Atitit.工作流系统的本质是dsl 图形化的dsl 4gl

[1. 工作流系统的本质是dsl 图形化的dsl 4gl 1](#_Toc30365)

[2. 为什么每个项目系统都需要工作流 1](#_Toc9693)

[3. 工作流dsl与java .net 这类语言的关系 1](#_Toc12531)

[4. 一个完整的工作流标准如下 2](#_Toc11486)

[4.1. 增强标准 2](#_Toc27268)

[5. BPM系统由三部分组成： 2](#_Toc17036)

[6. 工作流的适应性与优缺点 2](#_Toc22520)

[7. 工作流的趋势：： 3](#_Toc3544)

[8. 现有工作流jbpm 微软wwf的情况 3](#_Toc23998)

[9. 如何判断图灵完备 3](#_Toc23985)

[10. 工作流节点存储结构 4](#_Toc596)

[11. ---end 4](#_Toc3236)

# 工作流系统的本质是dsl 图形化的dsl 4gl

工作流是dsl ，是4gl，是图形化的dsl， 图形化的高层编程语言

工作流引擎即是vm。。。专门用来运行工作流这个dsl图形化编程语言

# 为什么每个项目系统都需要工作流

工作流dsl可以图灵完备，普通的项目都只支持seq顺序流程（用户层面上），并且用户自定义业务逻辑比较困难（必须使用实际的编程语言，对用户来说难度过高）

工作流WF 属于4gl ，大大提升了抽象层次，未来一定会普及。。成为通用的编程语言。。

作者:: 绰号:老哇的爪子 （ 全名：：Attilax Akbar Al Rapanui 阿提拉克斯 阿克巴 阿尔 拉帕努伊 ） 汉字名：艾龙，  EMAIL:1466519819@qq.com

转载请注明来源： http://blog.csdn.net/attilax

# 工作流dsl与java .net 这类语言的关系

Workflow wf是建立在java .net上的一种dsl ，，java 。Net又是建立在c c++之上的vm虚拟机的一个dsl。。。。C、c++则是建立在asm汇编语言之上的dsl

每一个层次抽象提升，会带来更高的开发效率。。

# 一个完整的工作流标准如下

图灵完备（具有seq,case,for等控制结构）

变量 数组 数据等临时数据存储单元

自定义子过程（子函数）

必要的算数，逻辑，移位等运算单元

## 增强标准

Oo支持

对频繁调用jit inline优化

# BPM系统由三部分组成：

1. 流程配置工具ide WorkFlowTool（简称WFTool），
2. 2、流程引擎 vm 虚拟机 WorkFlowEngine(简称WFEngine),
3. 3、业务应用平台C/S(WinApp)或B/S(WebApp)版 开发平台

# 工作流的适应性与优缺点

工作流并非面向普通用户，而是鉴于普通用户和专业编程人员之间的一类用户，他们没有能力使用专业的java c#这类型编程语言，但是又需要进行编程。。所以工作流 dsl 就是比较适合他们的编程语言了。。

工作流的缺点主要在于性能：特别是对于md5，base64这类编码算法流程，但是如果将来加入jit即时编译，应该性能可以得到极大提升。

许多工作流是基于图形展示的，存储是基于xml格式的。导致代码复用可能会不容易拷贝粘贴。。

此外如果逻辑复杂的话，图形会可能比文本化的dsl长度长很多。。如果工作流采用文本方式存储和展示，可能又会造成可视化方面的下降。。

# 工作流的趋势：：

不管怎么说，随着硬件性能的提升，4gl终将会流行。。。就像java c#虚拟机随着硬件性能的提升而取代了c++的大部分地位。。

工作流将会分为俩个部分，第一部分是图形化编程，适合于非专业人员。。

第二部分是基于文本源码的DSL编程，适合于专业编程人员。。

# 现有工作流jbpm 微软wwf的情况

jbpm 的最大问题是不是图灵完备的，不能作为一个通用图形化编程语言来实现所有系统。。

Wwf 是图灵完备的，可以实现所有业务系统。。但是没有一个好的图形化标准化类库来支持快速开发。。此外wwf的存储是基于xml的，只有图形化的实现。。而没有基于文本源码的实现模式，这导致了专业编程人员使用起来不方便。。

此外，vm虚拟机方面的差距也很大，工作流引擎的性能优化才刚刚起步，jit等即时编译技术都没应用到，导致了工作流如果编写细节性的代码（特别是底层算法类的代码），性能消耗比较大。。

Ide方面，工作流的ide 比较烂，根本还不能和vs eclipse系列相比。。工作流ide方面还有很大的提升空间。。

# 如何判断图灵完备

简单判定图灵完备的方法就是看该语言能否模拟出图灵机  
  
图灵不完备的语言常见原因有循环或递归受限(无法写不终止的程序,如 while(true){}; ), 无法实现类似数组或列表这样的数据结构(不能模拟纸带). 这会使能写的程序有限

图灵不完备也不是没有意义, 有些场景我们需要限制语言本身. 如限制循环和递归, 可以保证该语言能写的程序一定是终止的.

# 工作流节点存储结构

Id ：对应语言里面的标号，行址

Name： 显示名称

Ui\_id 对应的界面ui id

Next 下一个节点id （预定义有下一个id，调用子函数返回断点id ，case for结构block块结束id）

State 是否在此节点。。原来debug方便。。

# ---end

什么是图灵完备？ - 编程语言 - 知乎.htm