatura
 16:30 - 37
 16:00 - 36
 15:30 - 36
 15:00 - 37
 14:30 - 37
 14:00 - 37
 13:30 - 37
 13:00 - 37
 12:30 - 36
 12:00 - 36

Pressão Arterial
 16:30 - 220/230
 14:30 - 170/130
 12:30 - 120/80
 10:30 - 140/100
 08:30 - 150/90
 06:30 - 150/90
 04:30 - 140/90
 02:30 - 140/90
 00:30 - 140/100
 22:30 - 160/100

permed



Estudo de Caso 2

Alocação Dinâmica de Recursos - ADR

Objetivos da ADR

Objetivo central: explorar de modo oportunista os recursos computacionais em uma infraestrutura distribuída.

Objetivos específicos:

- ser possível selecionar o recurso computacional mais adequado, segundo um critério de adaptação, entre os disponíveis;
- permitir que a distribuição das computações possa ser iniciada, independentemente de ter processadores para todas as tarefas paralelas previstas;
- prover tomada de decisão de escalonamento em dois níveis: considerando o estado da infraestrutura computacional e as preferências do usuário.

Ambiente Computacional de Testes da ADR

Contempla quatro tipos de software:

- **aplicação paralela CalcPi** - calcula o número π pelo método Monte Carlo;
- **aplicação CpuSteal** - atua como um gerador de cargas, operando em ciclos de ativação e desativação;
- **aplicação LoadGen** - produz uma descrição de carga computacional. São gerados tempos de ativação e desativação de processador, baseado em uma distribuição de probabilidades exponencial, para ser utilizada pelo CpuSteal;
- **aplicação MemSteal** - promove a ocupação de memória. Se vale da instanciação de matrizes para gerar demanda de memória nos nodos processadores.

Premissas de Teste da ADR

A aplicação, cujas computações serão escalonadas, é do tipo sintético, e tem por objetivo central permitir a exploração dos recursos do EXEHDA-DA quando do controle das **adaptações não-funcionais**.

Duas arquiteturas paralelas do tipo *Cluster* constituem a grade computacional da infraestrutura de testes.

O critério geral é disponibilizar para a aplicação CalcPi, os nodos com maior poder computacional e menor nível de ocupação de processador e memória. Inicialmente serão alocados os nodos do *cluster* de preferência do usuário, uma vez esgotados os mesmos, serão escalonados os nodos melhor qualificados de outro *cluster*.

Organização da CalcPi - Classes Componente e Aplicacao da OntAdapt

Componente_Id	Componente_Desc	Componente_Adaptado
000700	Distribui tarefas nos nodos	0
000800	Algoritmo de cálculo do π - Monte Carlo	0

Classe Aplicacao

Aplicacao_Id = 200

Aplicacao_Desc = Cálculo do Pi - CalcPi

Relacionamento Aplicacao_Componente

Componentes 000700 e 000800 para a aplicação 200.

Organização da CalcPi - *Cluster H3P*

Composição do Cluster H3P

Nodo	Frequência (MHz)	Memória (MB)	Ocupação CPU (%)	Ocupação Mem. (%)
10001	1680	185	xxxxx	xxxxx
10002	2020	185	55,22	22,45
10003	2020	185	80,12	30,15
10004	2019	185	40,02	10,10
10005	2019	185	89,30	20,06
10006	1680	185	20,05	30,08

Organização da CalcPi - *Cluster* LANGTON

Composição do Cluster LANGTON

Nodo	Frequência (MHz)	Memória (MB)	Ocupação CPU (%)	Ocupação Mem. (%)
20001	1680	251	xxxxxx	xxxxxx
20002	1600	251	80,09	30,08
20003	1600	251	85,23	35,01
20004	1600	251	0,00	0,00
20005	1600	251	20,02	10,21
20006	1600	251	64,03	35,43
20007	1600	251	51,30	29,90
20008	1600	251	30,26	30,13
20009	1600	251	59,95	28,98
20010	1600	251	61,45	54,10

Adaptações na CalcPi - Classe Adaptador para Escalonamento de Nodos

Propriedade	Valor
Adaptador_Id	050
Adaptador_Desc	Escalonador de nodos disponíveis
Adaptador_RegraLogica	$ \begin{aligned} & \text{busca } ((\text{Contexto_Notificado.CN_Id} = \text{notificacao}) \text{ e } (\text{preferencia} = \\ & (\text{CN_ContextoNotificadoSensor.CN_Nodold} \geq \text{CN_Usuario.Usuario_ClusterInf} \text{ e } \leq \\ & \text{CN_Usuario.Usuario_ClusterSup}) \text{ e menor } ((\text{CN_ContextoNotificadoSensor.CN_SensorValor} \\ & (\text{CN_Sensor.Sensor_Id} = 104) * \text{ParamTipo_ParamValor.ParamValor_Utilidade } (\text{Param-} \\ & \text{Tipo_ParamValor.ParamTipo_Id} = 051)) + (\text{CN_ContextoNotificadoSensor.CN_SensorValor} \\ & (\text{CN_Sensor.Sensor_Id} = 105) * \text{ParamTipo_ParamValor.ParamValor_Utilidade } (\text{Param-} \\ & \text{Tipo_ParamValor.ParamTipo_Id} = 052)) + (\text{CN_ContextoNotificadoSensor.CN_SensorValor} \\ & (\text{CN_Sensor.Sensor_Id} = 106) * \text{ParamTipo_ParamValor.ParamValor_Utilidade } (\text{Param-} \\ & \text{Tipo_ParamValor.ParamTipo_Id} = 053))) \end{aligned} $

Adaptações na CalcPi - Tipos e Valores de Parâmetros de Adaptação

Propriedades	Ciclagem de CPU	Carga de CPU	Memória Ocupada
--------------	-----------------	--------------	-----------------

Param_Tipo

ParamTipo_Id	051	052	053
ParamTipo_Desc	Ciclagem CPU	Carga CPU	Ocupação Memória

Param_Valor

ParamValor_Id	151	152	153
ParamValor_Desc	Utilidade Ciclo CPU	Utilidade carga CPU	Utilidade Mem Ocup
ParamValor_Utilidade	0.2	0.8	0.5

ParamTipo_Sensor

Sensor_Id	104	105	106
-----------	-----	-----	-----

Adaptações na CalcPi - Relacionamentos Escalonador Nodos Disponíveis (050)

Relacionamento Componente_Adaptador:

Componente_Id = 000800 e Adaptador_Id = 050

Relacionamento Adaptador_ParamTipo:

Adaptador_Id = 050 para ParamTipo_Id = 051, 052, 053

Relacionamento ParamTipo_Sensor:

ParamTipo_Id para Sensor_Id: 051 para 104, 052 para 105, 053 para 106

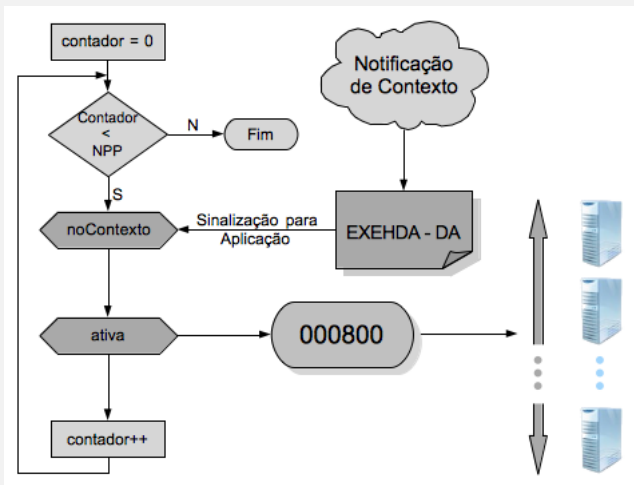
Relacionamento ParamTipo_Paramvalor:

ParamTipo_Id para ParamValor_Id: 051 para 151, 052 para 152, 053 para 153

Adaptações na CalcPi - Comando Adaptativo do Componente 000700

```
int contador = 0;
while (contador < NPP) // NPP=Número Processos Paralelos
{
    sync    noContexto 200, 000700
    {
        ativa 000800, 050;
    }
    contador++;
}
```

Adaptações na CalcPi - Escalonamento do Algoritmo do Cálculo do π



Processamento da Regra de Adaptação para CalcPi - Classe Contexto Notificado

Propriedade	Valor
-------------	-------

Contexto_Notificado

CN_Id	003
CN_AplicacaoId	200
CN_ComponentId	000800
CN_AdaptadorId	050

CN_Usuario

Usuario_Id	300
Usuario_ClusterInf	10002
Usuario_ClusterSup	10006

Processamento da Regra de Adaptação - Contexto Notificado Sensores e nodos

CN_ContextoNotificadoSensor

Propriedade	Frequência	Carga CPU	Carga Mem.
-------------	------------	-----------	------------

CN_ContextoSensorId	005	006	007
Sensor_Id	104	105	106
CN_SensorValor	2019	40,02	10,10
CN_SensorTrad	0	0	0
CN_Nodold	10004	10004	10004

CN_ContextoSensorId	008	009	010
Sensor_Id	104	105	106
CN_SensorValor	1680	20,05	30,08
CN_SensorTrad	0	0	0
CN_Nodold	10006	10006	10006

Processamento da Regra de Adaptação - Contexto Notificado Sensores e nodos (cont. ...)

CN_ContextoSensorId	011	012	013
Sensor_Id	104	105	106
CN_SensorValor	1600	0,00	0,00
CN_SensorTrad	0	0	0
CN_Nodold	20004	20004	20004
CN_ContextoSensorId	014	015	016
Sensor_Id	104	105	106
CN_SensorValor	1600	20,02	10,21
CN_SensorTrad	0	0	0
CN_Nodold	20005	20005	20005
CN_ContextoSensorId	017	018	019
Sensor_Id	104	105	106
CN_SensorValor	1600	30,26	30,13
CN_SensorTrad	0	0	0
CN_Nodold	20008	20008	20008

Processamento Regra de Adaptação - seleção de nodo para CalcPi

Nodo	Freq.	Util. Freq.	CPU	Util. CPU	Mem.	Util. Mem.	Total
20004	1600	0.2	0.00	0.8	0.00	0.5	320.00
20005	1600	0.2	20.02	0.8	10.21	0.5	341.13
20008	1600	0.2	30.26	0.8	30.13	0.5	359.28
10006	1680	0.2	20.05	0.8	30.08	0.5	367.08
10004	2019	0.2	40.02	0.8	10.10	0.5	440.86

Resultado da regra de adaptação:

nodo 10006, melhor nodo disponível do
cluster de preferência do usuário, H3P.

Processamento Regra de Adaptação para CalcPi - Classe Adapt

Propriedade	Adaptação Inferida
Adapt_AplicacaoId	200
Adapt_CompOrigId	000800
Adapt_AdaptadorId	051
Adapt_CompAdaptId	000800
Adapt_NodId	10006
Adapt_Estado	1

O EXEHDA-DA registra a adaptação inferida na classe Adapt, e notifica a aplicação CalcPi para que a ação seja executada.

A ação resultante é executada no momento em que é disparado o comando adaptativo ativa, colocado pelo programador no código do componente da aplicação.

Considerações Finais

Trabalhos Relacionados ao EXEHDA-DA

Características consideradas na comparação dos trabalhos com o EXEHDA-DA:

- 1 Adaptação funcional da aplicação e/ou do *middleware*;
- 2 Adaptação não-funcional;
- 3 Controle da adaptação externo à aplicação;
- 4 Modelagem Semântica para a política da adaptação da aplicação;
- 5 Dispositivos Móveis;
- 6 Reutilização de políticas a partir de um catálogo;
- 7 Tratamento autônomo da adaptação baseado em regras;
- 8 Função de Utilidade.

MODELOS	01	02	03	04	05	06	07	08
1. CARISMA	+		+		+	+	+	+
2. CHISEL		+	+				+	
3. QUO	+		+			+		
4. RAINBOW	+		+			+	+	
5. MADAM	+	+	+		+		+	+
6. PROTEUS	+		+	+			+	
7. SECAS	+		+		+		+	
8. EXEHDA-DA	+	+	+	+	+	+	+	+

- 1 - Adaptação funcional da aplicação e/ou do *middleware*;
- 2 - Adaptação não-funcional;
- 3 - Controle da adaptação externo à aplicação;
- 4 - Modelagem Semântica para a política da adaptação da aplicação;
- 5 - Dispositivos Móveis;
- 6 - Reutilização de políticas a partir de um catálogo;
- 7 - Tratamento autônomo da adaptação baseado em regras;
- 8 - Função de Utilidade.

Principais Comentários quanto aos Trabalhos Relacionados

- As funções de utilidade proporcionam um maior ajuste de adaptação e uma maior adequação às necessidades do usuário, não sendo considerada na maioria dos projetos.
- O EXEHDA-DA permite ao desenvolvedor da aplicação incluir novas adaptações, novas condições, devido à política da aplicação ser mantida externamente.
- Alguns dos trabalhos tratam apenas um tipo de adaptação, isto é, funcional ou não-funcional.
- Outros não fornecem a possibilidade de manutenção, incremental ou não, nas regras e políticas das aplicações.
- Quanto ao uso de modelo semântico, pela sua expressividade e possibilidade de reutilização e padronização, julgamos que proporciona uma maior facilidade ao desenvolvedor na definição do perfil da aplicação.

Principais Contribuições - Modelo Ontológico

- participação na criação da ontologia ONTUBI, que representa o ambiente ubíquo gerenciado pelo *middleware* EXEHDA, proporcionando uma visão única do modelo ontológico para o G3PD;
- emprego de linguagem de alto nível tipo OWL para potencializar a disponibilização e compartilhamento das informações do modelo ontológico entre os diversos serviços;
- permitir inferências e consultas lógicas em tempo de execução.

Principais Contribuições - Modelo Ontológico

- participação na criação da ontologia ONTUBI, que representa o ambiente ubíquo gerenciado pelo *middleware* EXEHDA, proporcionando uma visão única do modelo ontológico para o G3PD;
- emprego de linguagem de alto nível tipo OWL para potencializar a disponibilização e compartilhamento das informações do modelo ontológico entre os diversos serviços;
- permitir inferências e consultas lógicas em tempo de execução.

Principais Contribuições - Modelo Ontológico

- participação na criação da ontologia ONTUBI, que representa o ambiente ubíquo gerenciado pelo *middleware* EXEHDA, proporcionando uma visão única do modelo ontológico para o G3PD;
- emprego de linguagem de alto nível tipo OWL para potencializar a disponibilização e compartilhamento das informações do modelo ontológico entre os diversos serviços;
- permitir inferências e consultas lógicas em tempo de execução.

Principais Contribuições - *Framework* FWADAPT

- permite instanciação individualizada de aplicações, componentes, adaptadores, parâmetros, regras, critérios de utilidade e operações, bem como o relacionamento entre as classes;
- faculta inclusão incremental de novas aplicações, componentes, adaptações, parâmetros e operações;
- contribui no desenvolvimento de aplicações adaptativas, através das funcionalidades de inspeção, edição e inclusão de Políticas de Adaptação.

Principais Contribuições - *Framework* FWADAPT

- permite instanciação individualizada de aplicações, componentes, adaptadores, parâmetros, regras, critérios de utilidade e operações, bem como o relacionamento entre as classes;
- **faculta inclusão incremental de novas aplicações, componentes, adaptações, parâmetros e operações;**
- contribui no desenvolvimento de aplicações adaptativas, através das funcionalidades de inspeção, edição e inclusão de Políticas de Adaptação.

Principais Contribuições - *Framework* FWADAPT

- permite instanciação individualizada de aplicações, componentes, adaptadores, parâmetros, regras, critérios de utilidade e operações, bem como o relacionamento entre as classes;
- faculta inclusão incremental de novas aplicações, componentes, adaptações, parâmetros e operações;
- **contribui no desenvolvimento de aplicações adaptativas, através das funcionalidades de inspeção, edição e inclusão de Políticas de Adaptação.**

Principais Contribuições - Serviço EXEHDA-DA

- definição de uma arquitetura de software para o EXEHDA-DA, baseada em regras de adaptação personalizáveis por aplicação;
- concepção de uma interface integrada aos demais serviços do *middleware*;
- identificação dos comandos adaptativos a serem utilizados pelo desenvolvedor;
- gerência pró-ativa da decisão de adaptação, a qual pode ser disparada a qualquer momento, e sem intervenção do usuário, em reação às mudanças de contexto. Importante para ações emergenciais;
- faculta que seja explorada qualquer tipo de dependência de contexto mensurável;

Principais Contribuições - Serviço EXEHDA-DA

- definição de uma arquitetura de software para o EXEHDA-DA, baseada em regras de adaptação personalizáveis por aplicação;
- **concepção de uma interface integrada aos demais serviços do *middleware*;**
- identificação dos comandos adaptativos a serem utilizados pelo desenvolvedor;
- gerência pró-ativa da decisão de adaptação, a qual pode ser disparada a qualquer momento, e sem intervenção do usuário, em reação às mudanças de contexto. Importante para ações emergenciais;
- faculta que seja explorada qualquer tipo de dependência de contexto mensurável;

Principais Contribuições - Serviço EXEHDA-DA

- definição de uma arquitetura de software para o EXEHDA-DA, baseada em regras de adaptação personalizáveis por aplicação;
- concepção de uma interface integrada aos demais serviços do *middleware*;
- **identificação dos comandos adaptativos a serem utilizados pelo desenvolvedor;**
- gerência pró-ativa da decisão de adaptação, a qual pode ser disparada a qualquer momento, e sem intervenção do usuário, em reação às mudanças de contexto. Importante para ações emergenciais;
- faculta que seja explorada qualquer tipo de dependência de contexto mensurável;

Principais Contribuições - Serviço EXEHDA-DA

- definição de uma arquitetura de software para o EXEHDA-DA, baseada em regras de adaptação personalizáveis por aplicação;
- concepção de uma interface integrada aos demais serviços do *middleware*;
- identificação dos comandos adaptativos a serem utilizados pelo desenvolvedor;
- gerência pró-ativa da decisão de adaptação, a qual pode ser disparada a qualquer momento, e sem intervenção do usuário, em reação às mudanças de contexto. Importante para ações emergenciais;
- faculta que seja explorada qualquer tipo de dependência de contexto mensurável;

Principais Contribuições - Serviço EXEHDA-DA

- definição de uma arquitetura de software para o EXEHDA-DA, baseada em regras de adaptação personalizáveis por aplicação;
- concepção de uma interface integrada aos demais serviços do *middleware*;
- identificação dos comandos adaptativos a serem utilizados pelo desenvolvedor;
- gerência pró-ativa da decisão de adaptação, a qual pode ser disparada a qualquer momento, e sem intervenção do usuário, em reação às mudanças de contexto. Importante para ações emergenciais;
- **faculta que seja explorada qualquer tipo de dependência de contexto mensurável;**

Principais Contribuições - Serviço EXEHDA-DA (continuação ...)

- agiliza o desenvolvimento de aplicações adaptativas, através da independência da aplicação, em relação ao controle da adaptação, quanto à especificação da política de adaptação;
- possibilidade de inferências lógicas baseadas em semântica para a tomada de decisões de adaptação;
- emprego do conceito de utilidade como forma de expandir as alternativas de decisão geradas pela política de adaptação;
- facilidade de manutenção e reutilização das diferentes informações necessárias a operação do modelo, onde destacam-se os adaptadores, as regras de adaptação com seus parâmetros e operações, descritos e instanciados em linguagem de alto nível.

Principais Contribuições - Serviço EXEHDA-DA (continuação ...)

- agiliza o desenvolvimento de aplicações adaptativas, através da independência da aplicação, em relação ao controle da adaptação, quanto à especificação da política de adaptação;
- **possibilidade de inferências lógicas baseadas em semântica para a tomada de decisões de adaptação;**
- emprego do conceito de utilidade como forma de expandir as alternativas de decisão geradas pela política de adaptação;
- facilidade de manutenção e reutilização das diferentes informações necessárias a operação do modelo, onde destacam-se os adaptadores, as regras de adaptação com seus parâmetros e operações, descritos e instanciados em linguagem de alto nível.

Principais Contribuições - Serviço EXEHDA-DA (continuação ...)

- agiliza o desenvolvimento de aplicações adaptativas, através da independência da aplicação, em relação ao controle da adaptação, quanto à especificação da política de adaptação;
- possibilidade de inferências lógicas baseadas em semântica para a tomada de decisões de adaptação;
- **emprego do conceito de utilidade como forma de expandir as alternativas de decisão geradas pela política de adaptação;**
- facilidade de manutenção e reutilização das diferentes informações necessárias a operação do modelo, onde destacam-se os adaptadores, as regras de adaptação com seus parâmetros e operações, descritos e instanciados em linguagem de alto nível.

Principais Contribuições - Serviço EXEHDA-DA (continuação ...)

- agiliza o desenvolvimento de aplicações adaptativas, através da independência da aplicação, em relação ao controle da adaptação, quanto à especificação da política de adaptação;
- possibilidade de inferências lógicas baseadas em semântica para a tomada de decisões de adaptação;
- emprego do conceito de utilidade como forma de expandir as alternativas de decisão geradas pela política de adaptação;
- facilidade de manutenção e reutilização das diferentes informações necessárias a operação do modelo, onde destacam-se os adaptadores, as regras de adaptação com seus parâmetros e operações, descritos e instanciados em linguagem de alto nível.

Publicações Realizadas

- CIC-UCPEL 2008. Ontologias no Controle da Adaptação em Espaço Ubíquo. Pelotas, RS. Mostra de Pós-Graduação, 2008.
- ERAD 2009. Uma Contribuição ao Controle da Adaptação na Computação Ubíqua. Caxias do Sul-RS. ERAD 2009 - Fórum de Pós-Graduação, 2009.
- ERAD 2009. Explorando Sensibilidade ao Contexto com MoCA. Caxias do Sul-RS. ERAD 2009 - Fórum de Iniciação Científica, 2009.
- CSBC 2009. Controle da Adaptação na UbiComp. In: XXIX Congresso da SBC, 2009, Bento Gonçalves-RS. SBCUP - I Simpósio Brasileiro de Computação Ubíqua e Pervasiva - 2009.
- CIC-UCPEL 2009. Uma Proposta de Controle da Adaptação Dinâmica ao Contexto na Computação Ubíqua. Pelotas, RS. Mostra de Pós-Graduação, 2009. Prêmio Pesquisador em Pós-Graduação.
- ERAD 2010. EXEHDA-DA: Controle da Adaptação Dinâmica ao Contexto para o Middleware EXEHDA. Passo Fundo-RS. ERAD 2010 - Fórum de Pós-Graduação, 2010. Aceito para publicação.
- ERAD 2010. FWADAPT: Framework para Definição de Política de Adaptação Dinâmica de Aplicações na Computação Ubíqua. Passo Fundo-RS. ERAD 2010 - Fórum de Iniciação Científica, 2010. Aceito para publicação.

Trabalhos Futuros, destacam-se:

- possibilidade de ter regras de adaptação que modifiquem tanto aspectos funcionais como parâmetros de outras regras (adaptação ao contexto do próprio EXEHDA-DA);
- expandir funcionalidades do framework FWADAPT. Entre estas: a utilização da Classe Operacao na composição da regra lógica do adaptador; conversão automática da regra lógica em uma regra que possa ser usada diretamente pelo EXEHDA-DA, através da API Jena/SPARQL; mostrar a regra lógica num formato mais amigável para o usuário;
- aprofundar estudos das alternativas para especificação das operações que compõem a regra de adaptação;
- promover o uso do modelo de controle da adaptação dinâmica ao contexto em um ambiente com demandas potencias, como a Embrapa Clima Temperado;

Trabalhos Futuros (continuação ...)

- dar continuidade aos procedimentos de integração do EXEHDA-DA com o diferentes serviços e funcionalidades do *middleware* EXEHDA;
- revisar o atual modelo ontológico, documentando as suas classes, atributos e relacionamentos;
- revisar as taxonomias na área de aplicações ubíquas, sistematizando suas funcionalidades e propondo novos comandos adaptativos para atendimento das suas demandas;
- expandir o modelo de controle de adaptação dinâmica do EXEHDA-DA para que contemple tomada de decisão adaptativa considerando histórico de contexto e histórico de adaptações inferidas.

Outras Informações

WIKI

<http://olaria.ucpel.tche.br/nelsiw>

Uma Proposta de Controle da Adaptação Dinâmica ao Contexto na Computação Ubíqua

por

Nelsi Warken

Orientador

Prof. Dr. Adenauer Corrêa Yamin

Mestrado em Ciência da Computação

PPGINF - Centro Politécnico - Universidade Católica de Pelotas