

# COMP3029J 课程

## 软件架构

### 软件架构风格概述 (虚拟机风格)

邓永健

北京工业大学计算机学院

数据挖掘与安全实验室 (DMS 实验室)



# 大纲

虚拟机风格

2. 口译员

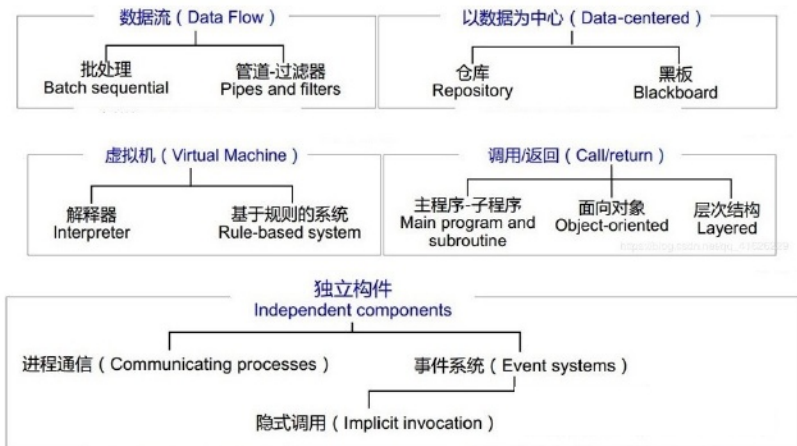
3. 基于规则的系统

4. 摘要



北京工业大学  
BEIJING UNIVERSITY OF TECHNOLOGY

# 风格分类学



# 大纲

## 1. 虚拟机风格

## 2. 口译员

## 3. 基于规则的系统

## 4. 摘要



北京工业大学  
BEIJING UNIVERSITY OF TECHNOLOGY

# 虚拟机风格

## 口译员

模拟硬件本身不具备的功能

## 基于规则的系统

口译员的专业化

## 其他

- 句法外壳
- 命令语言处理器



北京工业大学  
BEIJING UNIVERSITY OF TECHNOLOGY

# 大纲

## 1. 虚拟机风格

## 2. 口译员

## 3. 基于规则的系统

## 4. 摘要



北京工业大学  
BEIJING UNIVERSITY OF TECHNOLOGY

# 口译风格

**问题：**此模式适用于无法直接获取最适合执行解决方案的语言或机器的应用程序。该模式也适用于核心问题在于定义用于表达解决方案的符号表示法的应用程序，例如脚本。解释器有时会按顺序串联使用，在一系列阶段中从所需的语言/机器翻译为可用的语言/机器。

解释器通常会被设计成在目标机器或语言与执行环境中已支持的某种（可能是虚拟的）机器或语言之间架起桥梁。



# 解释器风格

## 解决方案

- 系统模型：虚拟机

组件：一个状态机（执行引擎）和三个存储器（执行引擎的当前状态、正在解释的程序、正在解释的程序的当前状态）

- 连接器：数据访问和过程调用

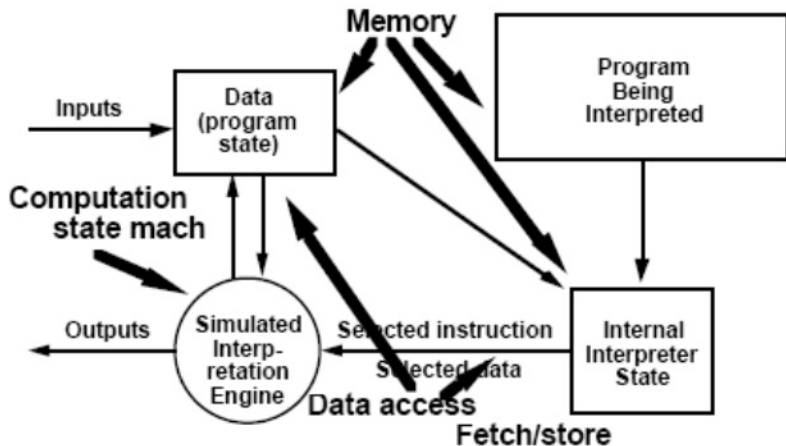
- 控制结构：通常执行引擎采用状态转换；输入驱动用于选择要解释的内容

**重要变体：**专家系统通常被实现为规则集或产生式集合的解释器，这些规则或产生式代表了专业知识。由于产生式需要复杂的选择规则，因此专门形式的解释器应运而生。





# 口译风格



# 解释器风格的优点

## 功能

- 可模拟非原生功能

## 测试

- 能够模拟“灾难”模式（例如用于安全关键型应用）

## 灵活性

- 非常通用的工具



# 解释器风格的缺点

## 效率

比硬件慢得多

比编译系统慢得多

## 测试

- 需要验证的额外软件层



北京工业大学  
BEIJING UNIVERSITY OF TECHNOLOGY

# 解释器风格的应用

## 解释型语言

- VB、JavaScript、VBScript、HTML、Java 字节码、Matlab
- 脚本、配置文件

## 通信协议

## 用户输入

游戏中的组合键



北京工业大学  
BEIJING UNIVERSITY OF TECHNOLOGY

# 大纲

虚拟机风格

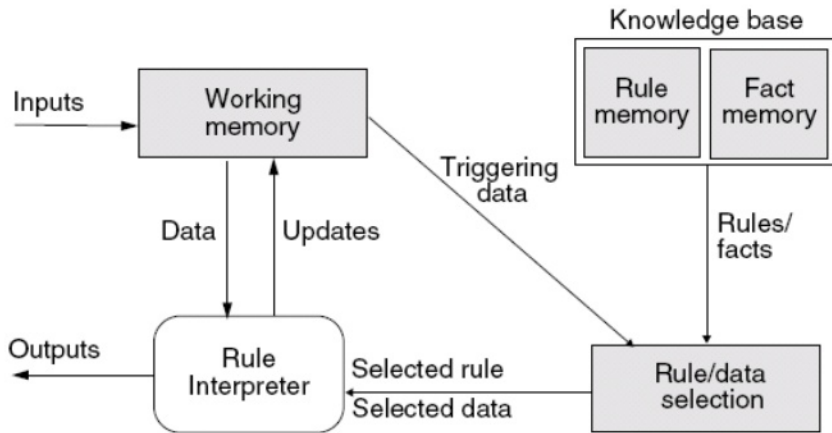
2. 口译员

**3. 基于规则的系统**

4. 摘要



# 基于规则的系统



# 基于规则的系统的系统特点

待执行的代码（知识库）

解释引擎（规则解释器）

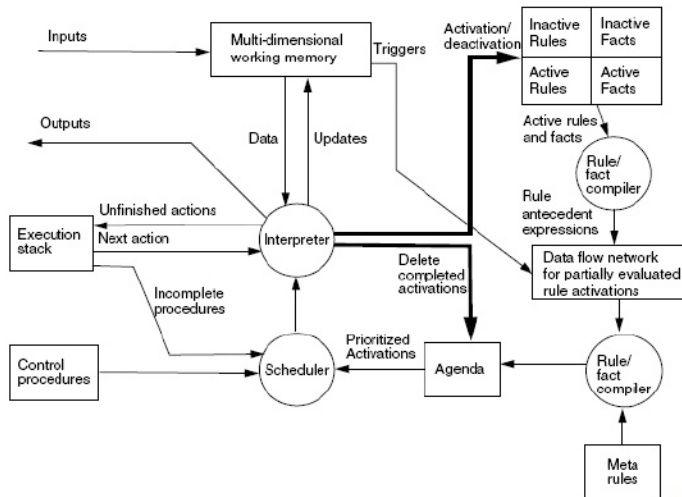
解释器的控制状态（规则/数据选择）

当前代码的状态（工作内存）



北京工业大学  
BEIJING UNIVERSITY OF TECHNOLOGY

# 复杂规则系统（示例）



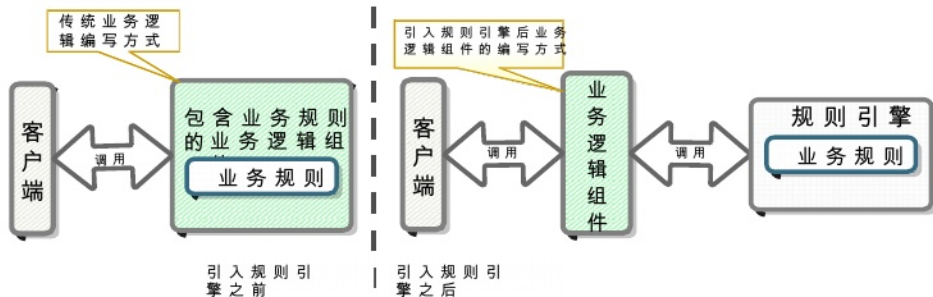


# 基于规则的系统的應用

Drools 是红帽公司旗下 JBoss 的业务逻辑智能模块。

· [www.jboss.org/drools/](http://www.jboss.org/drools/) （此网址无需翻译）

Drools 是一个用于处理规则的专家系统。



北京工业大学  
BEIJING UNIVERSITY OF TECHNOLOGY

# 基于规则的系统应用

- 声明式编程

规则引擎让您能够说明“做什么”，而非“怎么做”。

基于规则的系统能够解决非常复杂的难题。

- 逻辑与数据分离

快速且灵活

- 知识集中化
- 工具集成
- 良好的解释机制
- 易于理解的规则



北京工业大学  
BEIJING UNIVERSITY OF TECHNOLOGY

# 基于规则的系统应用

Java语言表示，如果有一个人的名字是“Joe”，而且是个男性，就会输出他的名字跟性别。

```
if ( "Joe".equal( people.getName() ) )
{
    if( "Male".equal( people.getSex() ) )
    {
        System.out.println("This is a man, name is Joe.");
    }
}
```

```
rule "GoodBye"
when
    People( name = "Joe", sex = "Male" )
then
    System.out.println(("This is a man, name is Joe.");
end
```



北京工业大学  
BEIJING UNIVERSITY OF TECHNOLOGY

# Drools 的背景

## 庞大的系统和复杂的业务

- 业务规则经常变更
- 24 小时服务
- 统一的企业管理
- 降低系统维护和升级成本



北京工业大学  
BEIHANG UNIVERSITY OF TECHNOLOGY

# Drools 的背景

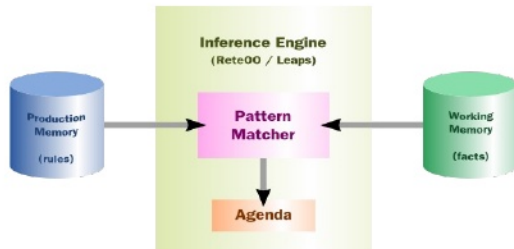
许多基于产生式规则的系统的“大脑”实际上是一个推理引擎，用于匹配**事实**和**规则**。

当匹配成功时，将**触发**（执行）与该规则相对应的**动作**。

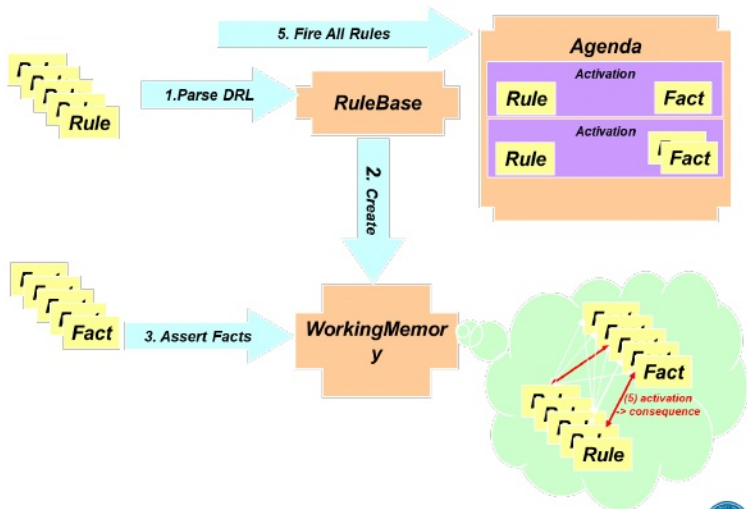
## · 行动

经常改变事实的状态，或者

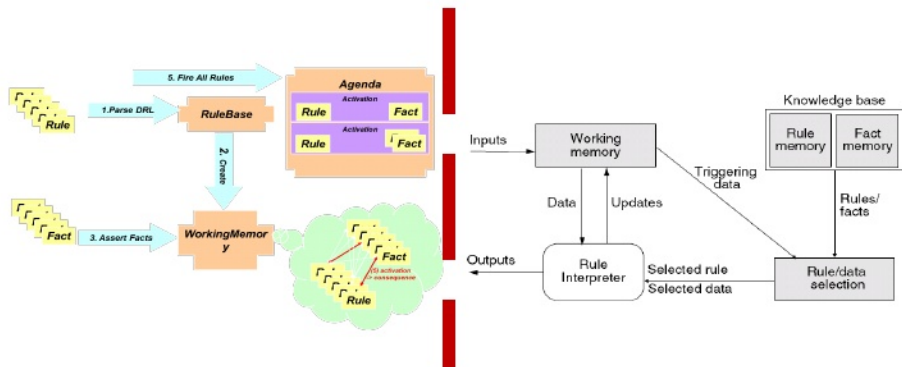
对应用程序执行一些“外部”操作



# Drools 架构



# Drools 与基于规则的系统



# 大纲

虚拟机风格

2. 口译员

3. 基于规则的系统

4. 总结



北京工业大学  
BEIJING UNIVERSITY OF TECHNOLOGY



# 摘要

## 解释器

模拟硬件本身不具备的功能

## 基于规则的系统

口译员的专业化

## 其他

- 句法外壳
- 命令语言处理器



北京工业大学  
BEIJING UNIVERSITY OF TECHNOLOGY

# 摘要

## 解释器

模拟硬件本身不具备的功能

## 基于规则的系统

口译员的专业化

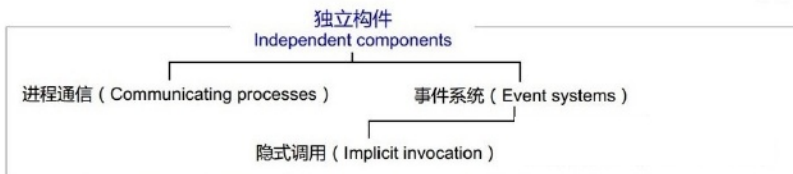
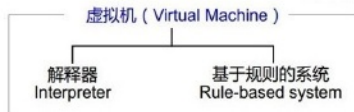
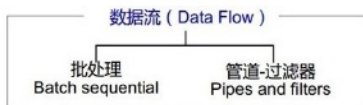
## 其他

- 句法外壳
- 命令语言处理器



北京工业大学  
BEIJING UNIVERSITY OF TECHNOLOGY

# Summary



北京工业大学  
BEIJING UNIVERSITY OF TECHNOLOGY

# 谢谢你!

