


Laboratório de Engenharia de Software

Máquinas de estado

Arndt von Staa
Departamento de Informática
PUC-Rio
Março 2015

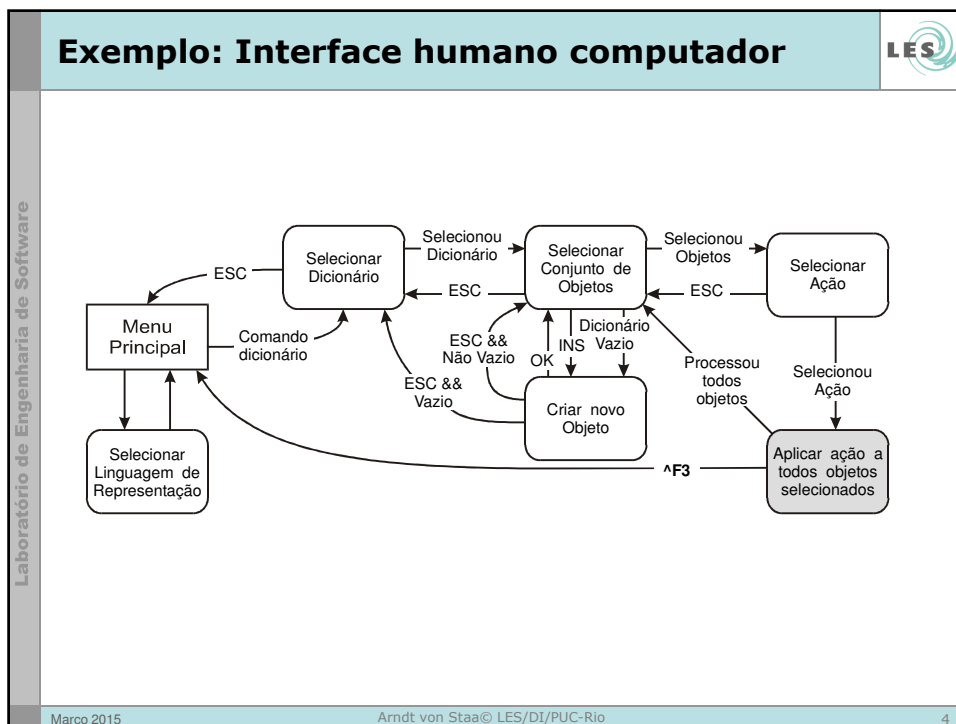
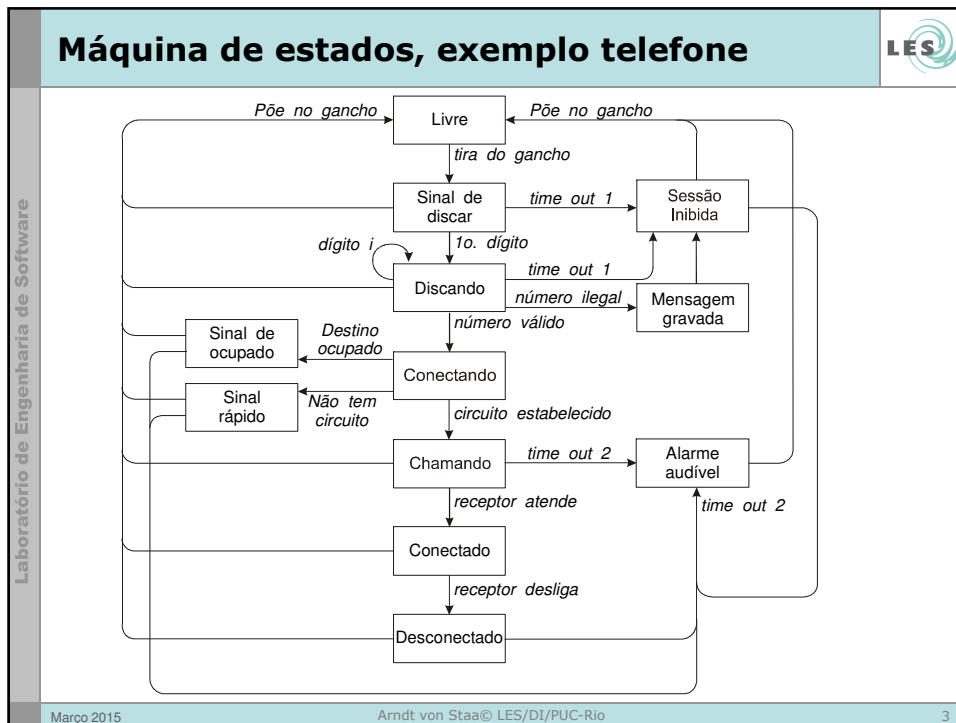
Especificação



Laboratório de Engenharia de Software

- Objetivo desta aula
 - Apresentar máquinas de estado e seu uso ao gerar testes funcionais
- Justificativa
 - Muitos testes dependem de uma sequência grande e complexa de decisões.
 - Determinar que dados devem ser fornecidos, em que ordem e segundo que condições é uma tarefa complexa e propensa a enganos.
 - Gerar e fornecer todos esses dados ao programa sob teste também tende a ser uma tarefa complexa e propensa a enganos.
 - Consequentemente, deseja-se estabelecer uma forma para gerar (quase) automaticamente os dados de teste
- Texto
 - Pezzè, M.; Young, M.; Teste e Análise de Software; Porto Alegre, RS: Bookman; 2008, capítulo 14

Março 2015
Arndt von Staa© LES/DI/PUC-Rio
2

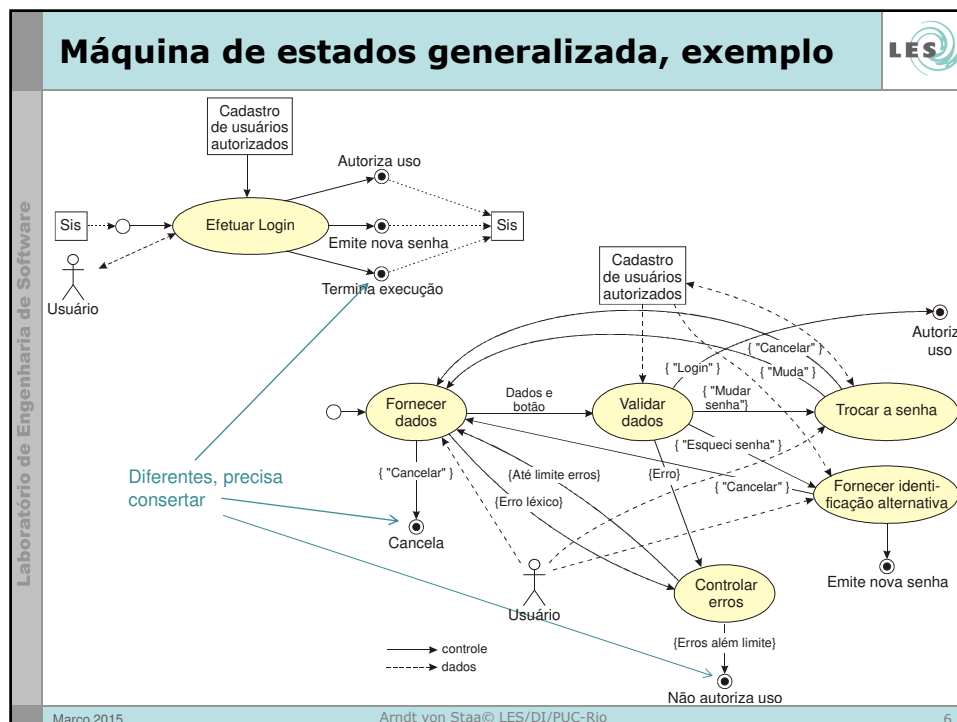


Casos de teste criados (incompleto)					
Ação	Entrada	Memória	Saída	Memória modificada	Risco de evolução
Selecionar linguagem	Menu princ: Linguagem	Tabela linguagens	Exibe linguagem selec	Linguagem corrente	Tabela linguagens pode evoluir
Ir dicionário	Menu princ: dicionário	Tabela dicionários	Exibe classes da linguagem		
Selecionar classe	Menu dic sel: classe	Tabela classes	Realça classe	Classe corrente	Tabela classes pode evoluir
Ir classe	Menu dic: aciona	Objetos classe corr	Exibe parte dos objetos		
Selecionar objeto	Menu classe marca: obj	Lista objetos selec	Objeto ticado	Obj adicionado lista	O conjunto de objetos evolui durante o uso
Ir objeto	Menu classe: aciona obj	Lista objetos selec	Menu ações	Obj acionado adicionado	
Selecionar ação	Acionar ação	Lista objetos	Janela ação	- Lista objetos - Ação a ser executada	Conjunto de ações depende da linguagem

Março 2015

Arndt von Staa© LES/DI/PUC-Rio

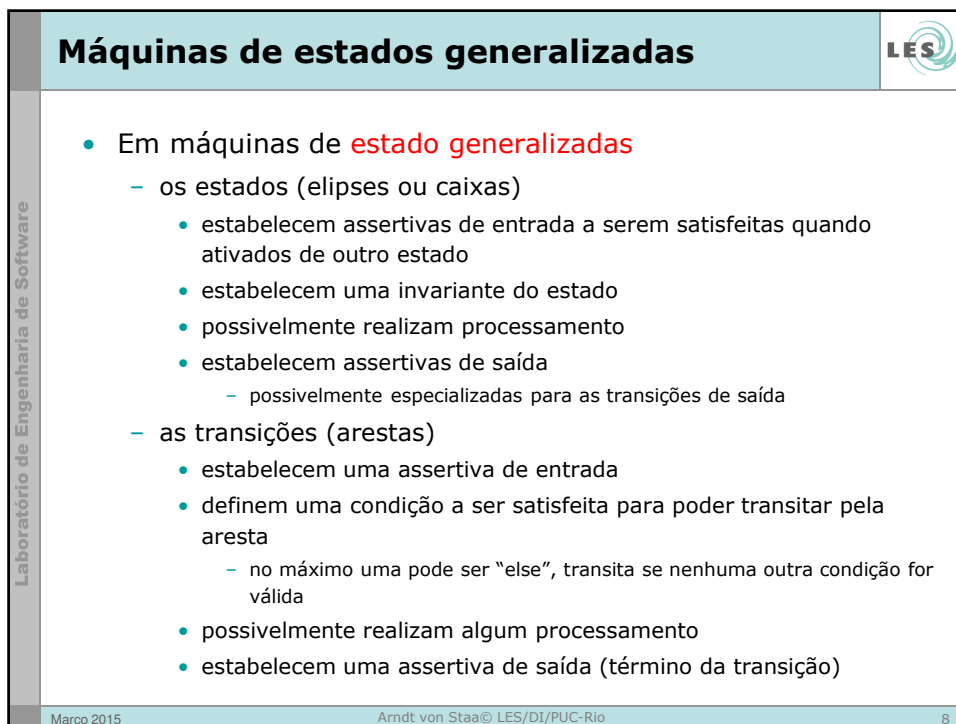
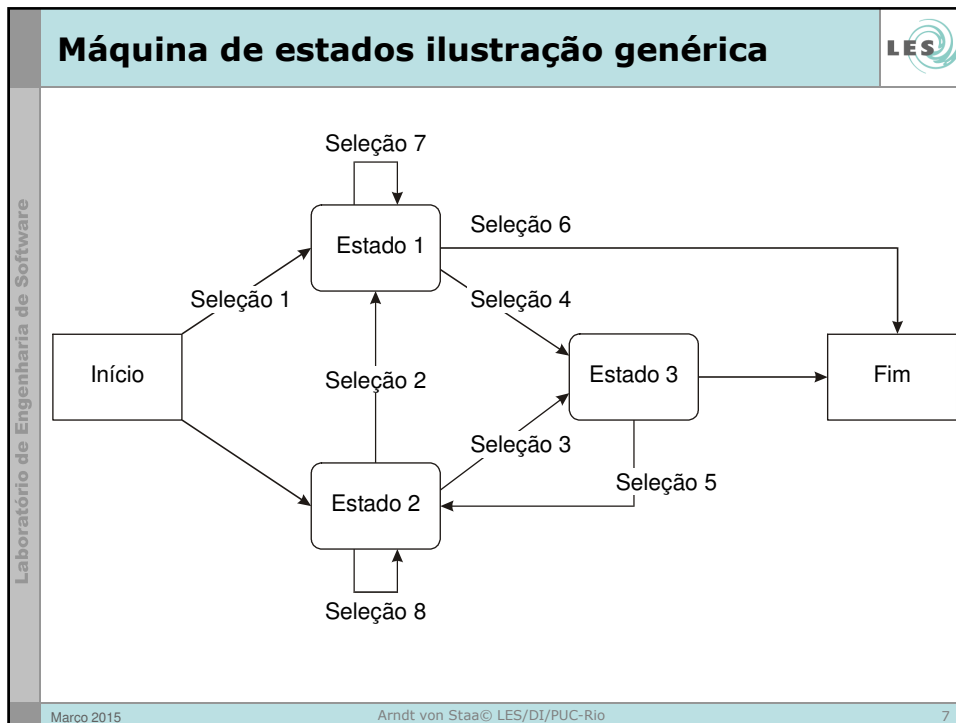
5



Março 2015

Arndt von Staa© LES/DI/PUC-Rio

6



Máquinas de estados generalizadas



- Em máquinas de estado generalizadas:
 - os estados podem conter **código executável**
 - as transições podem conter, além de condições, **ações** a serem efetuadas caso a máquina transite por aquela aresta
 - os fragmentos de código contidos nos estados, e/ou nas arestas podem **fazer uso de memória**
 - ex. pilha para análise de **linguagens livres de contexto**
 - ex. tabelas de símbolos para **filtros léxicos**
 - estados podem ser **decompostos** em máquinas de estado mais detalhadas.
 - nas máquinas resultado da decomposição
 - tudo que atinge o estado decomposto deve aparecer como origem
 - e tudo que sai do estado decomposto deve aparecer como término
 - pode-se criar uma **máquina nível zero** formada por um único estado e as origens e términos do processamento como um todo

Março 2015

Arndt von Staa© LES/DI/PUC-Rio

9

Máquina de estados



- Uma máquina de estados é um **grafo dirigido**
 - cada **vértice** é um estado
 - cada vértice pode conter uma **ação**
 - podem existir dois tipos de vértices especiais: **início** e **término**
 - nem sempre existem ⊗
 - cada **aresta** é uma transição
 - cada aresta possui no máximo um **rótulo condição** que designa a condição que permite seguir por aquela aresta
 - uma aresta sem rótulo condição corresponde a um **"else"**
 - no máximo uma aresta de saída de um estado pode estar sem rótulo
 - cada aresta pode possuir zero ou mais **rótulos de ação** que serão executadas caso o processamento siga pela aresta


Março 2015

Arndt von Staa© LES/DI/PUC-Rio

10

Laboratório de Engenharia de Software

Máquina de estados




- A partir do vértice inicial a execução prossegue de estado a estado, de acordo com a condição válida no momento
- Caso **nenhuma** das condições de saída do estado corrente valha
 - se for **estado de término** (final) o processamento termina
 - se não for, o **estado bloqueia** → **erro de projeto da máquina**
 - ocorre se a disjunção (**ou-tório** ☺) de todos os rótulos de condição de saída do estado não resultar em "true"
- Caso **mais de uma** das condições de saída do estado corrente valham
 - o **estado é ambíguo** → erro de projeto da máquina
 - ocorre se existirem um ou mais pares de rótulos condição de saída em que a conjunção das condições (**and**) não resulta em "false"

Março 2015
Arndt von Staa© LES/DI/PUC-Rio
11

Laboratório de Engenharia de Software

Máquinas de estado: aspectos positivos




- Permitem
 - visualizar e verificar se as transições estão completas e corretas
 - exibir condições compostas complexas
 - gerar código diretamente a partir do diagrama
 - máquinas de estado generalizadas são código procedural em nível de abstração mais alto
 - verificar as condições (assertivas) de entrada e saída
 - verificação de modelos
 - controle dinâmico da execução
- Grande parte das ações envolvendo máquinas de estado podem ser automatizadas
 - as anotações envolvendo assertivas e código podem ser criadas localmente sem necessitar conhecer o diagrama inteiro

Março 2015
Arndt von Staa© LES/DI/PUC-Rio
12

Laboratório de Engenharia de Software

Máquinas de estado: aspectos negativos



- Tendem a levar a um **conjunto muito grande de estados**
 - efeito *papel de parede*
 - dimensão grande pode tornar difícil entender o diagrama
- Pode-se atenuar isso através de uma hierarquia de máquinas
 - um estado pode ser decomposto (explodido) em uma nova máquina em um nível de abstração mais baixo
 - tudo que atingia (ou saía) o estado decomposto tem que atingir (ou sair de) algum estado da nova máquina


Março 2015

Arndt von Staa© LES/DI/PUC-Rio

13

Laboratório de Engenharia de Software

Casos de uso como máquina de estado



O **caso de uso** pode ser transformado em uma máquina de estados

- Interações com o usuário são tracejadas
- Pontos de início são disparados ao acionar
- Pontos de término informam o significado do término
- Arestas de controle são dirigidas e vão de estado a estado
- As arestas de controle são rotuladas
 - o rótulo informa a condição que faz seguir aquela aresta
 - condições que correspondem a ações do usuário, ex. teclou um botão, têm o seu texto redigido entre aspas
 - teclas de atalho devem ser identificadas por um nome entre aspas, e não pelo valor do atalho
- Inclusões e extensões seguem o padrão de caso de uso

Março 2015

Arndt von Staa© LES/DI/PUC-Rio

14

Exemplo: Caso de uso Efetuar Login	
Laboratório de Engenharia de Software	<div> <div>fluxo principal</div> <div> <p>A forma "... ativa ..." implica o uso de máquina de estados</p> </div> </div> <div> <ol style="list-style-type: none"> O componente limpa os campos e gera os caracteres de controle O usuário digita sua identificação, senha e caracteres de controle Quando o usuário selecionar a ação "Login" então <ol style="list-style-type: none"> O controle de acesso verifica sintaticamente os dados fornecidos O controle de acesso verifica se <usuario, senha> existe O controle de acesso retorna ao sistema sis, fornecendo a condição "autorizar uso" e os direitos de uso correspondentes a <usuário, senha> Quando o usuário selecionar a ação "Mudar senha" então <ol style="list-style-type: none"> O controle de acesso verifica sintaticamente os dados fornecidos O controle de acesso verifica se <usuario, senha> existe O controle de acesso ativa o caso de uso "Trocar a senha" Repete a partir de 1 Quando o usuário selecionar a ação "Esqueci a senha" então <ol style="list-style-type: none"> O controle de acesso verifica sintaticamente os dados fornecidos O controle de acesso verifica se usuário existe O controle de acesso ativa o caso de uso "Fornecer identificação alternativa" Se retornar do caso de 5.3, repete a partir de 1 Quando o usuário selecionar a ação "Cancelar" então <ol style="list-style-type: none"> O controle de acesso retorna ao sistema sis, fornecendo a condição "cancelar uso" e direitos de uso nulo </div>
	<div>Arndt von Staa©</div> <div>15</div> <div>Março 2015</div>

Componente Login: especificação	
Laboratório de Engenharia de Software	<div> <div>fluxos alternativos</div> </div> <div> <p>Evento 1/3.1, 4.1, 5.1 : O usuário digitou identificação, senha ou caracteres de controle incorretos</p> <p>E1.1 Se for a quarta ou mais vez que ocorreu um evento de erro então</p> <p>E1.1.1. O controle de acesso emite a mensagem "Acesso não autorizado"</p> <p>E1.1.2. O controle de acesso retorna ao sistema sis, fornecendo a condição "não autorizar uso" e direitos de uso nulo</p> <p>Fim se</p> <p>E1.2. Se caracteres de controle incorretos: o controle de acesso emite a mensagem "Caracteres de controle incorretos"</p> <p>E1.3. Se identificação do usuário ou senha incorretos: o controle de acesso emite a mensagem "Dados incorretos"</p> <p>E1.4. O controle de acesso retorna ao passo 1</p> <p>Fim evento E1</p> <p>Evento 2/3.2, 4.2 : O par <usuário, senha> não está definido</p> <p>E2.1 Se for a quarta ou mais vez que ocorreu um evento de erro então</p> <p>E2.1.1. O controle de acesso emite a mensagem "Acesso não autorizado"</p> <p>E2.1.2. O controle de acesso retorna ao sistema sis, fornecendo a condição "não autorizar uso" e direitos de uso nulo</p> <p>Fim se</p> <p>E2.2. O controle de acesso emite a mensagem "Usuário desconhecido"</p> <p>E2.3. O controle de acesso retorna ao passo 1</p> <p>Fim evento E2</p> </div>
	<div>Arndt von Staa©</div> <div>16</div> <div>Março 2015</div>

Laboratório de Engenharia de Software

Componente Login: especificação

fluxos alternativos

Evento 3/5.2 : A identificação do usuário não existe no cadastro

E3.1 Se for a quarta ou mais vez que ocorreu um evento de erro então

E3.1.1. O controle de acesso emite a mensagem "Acesso não autorizado"

E3.1.2. O controle de acesso retorna ao sistema **sis**, fornecendo a condição "não autorizar uso" e direitos de uso nulo

Fim se

E3.2. O controle de acesso emite a mensagem "Usuário desconhecido"

E3.3. O controle de acesso retorna ao passo 1

Fim evento E3

Evento E4: o usuário clica "Cancelar" em qualquer lugar

E4.1 O sistema solicita confirmação do cancelamento

E4.2 Se usuário confirma o cancelamento

E4.2.1 O controle de acesso retorna fornecendo o conjunto "cancelar uso" ao sistema **sis**

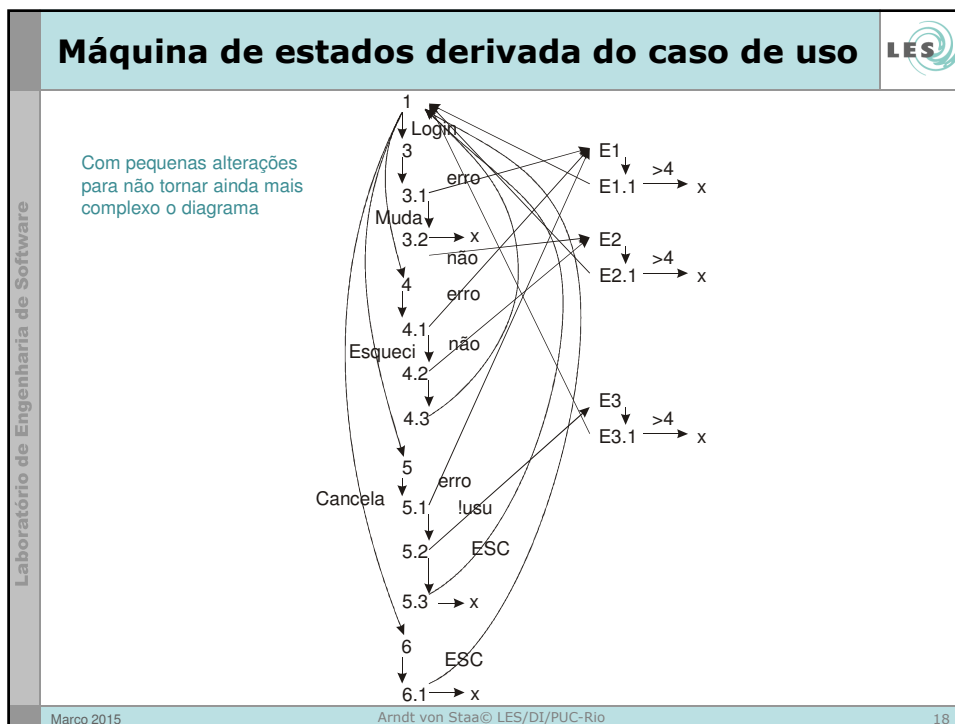
Fim se

E4.3 O controle de acesso retorna ao passo 1

Fim evento E4.

Veja *Aula 04 Especificações, resumo*

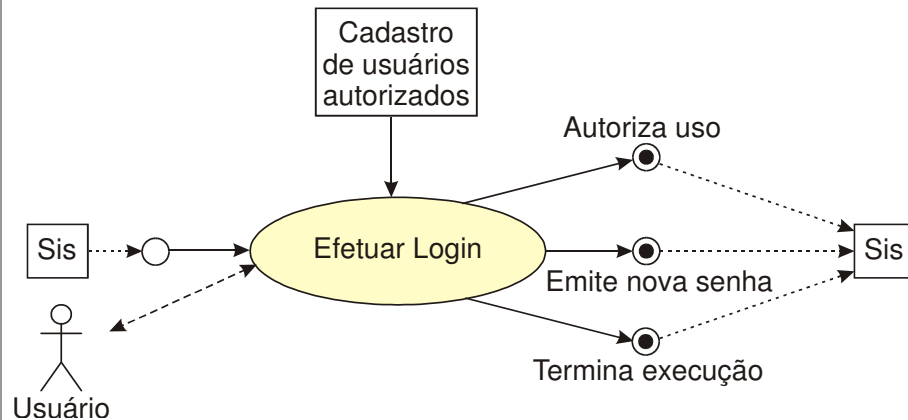
Arndt von Staa©
17
Março 2015



Máquina de estados criada diretamente



- Contexto do caso de uso → diagrama nível 0



Março 2015

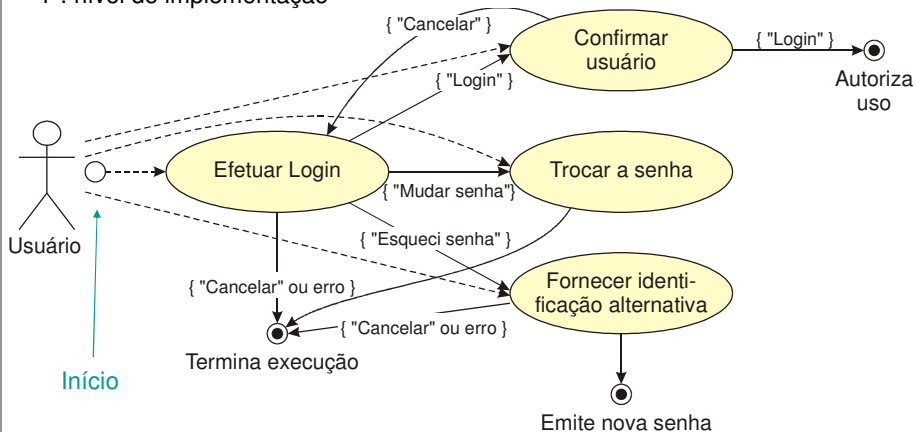
Arndt von Staa© LES/DI/PUC-Rio

19

Componente Login, máquina de estados



1º. nível de implementação



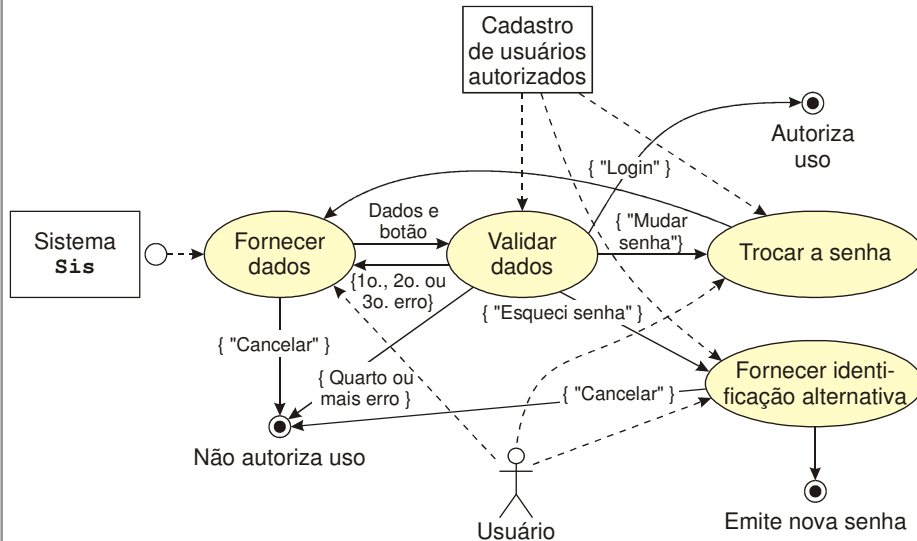
- A máquina pode ser melhorada?
- Considere a separação da entrada de dados da verificação dos dados
 - verificação léxica pode ser realizada imediatamente
 - verificação envolvendo bases de dados, métodos etc. devem estar separados da entrada

Março 2015

Arndt von Staa© LES/DI/PUC-Rio

20

Componente Login, máquina de estados



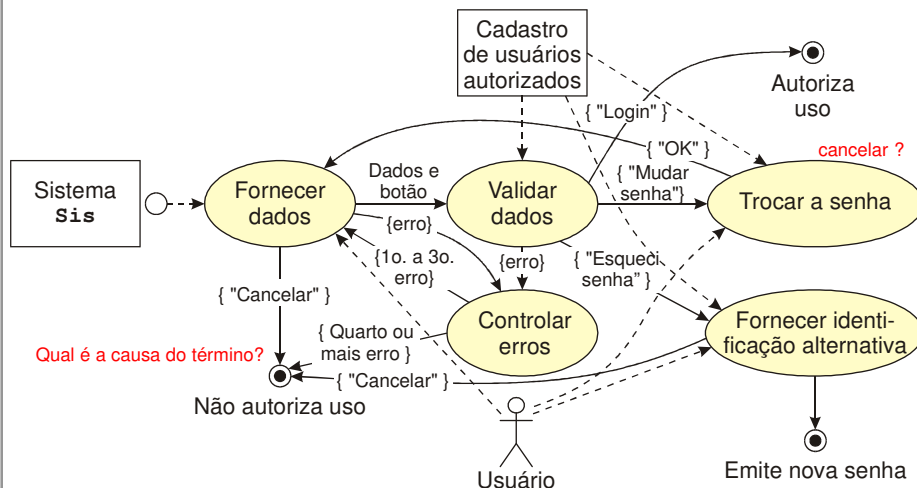
- A máquina pode ser melhorada mais ainda?
- Considere fatorar o controle de número de erros.

Março 2015

Arndt von Staa© LES/DI/PUC-Rio

21

Componente Login, máquina de estados



- Ainda tem problemas a serem resolvidos?
- Considere a separação de "cancelar" e "não autorizar", cancelar troca de senha, ...
 - cada saída de um estado ou da máquina deve explicitar a causa da escolha

Março 2015

Arndt von Staa© LES/DI/PUC-Rio

22

LES



Arndt von Staa© LES/DI/PUC-Rio

23


LES



Arndt von Staa© LES/DI/PUC-Rio

24

Exemplo: Fornecer dados 1 / 4




Laboratório de Engenharia de Software

Estado	Fornecer dados
Resumo	Recebe os dados do usuário e efetua a verificação independente de cadastro (<i>verificação léxica</i>)
Escopo	não se aplica a estados
Ator principal	não se aplica a estados
Interessados	não se aplica a estados
Invariante	O número de erros identificados na presente instância de uso é menor ou igual a três.
Pré condição	
Acionamento	<i>Obter dados</i> inicia quando <ul style="list-style-type: none"> ou o sistema sic solicitar dados do usuário para autorizar o uso ou o estado validar dados encontra dados ilegais e solicitar novos dados

Março 2015
Arndt von Staa© LES/DI/PUC-Rio
25


Exemplo: Fornecer dados 2 / 4



Laboratório de Engenharia de Software

Fluxo principal	<ol style="list-style-type: none"> 1. O controle de acesso limpa os campos de entrada 2. O controle acesso gera os caracteres de controle 3. O usuário digita sua identificação, senha e caracteres de controle 4. O usuário clica a ação a ser realizada 5. O controle de acesso valida a corretude léxica de idUsuario 6. O controle de acesso verifica os caracteres de controle digitados 7. O controle de acesso ativa o caso de uso <i>Validar Dados</i>
------------------------	--

Março 2015
Arndt von Staa© LES/DI/PUC-Rio
26

Exemplo: Fornecer dados 3 / 4


Laboratório de Engenharia de Software

Fluxos alternativos	<p>Evento E1: o usuário clica "cancelar"</p> <p style="margin-left: 20px;">E1.1 O controle de acesso fecha a janela de identificação de usuário</p> <p style="margin-left: 20px;">E1.2 O controle de acesso fornece o controle "cancelar uso" ao sistema sis</p> <p>Fim evento E1.</p> <p>Evento E2/6: o usuário digita caracteres de controle incorretos</p> <p style="margin-left: 20px;">E2.1 O controle de acesso emite a mensagem "Caracteres de controle incorretos"</p> <p style="margin-left: 20px;">E2.2 Ativa o estado "controlar erros"</p> <p>Fim evento E2.</p> <p>Evento E3/5: o usuário fornece identificação usuário lexicamente incorreta</p> <p style="margin-left: 20px;">E3.1 O controle de acesso emite a mensagem "Usuário incorreto"</p> <p style="margin-left: 20px;">E3.2 Ativa o estado "controlar erros"</p> <p>Fim evento E3.</p>
---------------------	--


E1. Melhor:

E1.1 O controle acesso prepara o retorno <Cancelar, direitos: vazio>

E1.2 Termina

E3 - Especificação léxica de usuário ver: regra de negócio idUsuario

Março 2015
Arndt von Staa© LES/DI/PUC-Rio
27

Exemplo: Fornecer dados 4 / 4


Laboratório de Engenharia de Software

Pós condições	Dados e ação a executar fornecidos ao caso de uso <i>Validar dados</i>
Garantia mínima	<ul style="list-style-type: none"> N/A
Requisitos	<ul style="list-style-type: none"> -,-
Regras de negócio	<ul style="list-style-type: none"> idUsuario deve ter entre 5 e 30 caracteres idUsuario não deve conter letras diacríticas idUsuario não deve conter dígitos idUsuario não deve conter brancos idUsuario pode conter somente os caracteres especiais: '-' e '.'
Casos de uso correlatos	Validar dados Reconfirmar usuário corrente Trocar a senha Fornecer identificação alternativa

Março 2015
Arndt von Staa© LES/DI/PUC-Rio
28

Laboratório de Engenharia de Software

Conversão "Fornecer dados" para tabela de decisão




		1	2	3	4	5	6	7	8	9	10
	Usuário correto	-	s	s	s	n	n	n	-	-	-
	Caracteres corretos	-	s	s	s	s	s	s	n	n	n
xorob	Tecla "Login"	n	s	n	n	s	n	n	s	n	n
xorob	"Mudar senha"	n	-	s	n	-	s	n	-	s	n
xorob	"Esqueci senha"	n	-	-	s	-	-	s	-	-	s
xorob	"Cancelar"	s	n	n	n	n	n	n	n	n	n
	Ativar "validar"		x	x	x						
	Ativar "erro"					x	x	x	x	x	x
	Erro "caracteres controle"								x	x	x
	Erro "léxico"					x	x	x			
	Termina "cancela"	x									

xorob – exclusive or obrigatório → exatamente uma das condições deve ser true

Março 2015
Arndt von Staa© LES/DI/PUC-Rio
29

Laboratório de Engenharia de Software

Exemplo: Validar dados 1 / 3




Estado	Validar dados e ação
Resumo	Valida os dados e a ação solicitada com relação ao cadastro de usuários autorizados
Invariante	
Pré condição	Nome correto, senha e ação selecionada é uma de { login, mudaSenha, novaSenha } resultado do estado fornecer dados
Acionamento	<i>Validar dados</i> inicia ao receber o controle de <i>Obter dados</i>
Fluxo principal	<ol style="list-style-type: none"> 1. O Controle de acesso busca os dados do idUsuario no cadastro 2. Se a ação solicitada for "Esqueci senha" <ul style="list-style-type: none"> Então 2.1 Ativa o estado <i>Fornecer identificação alternativa</i> FimSe 3. O Controle de acesso verifica se a senha fornecida corresponde a uma das registradas para este usuário 4. Se a ação solicitada for "Login" <ul style="list-style-type: none"> Então 4.1.1 O Controle de acesso prepara o retorno <autorizado, direitos: de < idUsuario, senha>> 4.1.2 Termina Senão 4.2 O Controle de acesso ativa o estado <i>Trocar senha</i> FimSe

Não deveria ser : se ação solicitada é "trocar senha" ?
→

Março 2015
Arndt von Staa© LES/DI/PUC-Rio
30

Exemplo: Validar dados 2 / 3



Laboratório de Engenharia de Software

Fluxos alternativos	<p>E1. Evento: Identificação do usuário não existe no Cadastro</p> <p style="padding-left: 20px;">E1.1. O Controle de acesso exibe a mensagem "Usuário desconhecido"</p> <p style="padding-left: 20px;">E1.2. O Controle de acesso ativa o estado "Controlar erros"</p> <p>Fim evento</p> <p>E2. Evento: Senha não fornecida, ou lexicamente errada ou não corresponde a qualquer uma das senhas de idUsuario</p> <p style="padding-left: 20px;">E2.1. O Controle de acesso exibe a mensagem "Usuário desconhecido"</p> <p style="padding-left: 20px;">E2.2. O Controle de acesso ativa o estado "Controlar erros"</p> <p>Fim evento</p>
---------------------	---


Razões de segurança indicam que as duas mensagens devem ser alguma coisa similar a "Usuário desconhecido". Um possível agressor ficará na dúvida se o problema é do idUsuario ou da Senha.

Problema 1: como discernir o erro de dados observado durante os testes?

Problema 2: a condição do evento E2 contém uma expressão lógica composta. Deveria ser simples para que se possa criar tabelas de decisão com condições binárias.

Março 2015
Arndt von Staa© LES/DI/PUC-Rio
31

Exemplo: Validar dados 3 / 3



Laboratório de Engenharia de Software

Pós condições	
Garantia mínima	
Requisitos	<ul style="list-style-type: none"> Não deve ser possível discernir se o erro foi idUsuario incorreto ou se senha foi incorreta
Regras de negócio	
Casos de uso correlatos	<p>Fornecer dados</p> <p>Reconfirmar usuário corrente</p> <p>Trocar a senha</p> <p>Fornecer identificação alternativa</p>

Março 2015
Arndt von Staa© LES/DI/PUC-Rio
32

Laboratório de Engenharia de Software

Conversão "Validar dados" para tabela de decisão



		1	2	3	4	5	6	7	8	9
	Usuário conhecido	s	s	s	s	s	n	n	n	-
	Senha corresponde	s	s	-	n	n	-	-	-	-
xorob	Tecla "Login"	s	n	n	s	n	s	n	n	n
xorob	"Mudar senha"	-	s	n	-	s	-	s	n	n
xorob	"Esqueci senha"	-	-	s	-	-	-	-	s	n
	Autorizar	x								
	Ativar "Controlar"				x	x	x	x	x	
	Ativar "Trocar"		x							
	Ativar "Esqueci"			x						
	Impossível									x
	Msg: "Usuário desconhecido"				x	x	x	x	x	


Março 2015

Arndt von Staa© LES/DI/PUC-Rio

33

Laboratório de Engenharia de Software

Nível 0 – Final Autoriza uso



Estado	Final: Autoriza uso
Assertivas de saída	1. retornou < Autorizado, direitos de acesso de <idUsuario, Senha>>
Fluxo	1. Controle de acesso oblitera os registros decriptados do cadastro 2. Controle de acesso fecha a janela 3. Controle de acesso retorna < Autorizado, direitos: <idUsuario, Senha>>

O valor retornado pode ser um objeto da classe "Autorizacao". Exemplos de coisas que essa classe pode definir :

- atributo idUnicaUsuario – um inteiro identificador interno gerado ao cadastrar o par <idUsuario, senha>
- atributo condicao uma enumeração tpCondicao :: {AUTORIZA, CANCELA, NOVA_SENHA, NÃO_AUTORIZA}
- atributo lista de idDireito's – criptografada
- bool TemDireitos(char idDireito) :: retorna true sse a condição é AUTORIZA e a lista de idDireitos contém idDireito
- tpCondicao RevalidaUsuario() :: abre uma janela similar a login e que contém somente os botões Login e Cancelar. Retorna AUTORIZA sse os dados fornecidos correspondem à idUnicaUsuario

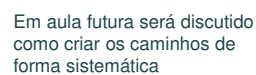
Março 2015

Arndt von Staa© LES/DI/PUC-Rio

34




LES



Fornecer dados; Validar dados; Autoriza
Fornecer dados; Validar dados; Controlar Erros; Fornecer dados; Validar dados; Trocar
senha; Muda; Fornecer dados; Validar dados; Autoriza
...

Laboratório de Engenharia de Software

Máquina de estados, casos teste semânticos




- Dado o caminho, um caso de teste abstrato
 - Fornecer dados; Validar dados; Autoriza
- Converter para caso de teste semântico
 - precisa analisar o caminho,
 - usualmente faz-se de trás para diante examinando
 - a condição associada à aresta
 - as assertivas de saída (origem) e entrada (destino) dos estados
 - processamento no estado origem da aresta
- Resultado
 - Autoriza: par <Usuário , senha> existe, direitos par definidos
 - Validar dados -> Autoriza → usuário existe, par <usuário, senha> existe, botão = Login
 - Fornecer dados -> Validar dados → usuário correto, senha digitada, caracteres controle corretos, botão = Login

Março 2015
Arndt von Staa© LES/DI/PUC-Rio
37

Laboratório de Engenharia de Software

Qual é a ideia?




- Criar uma máquina de estados em que cada estado
 - ou é "simples"
 - ou é decomposto em outra máquina de estados
- Para cada estado
 - criar um caso de uso do estado
 - fluxo principal
 - zero ou mais fluxos alternativos
 - o término de um fluxo é uma transição de estado ou o término da máquina raiz
- O término de um fluxo corresponde
 - ou a uma mudança de estado
 - ou ao término da máquina raiz
 - ou ao término da máquina de decomposição
 - neste caso a transição (oráculo da tabela de decisão) é uma das transições de saída do estado "pai"

Março 2015
Arndt von Staa© LES/DI/PUC-Rio
38

Laboratório de Engenharia de Software

Qual é a ideia?

- Para cada estado cria-se uma tabela de decisão
 - a seleção de valores deve obedecer aos critérios de valoração
 - espera-se que o número de condições por estado seja pequeno, de modo que se controle o número de colunas das tabelas de decisão
- Cria-se a lista completa de caminhos da máquina de estados
 - repetições são resolvidas com base no arrasto e no limite de iterações caso exista
 - cada caminho na máquina de estados corresponde a um cenário
 - espera-se que cada máquina contenha poucos estados, de modo que o número de cenários seja pequeno
 - cada caminho determina as condições saída da tabela de decisão de cada um dos estados
- O teste cobre a combinação de condições em cada estado, mas não realiza a combinação entre estados



Março 2015
Arndt von Staa© LES/DI/PUC-Rio
39

Laboratório de Engenharia de Software

Referências bibliográficas

- Holcombe, M.; Bogdanov, K.; Gheorghe, M.; "[Functional Test Generation for Extreme Programming](#)"; *Proceedings of the XP2001 Second International Conference on Extreme Programming and Flexible Processes in Software Engineering*; 2001; pags 109-113 Buscado em: 19/08/2004; URL: <http://www.dcs.shef.ac.uk/~wmlh/XPtest.pdf>



Março 2015
Arndt von Staa© LES/DI/PUC-Rio
40

Laboratório de Engenharia de Software

LES

FIM

Março 2015Arndt von Staa© LES/DI/PUC-Rio41