

D.R.A.W.

Desenvolvimento Rápido de Aplicações Web

Projecto Integrado de Engenharia de Linguagens

2008/2009

Conteúdo

1	Introdução	1
2	DRAW	3
2.1	Modelo de Dados	4
2.2	Controlador: Autómato Reactivo	4
2.3	Vista	4

1 Introdução

As aplicações (sejam elas Web ou Locais) são desenvolvidas para o manuseamento de dados. Estes dados são inseridos, pesquisados, removidos, processados ou exportados. Cada uma destas operações está disponível ao utilizador mediante uma *interface*. Quando a *interface* recebe um estímulo (por exemplo, a escolha de uma opção num menu) é despoletada uma acção, e é apresentado ao utilizador o resultado da operação. A partir deste momento o utilizador pode, de novo, escolher uma operação a executar.

Este processo simples pode ser representado como um grafo ou autómato reactivo, conforme a figura 1;

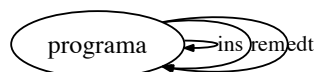


Figura 1: Autómato reactivo simplificado de uma aplicação.

Na verdade, podemos explicitar este fluxo de interacção com maior detalhe. Quanto realizamos a operação de inserção de dados, as aplicações pedem-nos que preenchamos um

formulário. Do mesmo modo, para remover ou editar um registo temos de indicar qual o registo que vai sofrer as alterações. No caso de remover vamos pedir ao utilizador que confirme esta operação, enquanto que para a edição vamos solicitar que nos dê a nova informação a colocar na base de dados. A figura 2 mostra o grafo anterior com maior detalhe.

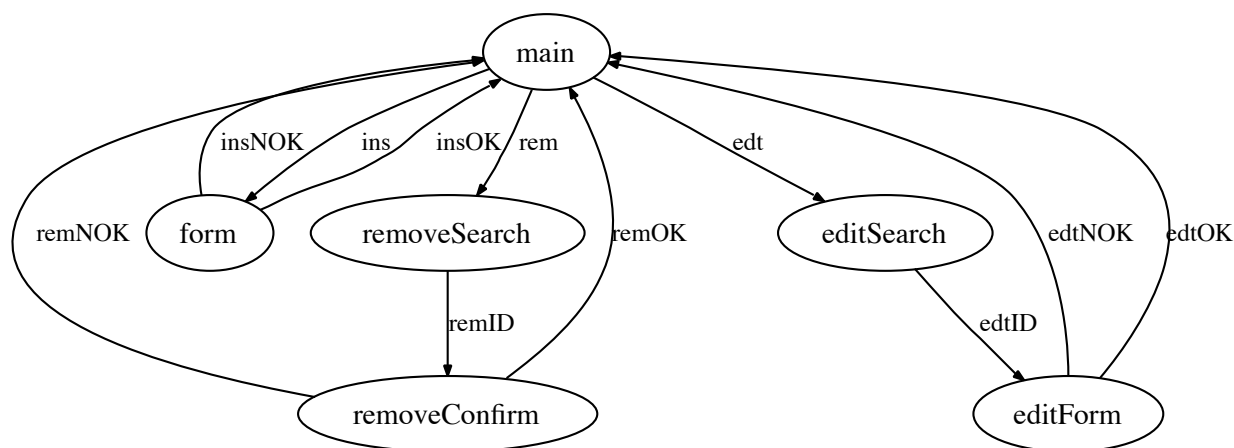


Figura 2: Autómato reactivo de uma aplicação.

O módulo que gere o fluxo de interacção com o utilizador e invoca as acções associadas a cada um desses estímulos é habitualmente designado de *Controlador*.

Cada um destes estímulos leva a que exista algum tipo de processamento. Por exemplo, que seja adicionado um registo na base de dados. Ao módulo que trata da chamada *lógica do negócio* e da *gestão da base de dados* é chamado *Modelo*.

Se adicionarmos ao grafo anterior nodos que representem as acções tomadas deixamos de ter ligações ambíguas: cada uma passa a ter uma acção muito própria (de acordo com a figura 3).

Sempre que o controlador recebe um estímulo do utilizador e realiza determinada acção no modelo é necessário apresentar o resultado desse processamento ao utilizador. Este resultado não é necessariamente composto de dados. Pode ser qualquer coisa desde um formulário para ser preenchido até uma tabela com uma listagem de registos. O módulo responsável para a representação dos dados resultantes da operação do Modelo é conhecido por *Vista*.

Estes três módulos (Modelo, Vista e Controlador) fazem parte de uma filosofia de desenvolvimento de aplicações (essencialmente Webs), conhecida por MVC (iniciais de Modelo, Vista, Controlador).

Cada linguagem de programação tem um ou mais *frameworks MVC* para o desenvolvimento de aplicações Web. Por exemplo, o *Catalyst* para Perl, o *Symfony* para PHP, ou

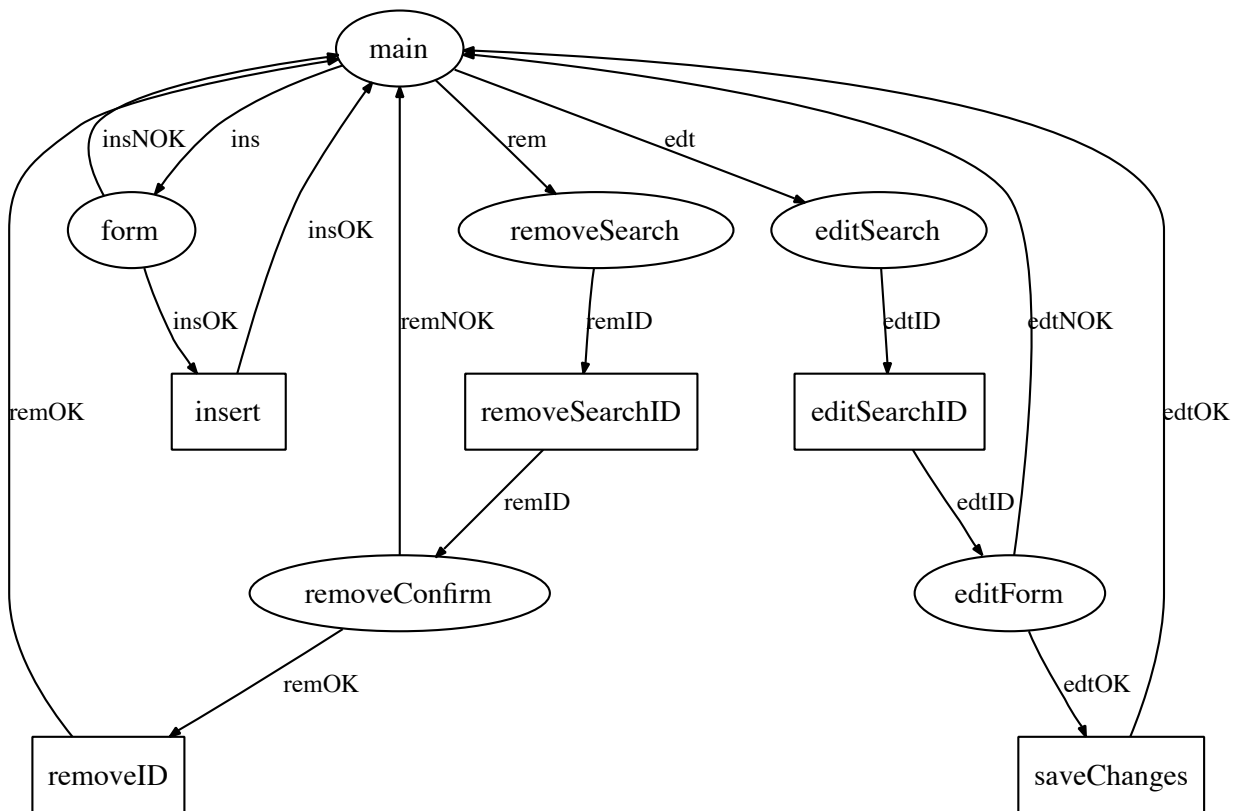


Figura 3: Autômato reativo de uma aplicação com nodos de processamento.

o *Plone* ou *django* para Python.

2 DRAW

O DRAW é (assim que o implementar) uma *framework MVC* básica e completamente *opinioned*, ou seja, uma ferramenta que tenta privilegiar a automatização no desenvolvimento da aplicação em vez da flexibilidade.

Uma aplicação formalizada em DRAW é composta por quatro partes principais:

1. a descrição formal do modelo de dados (que será analisada para dar origem a um modelo de dados convencional, como seja uma base de dados relacional);
2. um grafo (autômato) que especifica os vários estados da aplicação, quais os estímulos que cada estado pode receber, e qual (ou quais) as funções que devem ser invocadas durante a transição de estado;

3. para cada estado, a definição opcional de uma vista: como os resultados (ou formulários) devem ser apresentados ao utilizador;
4. um conjunto de funções de processamento, como sejam todas as funções referentes à *lógica de negócio*, bem como um conjunto de funções auxiliares;

Cada um destes quatro tipos de informação será especificado numa linguagem de domínio específico (*Domain Specific Language*). Enquanto que para as três primeiras deverá desenhar uma linguagem própria (diferente para cada uma, ou que partilhem partes), para a quarta sugere-se o uso da linguagem de programação base do sistema.

2.1 Modelo de Dados

2.2 Controlador: Autómato Reactivo

2.3 Vista