DiF v2.x

Digitalis Framework

Apresentação Comercial

Departamento de Desenvolvimento

Digitalis Informática ©2008

Pedro Viegas (pviegas@digitalis.pt)

Agenda

Estrutura da apresentação e tópicos a abordar

- O que é a DiF?
- Porquê a DiF?
- O que há de novo na v2.0?
- Arquitectura e seus benefícios
- Controlador da DiF
- O que faz a DiF?
- Built on top of Best-of-breed
- Tools
- Resources
- Quem está a usar a DiF?
- O que vem a seguir?
- Resumo

O que é a DiF?

Características que descrevem a DiF

- Open-source J2EE framework
- Conponent-driven development
- Arquitectura modular, potencia integrações/customizações
- Framework adaptável
- Sem ficheiros de configuração, 100% java code!
- Multi-canal, não apenas Web
- RAD – Rapid Application Development
- Focus na performance
- Web 2.0 UI
- Focus na Qualidade (QAG)
- Lema: Convenção vs Configuração vs Programação
- Plataforma de desenvolvimento criada e utilizada pela Digitalis Informática Ida

Porquê a DiF?

Reinventar a roda?

- Saberemos mais que estes Gurus e outros?
  - Craig McClanahan (Struts)
  - Rod Johnson (Spring)
  - Howard Lewis Ship (Tapestry)
  - Ed Burns, Craig McClanahan “and a whole bunch of brain power” (JSF – JSR-127)
- Qual a visão que a DiF introduz que não encontramos noutras soluções?
  - O Java não tem que ser complexo
  - Não temos que tirar um curso de mecânica para pilotar um automóvel
  - A produtividade em Java DEVE ser tão alta como qualquer outra RAD
  - Repetições, é mau...
    - Se é comum, a framework deve estar preparada para me ajudar
  - Não devo precisar usar uma framework por projecto, se tiver uma que se adapte a vários tipos de projecto, com uma reconfiguração/escolha de módulos/funcionalidades
  - Se dá muito trabalho a fazer...
    - Falta alguma coisa na framework!
  - Não quero que seja assim...
    - Troca a peça respectiva da framework por uma outra...
    - Ou cria a tua!

Porquê a DiF?

Razões para a escolha da DiF para um GESTOR

- Open-source framework
- Standards de mercado
- Integra as melhores soluções tecnológicas existentes para cada necessidade
- Funciona com vários servidores aplicacionais, RDBMS e outros
- Potencia a performance
- Reduz os tempos de desenvolvimento
- Reduz a necessidade de conhecimento tecnológico dos developers
- Plataforma modular, muito configurável e extensível
- Feita de raiz com as necessidades de integração em mente
- Out-of-the-box
  - Integração com várias tecnologias como LDAP, RDBMS, Identity/Authorization managment, etc.
  - Interfaces de administração
  - Pacote de componentes que possibilitam normalizar os produtos desenvolvidos
  - Usabilidade/Acessibilidade
  - Internacionalização
  - Etc...
- Software-house criadora do projecto disponibiliza serviços de apoio ao desenvolvimento

Porquê a DiF?

Razões para a escolha da DiF para um DEVELOPER

- J2EE framework
- Rica em features
  - Inúmeros Componentes de UI Web 2.0
  - Gestão de sessão, users, grupos, acessos, excepções, logging, etc.
  - Tratamento de parâmetros de chamada
  - Configurações
- Adapta-se ao código do Developer
- Abundante geração de código
- Integração com inúmeras ferramentas e APIs out-of-the-box
- Potencia o desenvolvimento orientado por componentes
- Potencia a performance
- IoC centric, totalmente modular e plugable
- Multi-canal, abstrai o developer das especificidades dos mesmos
- Ambiente para desenvolvimento (light and simple!)
- 100% java code (+Annotations). Não necessita de XML, BD, ou outra forma de defenir meta-data

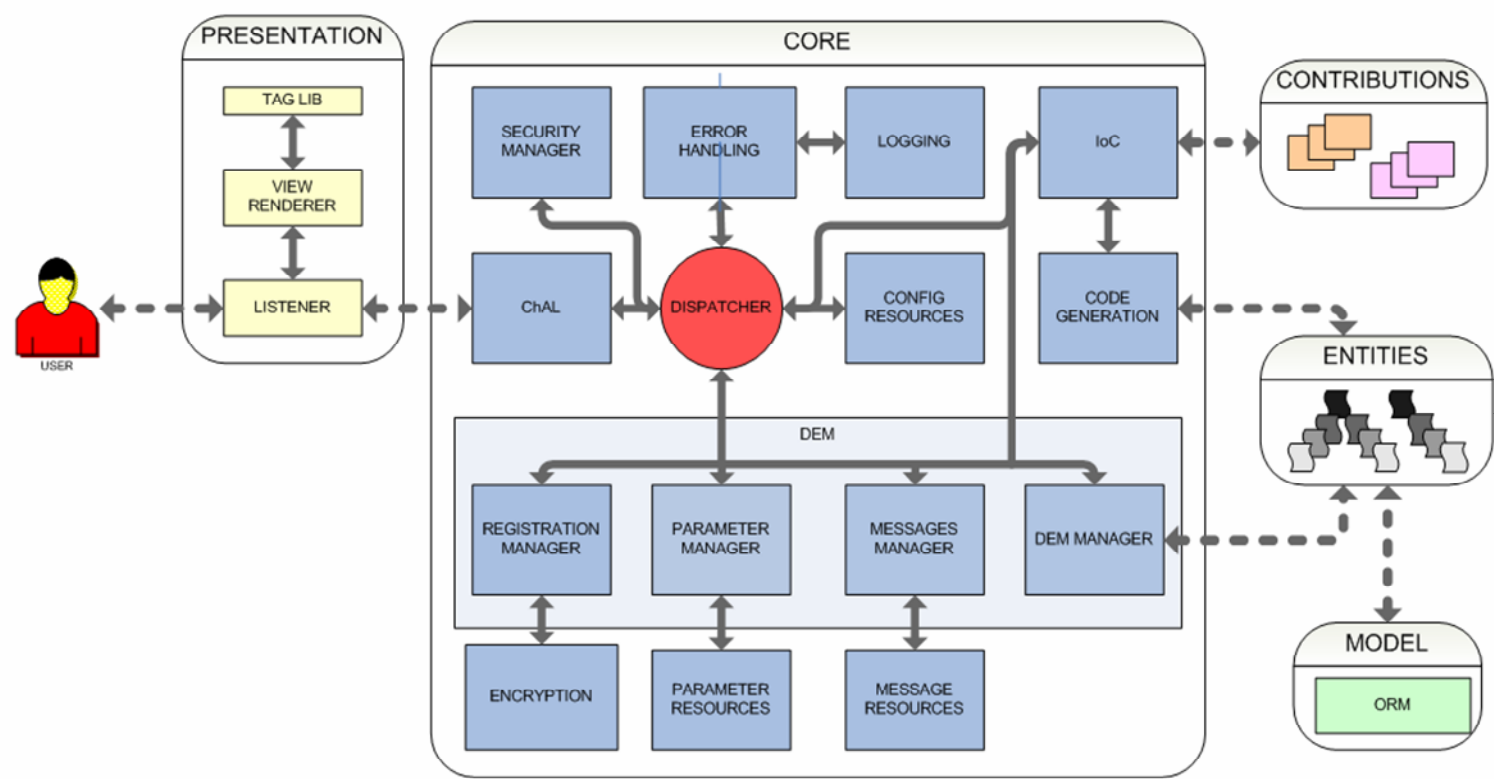
O que há de novo na DiF v2.x

Novidades relativas à anterior framework

- Multi-canal
- O fim dos metadados externos. Annotation-centric Model
- Component-driven development
- Completamente modular, mesmo dentro do seu Core
- IoC como peça base para integração de todos os módulos
- Geração de código, para inúmeras necessidades
- Web UI 2.0 (inúmeros componentes, acessibilidade, usabilidade)
- Integração com Hibernate para ORM, com geração de código (+/- 80%)
- Automação da gestão do projecto: Maven
- Suporte a vários application servers
- Mais...
  - Produtividade
  - Performance
  - Simplicidade
  - Flexibilidade

Arquitectura da DiF2

Descrição da arquitectura da nova plataforma



Arquitectura da DiF2

Descrição da arquitectura da nova plataforma

- JDK Java 5.0
  - Especial destaque para o uso extensivo de annotations
- Arquitectura MVC (multi-canal)
- Servlet 2.1 para Channel HTTP (Aplicações Web)
- Guice IoC container
  - Para integrações com módulos externos, e, para integração de todas as peças internas da framework
  - Arquitectura completamwente modular
- Testada com:
  - JBoss
  - Jetty
  - Tomcat

Arquitectura da DiF2

Model View Controller – Implementações comuns

- Model View Controller em Web frameworks
  - O controler é uma servlet. Preso à Servlet API
  - O fluxo e interpretação dos pedidos é muitas vezes indissociável
  - Não possibilita aproveitar o Model de forma fácil para outros canais, muito menos as Views
- MVC + Front Controler
  - Separa o flow da interpretação do pedido e sua resposta. Mantem no entanto as restantes características
- Várias implementações do MVC Multi-canal
  - Duplicam parte da lógica de controller em várias implementações
  - Não abstraem totalmente o Model do canal, possibilitando e até forçando o uso de APIs específicas do canal como a Servlet API, SOAP implementation, ou outra.

Arquitectura da DiF2

MVC da DiF2, um MVC Multi-Channel que simplifica o desenvolvimento

- Model
  - Simples POJOs anotados. Total liberdade, sem uso de nenhuma API de canal
- View
  - Múltiplas possibililidades
  - Actualmente JSPViewRenderer para HTTP, mas outras surgirão como PDF, Freemarker, etc.
  - Add your own!
- Controller
  - Listener
  - ChAL
  - Dispatcher

Arquitectura da DiF2

ChAL & Dispatcher

- Listener
  - Os listeners são quem recebe os pedidos e envia ao ChAL. Estes são “channel dependent” naturalmente.

São as portas de entrada. Pedidos Web, ligações a gateways SMS, clientes de email, etc.

Arquitectura da DiF2

ChAL & Dispatcher

- Listener
  - Os listeners são quem recebe os pedidos e envia ao ChAL. Estes são “channel dependent” naturalmente.
- ChAL
  - Converte os pedidos do canal original para o formato standardizado da DiF a as respostas inversamente.
  - É o garante da abstração da DiF e seus developers do canal de comunicação das aplicações. As especificidades e complexidades do canal são aqui reestruturadas numa forma simples e comum para os developers. Informações de cliente centralizadas numa forma simples num objecto de cliente, atributos e parametros após tratamento colocados nas versões funcionais da DiF.

Tradutores de pedidos para o standard da DiF. Possibilitam que qualquer desenvolvimento seja idêntico para qualquer canal

Arquitectura da DiF2

ChAL & Dispatcher

- Listener
  - Os listeners são quem recebe os pedidos e envia ao ChAL. Estes são “channel dependent” naturalmente.
- ChAL
  - Converte os pedidos do canal original para o formato standardizado da DiF a as respostas inversamente.
  - É o garante da abstração da DiF e seus developers do canal de comunicação das aplicações. As especificidades e complexidades do canal são aqui reestruturadas numa forma simples e comum para os developers. Informações de cliente centralizadas numa forma simples num objecto de cliente, atributos e parametros após tratamento colocados nas versões funcionais da DiF.
- **Dispatcher**
  - **Cada canal poderá impor uma forma diferente de autenticar, executar, publicar os resultados, etc. Podemos se tal for necessário prover uma implementação alternativa para cada um deles, ou desligá-lo por não ser necessário.**

O fluxo e tarefas para dar resposta a um pedido. Poderá ser custumizado por canal ou mesmo personalizado por um parceiro/developer externo à Digitalis.

Arquitectura da DiF2

ChAL & Dispatcher

- Listener
  - Os listeners são quem recebe os pedidos e envia ao ChAL. Estes são “channel dependent” naturalmente.
- ChAL
  - Converte os pedidos do canal original para o formato standardizado da DiF a as respostas inversamente.
  - É o garante da abstração da DiF e seus developers do canal de comunicação das aplicações. As especificidades e complexidades do canal são aqui reestruturadas numa forma simples e comum para os developers. Informações de cliente centralizadas numa forma simples num objecto de cliente, atributos e parametros após tratamento colocados nas versões funcionais da DiF.
- Dispatcher
  - Cada canal poderá impor uma forma diferente de autenticar, executar, publicar os resultados, etc. Podemos se tal for necessário prover uma implementação alternativa para cada um deles, ou desligá-lo por não ser necessário.
- **Novas implementações**
  - **A DiF recebe novas implementações destes módulos e sem mais configuração necessária trata pedidos nos novos canais. Em teoria um serviço desenvolvido para outro canal poderá ser executado nestes novos canais, tendo apenas a preocupação de que a View que constroi a resposta possa funcionar para o Channel aqui referido.**

O que faz a DiF?

Descrição das principais funcionalidades da DiF

O que faz a DiF?

DEM: DiF Entity Model

- Hierarquia SOA
  - Todos os serviços na DiF são estruturados numa hierarquia:
    - Provider – Application – Service - Stage.
  - Esta hierarquia permite definir e empacotar serviços e decompo-los em passos de execução, stages.
  - A implementação desta hierarquia é agora tão simples como anotar uma classe java com: @ProviderDefinition, @ApplicationDefinition, @ServiceDefinition, @StageDefinition
- O DEM está disponível programáticamente pela API
- Automatismos possíveis
  - UDDI e WSDL: Prevê-se vir a gerar estes protocolos directamente do DEM
  - WSRP/JSR-168: Portlets disponíveis e como os ivocar

```
@StageDefinition(name = "Sample Service", service="sampleservice")
@View(target="sample.jsp")
public class SampleStage {

   @Execute
   public void myMethod() {
      // Do something...
   }
}
```

O que faz a DiF?

Dynamic Code Generation

- Código gerado pela DiF para implementar features convencionadas ou configuráveis
  - Annotation driven. Existem dezenas de anotações possíveis de utilizar
  - Abstrai o developer da utilização directa da maioria da API da DiF
  - Faz com que a DiF de adapte ao código do developer e não o contrário
  - Possibilita escudar o código do developer de evoluções futuras da DiF
  - Possibilita que código existente vá sendo enriquecido com novas funcionalidades, melhorias de performance ou boas práticas instituidas posteriormente ao mesmo sem qualquer intervenção ou alteração do código
  - As implentações são sempre segundo a máxima "convenção vs configuração vs programação", ou seja, existem comportamentos padrão convencionados, várias formas de os ajustar e em última análise podemos altera-los programáticamente

A base de toda a DiF. É a sua mais valia e a base que permite colocar todos os conceitos em prática.

O que faz a DiF?

IoC – Inversion of Control container

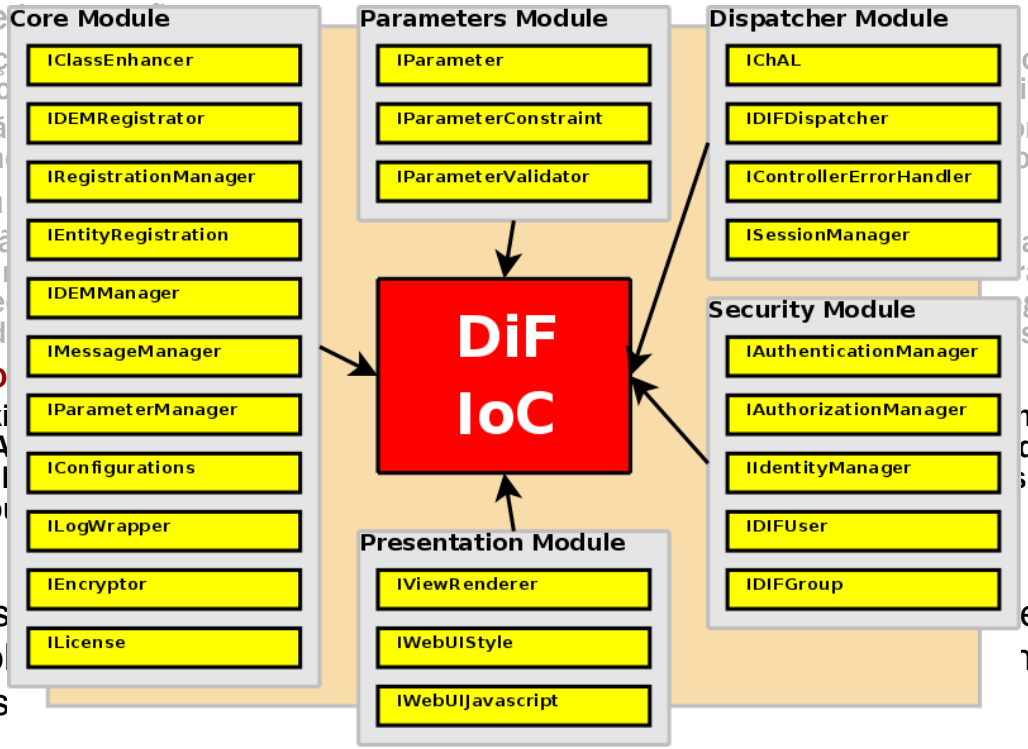
- **Camada de integração**
  - É a peça central da camada de integração da DiF. A própria DiF preveligiando o SoC integra todos os seus módulos por IoC, tornando fácil a sua integração com outras tecnologias. Muito similar à filosofia Spring.
  - Surgirão no futuro novos módulos de integração que permitirão extender a DiF com funcionalidades e integrações com diversas tecnologias/produtos. No entanto qualquer pessoa o pode fazer de forma simples.
- **Plataforma Modular**
  - Tudo são módulos na DiF. Podemos substituir módulos existentes por implementações particulares nossas (i.e.: O nosso próprio motor de autenticação que acede a uma BD proprietária para validar e gerir users). A arquitectura possibilita ainda colecções de contribuições para por exemplo conseguir ter uma lista de possíveis validadores de parâmetros, ou formatos de exportação disponiveis aos developers.

A própria DiF é um conjunto de módulos, logo é muito simples substituir uma implementação default de qualquer parte da DiF por uma outra que a integre com qualquer plataforma externa. O mesmo conceito se aplica a aplicações que sigam esta filosofia com base no que a DiF propícia.

O que faz a DiF?

IoC – Inversion of Control container

- Camada de
  - É a peç
    módulo
  - Surgirã
    integra
- Plataforma
  - Tudo sã
    (i.e.: O
    arquite
    validad
- **Dinamismo**
  - **Não exi**
    num JA
    mais). I
    contribu



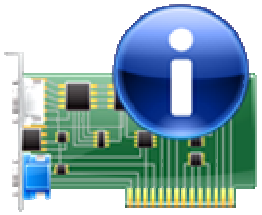
oC integra todos os seus
imilar à filosofia Spring.
om funcionalidades e
ode fazer de forma simples.

ações particulares nossas
ra validar e gerir users). A
guir ter uma lista de possíveis
s.

na nova feature é empacotada
de classes - módulos, nada
s

A ins
ap
exis

ervidor
nte aos
editar!



O que faz a DiF?

AOP – Aspect Oriented Programming

- Integração com AOP
  - A DiF inclui a integração de AspectJ como API de AOP para cross-cutting concerns
- Implementações existentes
  - Logging.
  - Excepções

O que faz a DiF?

Model ORM Generator

- Integração com Hibernate nativa
  - A DiF inclui a integração de Hibernate e geração de classes de Model (através do plugin Maven ORM)

O Hibernate é a solução de ORM mais utilizada. Oferece:

- Abstracção da BD em uso (suporta virtualmente qualquer BD que tenha um driver JDBC)
- Gestão de cache
- Optimização de querys
- Gestão de relações/constraints, atribuição de PKs, etc.
- Gestão de transações, etc.

O que faz a DiF?

Model ORM Generator

- Integração com Hibernate nativa
  - A DiF inclui a integração de Hibernate e geração de classes de Model (através do plugin Maven ORM)
- Geração de todo o ORM por reverse engeneering (reveng) do Modelo de Dados
  - O ORM inclui a camada de Data Objects, Mapping files, DAOs e Services
  - As classes geradas podem ser extendidas pelo developer, não sobrepondo em novas gerações
- Integração por IoC
  - O ORM gerado é acessível por contribuições de IoC automáticamente geradas no processo de reveng

```
@StageDefinition(name = "Sample Service", service="sampleservice")
@View(target="sample.jsp")
public class SampleStage {

    @Inject
    protected IService cseService;

    @Execute
    public void myMethod() {
        int total = cseService.getCursosDAO().findById(1000).getAlunos().size();
    }
}
```

O que faz a DiF?

Model ORM Generator

- Integração com Hibernate nativa
  - A DiF inclui a integração de Hibernate e geração de classes de Model (através do plugin Maven ORM)
- Geração de todo o ORM por reverse engeneering (reveng) do Modelo de Dados
  - O ORM inclui a camada de Data Objects, Mapping files, DAOs e Services
  - As classes geradas podem ser extendidas pelo developer, não sobrepondo em novas gerações
- Integração por IoC
  - O ORM gerado é acessível por contribuições de IoC automáticamente geradas no processo de reveng
- Customização de todo o processo
  - Serviços parametrizados num XML
  - Templates em freemarker
  - Várias configurações oriundas do Hibernate Tools Reverse Engeneering

O que faz a DiF?

Security

- Divisão em módulos das várias necessidades
  - IdentityManager: Users, Grupos, suas relacções e atributos
  - AuthenticationManager: Mecanismos e gestão de autenticação
  - AuthorizationManager: Mecanismos e gestão de credenciais de acesso de users/grupos a serviços

Separação de conceitos: Identidade (quem és?), autenticação (validação de que és quem dizes ser) e autorização (o que podes fazer?). Facilita integrações.

O que faz a DiF?

Security

- Divisão em módulos das várias necessidades
  - IdentityManager: Users, Grupos, suas relacções e atributos
  - AuthenticationManager: Mecanismos e gestão de autenticação
  - AuthorizationManager: Mecanismos e gestão de credenciais de acesso de users/grupos a serviços
- Vários flavors possíveis
  - Actual: Static, LDAP, Database
  - Futuro: Mais LDAPs, Kerberos, outros
- Contribuições externas
  - Mais uma vez é fácil que qualquer developer contribua com implementações específicas de um ou todos estes módulos. Seja para implementar alguma particularidade do seu cliente, ou para integrar com um standard ainda não suportado.
- Ambiente default
  - Criação automática de user e grupos default em qualquer sistema integrado

O que faz a DiF?

Gestão de Parâmetros

- Âmbito (scope) de parâmetros
  - Request: Parâmetros de request que morrem no final do tratamento do mesmo
  - Static: Parâmetros estáticos que são partilhados por todas as instancias da mesmas entidade
  - Session: Parâmetros que se mantêm ao longo de uma sessão
  - User: Parâmetro de utilizador (partilhado pelas várias sessões do mesmo se existirem)

Todas as páginas Web comunicam através da passagem de parâmetros. Um trabalho necessário e muito recorrente

Ex: http://www.server.com?**num=10**&**data=11-11-2010**&**active=true**

O que faz a DiF?

Gestão de Parâmetros

- Âmbito (scope) de parâmetros
  - Request: Parâmetros de request que morrem no final do tratamento do mesmo
  - Static: Parâmetros estáticos que são partilhados por todas as instancias da mesmas entidade
  - Session: Parâmetros que se mantêm ao longo de uma sessão
  - User: Parâmetro de utilizador (partilhado pelas várias sessões do mesmo se existirem)
- Vantagens da gestão de parâmetros da DiF 2.x
  - Colocação em atributos da classe
  - Conversão automática para tipos Java
  - Valores por default
  - Validações com base num sistema extenso e flexível de constraints e validators
  - Gestão do scope automáticamente
  - Herança de parâmetros no DEM
  - Reflexo de tudo isto nos inputs de dados numa form da View
  - Persistência
  - Definição com uma anotação num atributo

Automatizar é a palavra de ordem.
Pedro Viegas a implementar!

O que faz a DiF2

Gestão de

- Âmbito (sc
  - Reque
  - Static
  - Sessio
  - User: I
- Vantagens
  - Coloca
  - Conve
  - Valore
  - Valida
  - Gestã
  - Heran
  - Reflex
  - Persis
  - Defini

```
@StageDefinition(name = "Sample Service", service="sampleservice")
@View(target="sample.jsp")
public class SampleStage {

    @Parameter
    String someStringParameter;

    @Parameter(defaultValue="20", constraints="required,max=100,min=10")
    @Persist(scope=ParameterScope.SESSION)
    Long someLongParameter;

    @Parameter(defaultValue="true")
    @Persist(scope=ParameterScope.USER)
    Boolean someBooleanParameter;

    @Parameter(constraints="required")
    @Persist(scope=ParameterScope.SESSION)
    Date someDateParameter;

    @Execute
    public void myMethod() {
        // Do something...
    }
}
```

tidade

Automatizar é a palavra de ordem.
Pedro Viegas a Pedir um comportamento e não implementá-lo!

O que faz a DiF?

Mensagens

- Definição de mensagens simples
  - As mensagens são definidas em simples ficheiros ".properties". Um por entidade
  - Herança de mensagens pelo DEM
  - Pacotes reutilizáveis de mensagens.
  - Internacionalização automatizada (i.e. SomeStage.messages.en)

O que faz a DiF?

Mensagens

- Definição de mensagens simples
  - As mensagens são definidas em simples ficheiros ".properties". Um por entidade
  - Herança de mensagens pelo DEM
  - Pacotes reutilizáveis de mensagens.
  - Internacionalização automatizada (i.e. SomeStage.messages.en)
- Features & Automatismos
  - Embora as definições sejam desta forma definidas ("no código") será possível customizar as mensagens.
  - Estas customizações serão armazenadas num repositório externo, este repositório pode ser facilmente colocado em qualquer formato, com uma contribuição de IoC para esta feature.

O que faz a DiF?

Mensagens

- Definição de mensagens simples
  - As mensagens são definidas em simples ficheiros ".properties". Um por entidade
  - Herança de mensagens pelo DEM
  - Pacotes reutilizáveis de mensagens.
  - Internacionalização automatizada (i.e. SomeStage.messages.en)
- Features & Automatismos
  - Embora as definições sejam desta forma definidas ("no código") será possível customizar as mensagens.
  - Estas customizações serão armazenadas num repositório externo, este repositório pode ser facilmente colocado em qualquer formato, com uma contribuição de IoC para esta feature.
- Gestão & Performance
  - Gestão de memória
  - Lazy loading

O que faz a DiF?

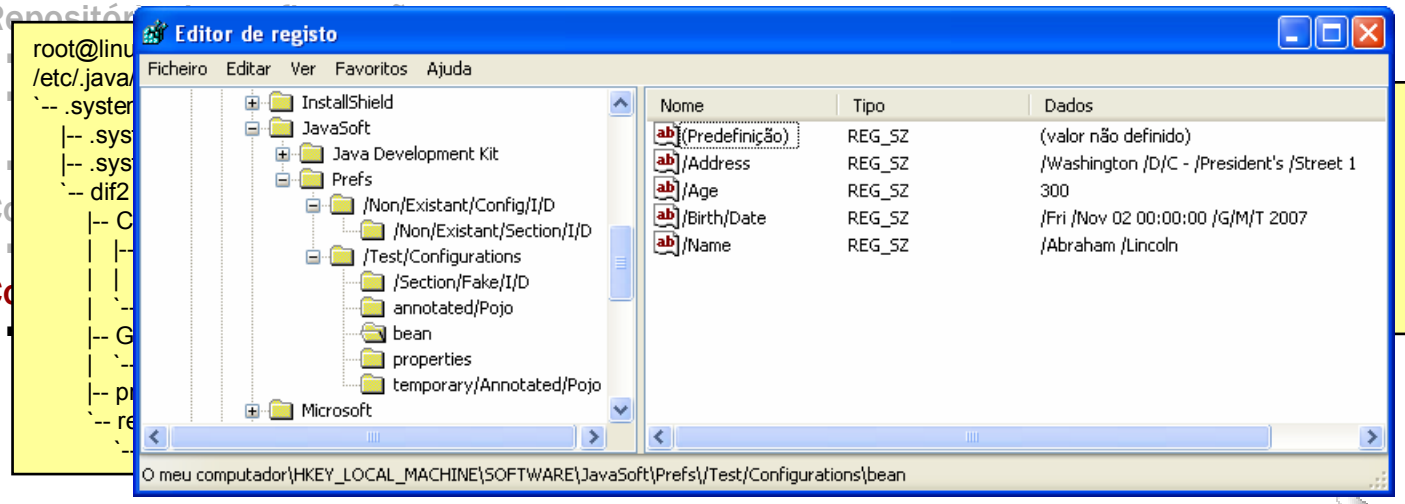
Configurações

- Repositório de configurações
  - Fim de vários XML/properties em várias difectorias, ou BDs de apoio
  - Na DiF os parâmetros by default são guardados no Registry (em Windows) ou numa estrutura de file system (em Unix/Linux), num formato legível e hierárquico, em ficheiros XML
  - Qualquer outra implementação facilmente poderá ser fornecida para usar outro repositório.
- Como poderá um utilizador editar as configurações?
  - Estará disponível um interface na própria DIF para este propósito.
- Como usar pelo Developer?
  - Simples POJOs, ou anotados para configurações não default

Redução da complexidade para developers, centralização para administradores, maior organização e facilidade para ambos.

O que faz a DiF?

Configurações



Redução da complexidade para developers, centralização para administradores, maior organização e facilidade para ambos.

O que faz a DiF?

Web 2.0: Components & Component Library

- Component driven
  - Tags são componentes
  - Views (Páginas) são componentes

Component driven development nativamente suportado pela framework. Em componentes (Tags) e etapas (stages ou páginas)

O que faz a DiF?

Web 2.0: Components & Component Library

- Component driven
  - Tags são componentes
  - Views (Páginas) são componentes
- Component Library
  - Variada biblioteca de componentes de UI
  - Integração de bibliotecas de JavaScript
  - Versões acessíveis e/ou ricas à escolha do utilizador
  - Componentes dinâmicos que geram interfaces ricos com pouquíssimo esforço do developer
    - Grid
    - BeanEditForm

Componentes e mais componentes…
reutilização de inúmeras peças de UI disponibilizadas pela DiF

O que faz a DiF?

Maven: Project Automation

- **Maven, o que é?**
  - O Maven é uma ferramenta para gestão técnica de um projecto de software
  - Foi criada pela Jakarta para permitir padronizar e integrar facilmente a panóplia de todos os projectos da própria. Tem hoje apenas no repositório central perto de 25.000 livrarias

Maven: Criar projecto, obter recursos, empacotar, deploy, testar, instalar, documentar, destribuir, etc!

O que faz a DiF?

Maven: Project Automation

- Maven, o que é?
  - O Maven é uma ferramenta para gestão técnica de um projecto de software
  - Foi criada pela Jakarta para permitir padronizar e integrar facilmente a panóplia de todos os projectos da própria. Tem hoje apenas no repositório central perto de 25.000 livrarias
- Integração das fases de desenvolvimento
  - Todo o processo é gerido pelo Maven, desde:
    - Criação de um projecto baseado num template
    - Obtenção da framework pela net e suas dependências de livrarias (para cada versão das mesmas!)

Ex.: `user@linux:~$ mvn archetype:create…`

O que faz a DiF?

Maven: Project Automation

- Maven, o que é?
  - O Maven é uma ferramenta para gestão técnica de um projecto de software
  - Foi criada pela Jakarta para permitir padronizar e integrar facilmente a panóplia de todos os projectos da própria. Tem hoje apenas no repositório central perto de 25.000 livrarias
- Integração das fases de desenvolvimento
  - Todo o processo é gerido pelo Maven, desde:
    - Criação de um projecto baseado num template
    - Obtenção da framework pela net e suas dependências de livrarias (para cada versão das mesmas!)
    - Criação de um projecto para o IDE (Eclipse ou IntelliJ)

Ex.: `user@linux:~$ mvn eclipse:eclipse…`

O que faz a DiF?

Maven: Project Automation

- Maven, o que é?
  - O Maven é uma ferramenta para gestão técnica de um projecto de software
  - Foi criada pela Jakarta para permitir padronizar e integrar facilmente a panóplia de todos os projectos da própria. Tem hoje apenas no repositório central perto de 25.000 livrarias
- Integração das fases de desenvolvimento
  - Todo o processo é gerido pelo Maven, desde:
    - Criação de um projecto baseado num template
    - Obtenção da framework pela net e suas dependências de livrarias (para cada versão das mesmas!)
    - Criação de um projecto para o IDE (Eclipse ou IntelliJ)
    - Execução de testes unitários ou de integração

Automático, sempre que são gerados os “pacotes” são precedidos da execução dos testes que condicionam o geração do “pacote”

O que faz a DiF?

Maven: Project Automation

- Maven, o que é?
  - O Maven é uma ferramenta para gestão técnica de um projecto de software
  - Foi criada pela Jakarta para permitir padronizar e integrar facilmente a panóplia de todos os projectos da própria. Tem hoje apenas no repositório central perto de 25.000 livrarias
- Integração das fases de desenvolvimento
  - Todo o processo é gerido pelo Maven, desde:
    - Criação de um projecto baseado num template
    - Obtenção da framework pela net e suas dependências de livrarias (para cada versão das mesmas!)
    - Criação de um projecto para o IDE (Eclipse ou IntelliJ)
    - Execução de testes unitários ou de integração
    - Criação da documentação e site

Ex.: `user@linux:~$ mvn site site:deploy…`

O que faz a DiF?

Maven: Project Automation

- Maven, o que é?
  - O Maven é uma ferramenta para gestão técnica de um projecto de software
  - Foi criada pela Jakarta para permitir padronizar e integrar facilmente a panóplia de todos os projectos da própria. Tem hoje apenas no repositório central perto de 25.000 livrarias
- Integração das fases de desenvolvimento
  - Todo o processo é gerido pelo Maven, desde:
    - Criação de um projecto baseado num template
    - Obtenção da framework pela net e suas dependências de livrarias (para cada versão das mesmas!)
    - Criação de um projecto para o IDE (Eclipse ou IntelliJ)
    - Execução de testes unitários ou de integração
    - Criação da documentação e site
    - Controle da qualidade do projecto através de vários relatórios ou validadores

Incluídos relatórios no site, podendo ser criados automatismos para impedir passos se a qualidade não for adequada.

O que faz a DiF?

Maven: Project Automation

- Maven, o que é?
  - O Maven é uma ferramenta para gestão técnica de um projecto de software
  - Foi criada pela Jakarta para permitir padronizar e integrar facilmente a panóplia de todos os projectos da própria. Tem hoje apenas no repositório central perto de 25.000 livrarias
- Integração das fases de desenvolvimento
  - Todo o processo é gerido pelo Maven, desde:
    - Criação de um projecto baseado num template
    - Obtenção da framework pela net e suas dependências de livrarias (para cada versão das mesmas!)
    - Criação de um projecto para o IDE (Eclipse ou IntelliJ)
    - Execução de testes unitários ou de integração
    - Criação da documentação e site
    - Controle da qualidade do projecto através de vários relatórios ou validadores
    - Lançamento do servidor aplicacional (instalando/configurando-o se necessário – JBoss, Jetty, Tomcat e outros)

Ex.: `user@linux:~$ mvn cargo:start…`

O que faz a DiF?

Maven: Project Automation

- Maven, o que é?
  - O Maven é uma ferramenta para gestão técnica de um projecto de software
  - Foi criada pela Jakarta para permitir padronizar e integrar facilmente a panóplia de todos os projectos da própria. Tem hoje apenas no repositório central perto de 25.000 livrarias
- Integração das fases de desenvolvimento
  - Todo o processo é gerido pelo Maven, desde:
    - Criação de um projecto baseado num template
    - Obtenção da framework pela net e suas dependências de livrarias (para cada versão das mesmas!)
    - Criação de um projecto para o IDE (Eclipse ou IntelliJ)
    - Execução de testes unitários ou de integração
    - Criação da documentação e site
    - Controle da qualidade do projecto através de vários relatórios ou validadores
    - Lançamento do servidor aplicacional (instalando/configurando-o se necessário – JBoss, Jetty, Tomcat e outros)
    - Gestão da criação e distribuição dos pacotes de software

Ex.: `user@linux:~$ mvn package deploy release…`

Suportada em tecnologias “Best-of-breed”

Tecnologias base para a DiF2. Reutilização e integração.

JEE 5.0	Core
Servlet API	View Renderer
Tag libraries	View Renderer
AspectJ	AOP
Guice (Google)	IoC
Javassist (JBoss)	Bytecode enhancement
JUnit, HttpUnit, easymock…	Testing
Hibernate (JBoss)	ORM (PostgreSQL, mySQL, Oracle, SQL Server, DB2, etc…)
Maven (Apache)	Project Managment
Ext JS	JavaScript library
JNDI	LDAP

Projectos satélites

Projectos que originaram de necessidades da DiF2

- IoC Utils
  - Implementação de um IoC container. Abstrai e extende as suas funcionalidades
  - Fornece actualmente uma implementação baseada no Google Guice
- Configuration Utils
  - Implementação de uma API para gerir pontos de configuração baseado em POJOs
  - Fornece actualmente uma implementação baseada no Preferences API da Sun
- Bytecode Utils
  - Implementação de uma API para bytecode enhancement. Simplificação da tarefa.
  - Fornece actualmente uma implementação baseada em Javassist da JBoss
- LDAP Utils
  - Implementação de uma API para gestão de um servidor LDAP.
  - Extende a informação possível de armazenar num LDAP
  - Normaliza o acesso a features não padronizadas nas implementações dos LDAPs existentes.
  - Baseado em JNDI
  - Suporta actualmente: Active Directory (AD).
  - Em breve: Oracle Internet Directory (OID) e OpenLDAP.

Projectos satélites

Projectos que originaram de necessidades da DiF2

- Logger
  - Implementação de uma logging API com funcionalidades estendidas.
  - Fornece actualmente uma implementação baseada no Log4J da Apache
- ISS
  - Gestão de processos sincronos/assincronos.
  - Permite gerir os recursos do servidor para atingir sempre a máxima performance sem causar carga exagerada.
- Maven plugins
  - Archetypes Templates para geração de novos projectos
  - Packager Geração de pacotes com source, livrarias ou livrarias executáveis automatizada
  - ORM Generator Reverse engeneering de um ORM partindo de um modelo de dados numa RDBMS

Tools

Ferramentas de produtividade

- Maven
- ORM Generator
- DiF IDE
  - Baseado em Eclipse
  - Bundled com um conjunto de plugins seleccionado para o desenvolvimento com as tecnologias da DiF

Recursos

Recursos aos DiF adopters

- Sites dos vários produtos
  - http://development.digitalis.pt
- Mailing list
  - http://development.digitalis.pt/apache2-default/dif2/dif-core/mail-lists.html
- Issue tracking
  - http://development.digitalis.pt/apache2-default/dif2/dif-core/issue-tracking.html
- Wiki
  - http://development.digitalis.pt/mediawiki/
- Serviços prestados pela Digitalis
  - DiF startup-kit – In progress
  - User Guide (livro em formato digital) – In progress
  - Formação em desenvolvimento sobre a DiF
  - Consultadoria sobre desenvolvimento sobre a DiF
  - Out-sourcing de DiF developers
  - Suporte sobre desenvolvimento sobre a DiF

Quem está a usar a DiF

DiF Adopters

- Digitalis Informática
- COFAC – Cooperativa de Ensino (Universidade Lusófona)
- FCUL – Faculdade de Ciências da Universidade de Lisboa
- Empresa de retalho (Prospect, não concretizada)
- Projecto open-source (Prospect, não concretizada)

O que vem a seguir?

Próximos desenvolvimentos e iniciativas previstas.

- WebServices como um Channel
- Cache, WebCache
- AJAX nativo
- View Renderers: PDF, Freemarker
- Portlets (JSR-168, WSRP) como um Channel
- Interfaces de administração de todos os plugins previstos pela DiF
- Workflow (BPEL provavelmente)
- Reporting framework
- Auditoria de serviços e dados (integrada na ORM generation)
- DiF IDE (Wizards e editores de produtividade) – Potenciar o RAD

Q&A

Obrigado

Departamento de Desenvolvimento

http://development.digitalis.pt

Digitalis Informática ©2008

http://www.digitalis.pt