

TRUSTIE

国家软件资源共享与协同生产环境 软件生产线集成框架参考模型

中科院软件所

2009年1月16日

简介

软件生产线集成框架概述

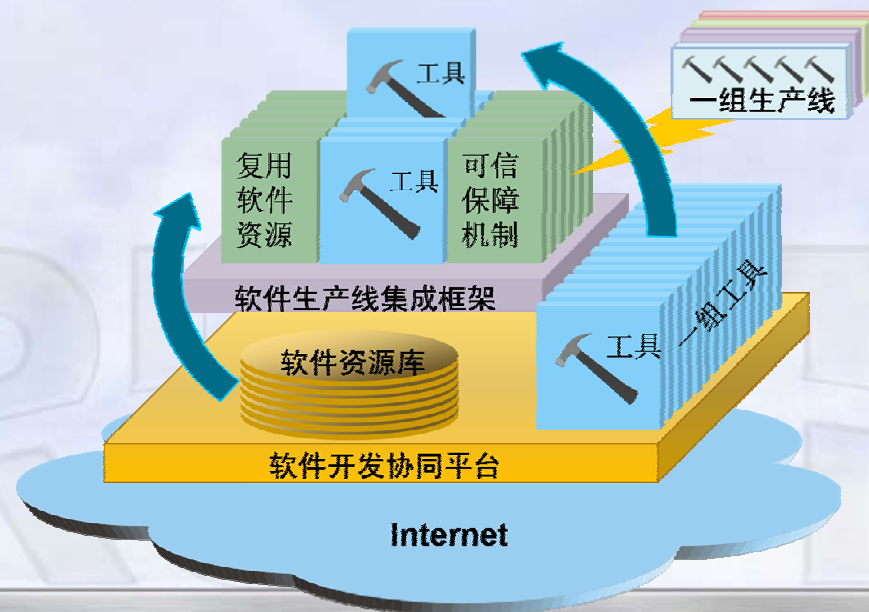
参考模型

后续工作

规范背景

“以可信软件生产活动为中心，研究可信软件生产的基本特点，针对工具的集成化、可扩展、协同化开发需求，研究灵活可扩展的软件工具集成框架、生产线数据集成方法和机制，实现生产线与各类工具、软件资源库和开发协同平台之间的集成方法和交互机制，支持生产线的灵活集成，进而形成软件工具与生产线集成的系列国家行业标准，规范软件工具、制品的集成和共享能力”

——《可信的国家软件资源共享与协同生产环境项目申请书》



规范目的

- 阐述生产线工具集成类型及其属性特征
- 明晰软件生产线中的几类实体及其集成关系
- 定义软件生产线集成框架参考模型
 - ▶ 软件生产线工具参考模型，为工具开发人员提供工具开发的参考方案
 - ▶ 软件生产线制品的参考模型，为工具开发人员提供生产线制品表示和操纵的参考方案
 - ▶ 软件生产线通信的参考模型，为工具开发人员提供工具使用的参考方案
- 为软件生产线开发人员提供使用集成框架构建软件生产线的方案
- 指导软件生产线集成框架相关子规范的编写，包括（暂定）：
 - ▶ 《软件生产线工具规范》
 - ▶ 《软件生产线制品规范》
 - ▶ 《软件生产线集成框架基础服务规范》

规范相关定义

■ 软件生产线

- ▶ 根据特定领域需求，按照一定软件生产过程，使用软件生产线集成框架，将一系列软件工具集成起来而形成的一个软件产品开发环境

■ 软件生产线集成框架

- ▶ 一个符合软件生产线集成框架规范的工具集成总线，并具有软件工具生命周期管理、软件制品管理和工具通信管理等支持工具集成的基础服务，支持符合软件生产线工具规范的工具集成

■ 软件工具

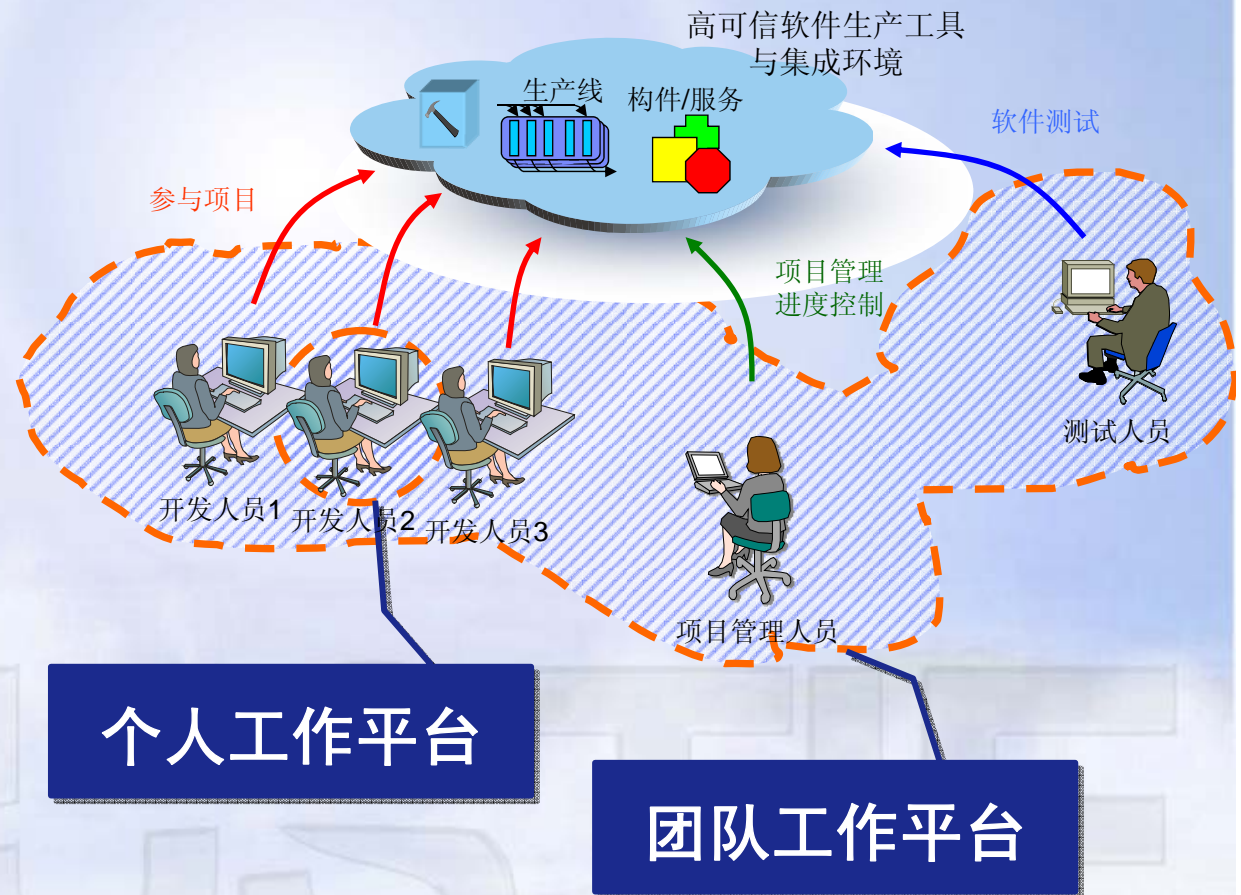
- ▶ 一类符合软件生产线集成框架规范的软件插件，其功能以插件服务的形式发布

■ 软件制品

- ▶ 符合软件生产线工程模型和数据模型的软件相关实体

软件生产线使用方式：个人与团队

- 个人工作平台
- 集成框架依照项目成员的角色职能，集成相关工具，形成个人开发平台
- 团队工作平台
- 个人开发平台依托协同平台、资源库，集成为团队工作平台，支持资源共享和协同开发



软件生产线使用示例

■ 协同平台创建项目工程

- 项目管理人员通过协同平台定义软件生产过程，包括软件制品加工工序、团队分工、制品管理等

■ 项目成员通过协同平台进行制品管理

- 项目成员通过协同平台，按照角色划分，从制品管理库获取待加工制品
- 项目成员利用集成框架安装、加载相关工具，进行制品加工

■ 项目团队进行协同开发

集成框架如何支持制品管理？

- 团队成员在获得协同平台的通知后，通过制品管理库获得待加工制品
- 团队成员完成制品加工后，向制品管理库提交制品，同时通知协同平台，更新软件生产过程的状态

集成框架如何支持

工具的安裝、加载和运行？

集成框架如何支持

个人工作平台、协同平台、制品管理库之间的交互？

简介

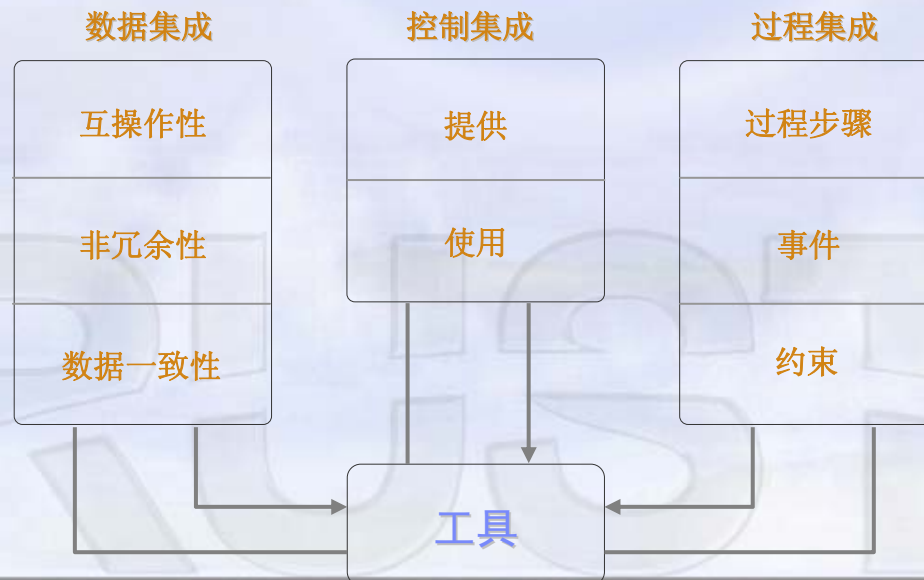
软件生产线集成框架概述

参考模型

后续工作

工具集成类型和属性特征

- 面向**数据**的工具集成
 - 保证工具间数据的互操作性、非冗余性和一致性
- 面向**控制**的工具集成
 - 从提供和使用两方面阐述软件生产线工具的功能组合机制
- 面向**过程**的工具集成
 - 显示描述工具间的调用关系和软件制品间的转换关系及约束



面向数据的工具集成

■ 互操作性

- ▶ 工具间数据的互操作性是为了保障数据表示的一致性。对于软件生产线的使用者而言，该性质体现为工具间数据具有公共视图。

■ 非冗余性

- ▶ 非冗余性强调工具间数据的冗余程度。

■ 数据一致性

- ▶ 在工具对数据进行处理时，数据一致性是为了维护各工具数据间的约束。

面向控制的工具集成

■ 提供

- ▶ 控制关系中的提供关系是指被集成的两工具间，一个工具提供自己的服务供另一个工具使用。
- ▶ 为了满足工具间的提供关系，要求工具在实现时必须具有良好的接口描述。可以为其它工具所发现、识别和使用。

■ 使用

- ▶ 控制关系中的使用关系是指被集成的两工具间一个工具使用另一个工具所提供的服务。
- ▶ 为了满足工具间的使用关系，要求工具在实现时必须具有清晰的模块化特征。

软件生产线中的集成关系

■ 基本集成关系

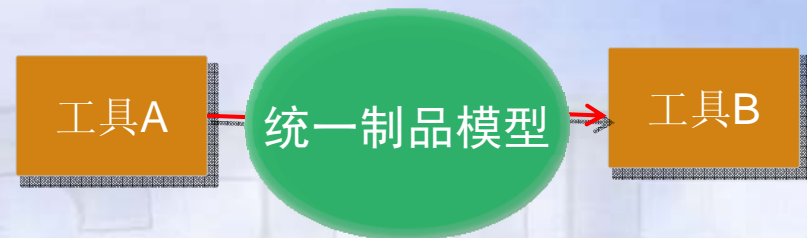
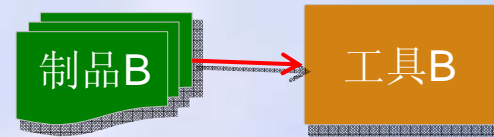
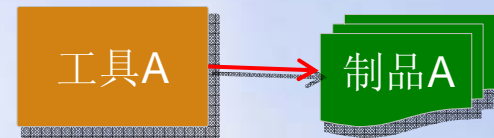
- ▶ 工具之间的**软件制品共享关系**
- ▶ 工具之间的**服务调用关系**

■ 集成框架的集成关系

- ▶ 集成框架和软件工具之间的**制品管理关系**
- ▶ 集成框架和软件工具之间的**生命周期管理关系**
- ▶ 集成框架和软件工具之间的**服务调用关系**

工具之间的软件制品共享关系

- 软件工具加工或生产某种特定的软件制品
- 不同软件工具需要共享某些软件制品
 - ▶ 两个软件工具同时加工同一个软件制品
 - ▶ 一个软件工具将生产的软件制品输出给另一个软件工具作为加工过程的输入



生产线必须定义软件工具间
可以共同理解的软件制品模型

工具之间的功能调用关系

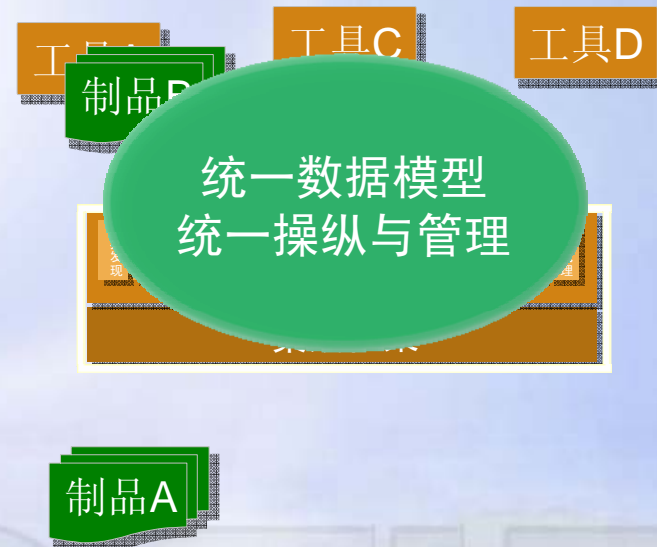
- 软件工具之间存在使用与被使用的关系
- 软件工具存在多种功能发布方式
 - ▶ API
 - ▶ Web service
 - ▶ . . .
- 软件工具间存在多种调用方式
 - ▶ 本地调用
 - ▶ 远程调用



生产线必须有统一的方式来发布和调用软件工具的功能

集成框架与软件工具间的制品管理关系

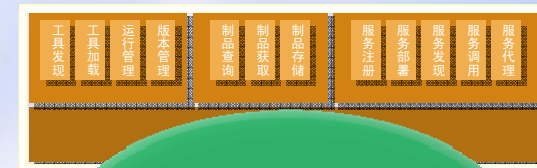
- 集成框架是软件制品传递的通道
- 集成框架为软件工具提供制品管理服务
 - ▶ 软件制品发现
 - ▶ 软件制品获取
 - ▶ 软件制品存储



**集成框架需要定义
统一的软件制品数据模型
和软件制品操纵与管理模型**

集成框架与软件工具间的生命周期管理关系

- 集成框架是软件工具运行的载体
- 集成框架为工具提供生命周期管理服务
 - ▶ 软件工具的发现、识别
 - ▶ 软件工具的加载
 - ▶ 软件工具运行态管理
 - 启动
 - 挂起
 - 停止
 - ...

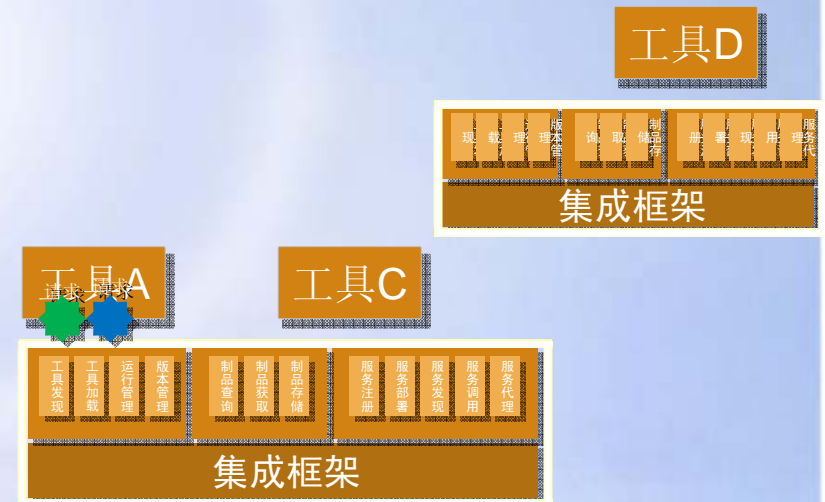


统一工具表示模型
工具管理模型

集成框架需要定义统一的
软件工具表示模型
和软件工具管理模型

集成框架与软件工具间的服务调用关系

- 集成框架是软件工具通信的媒介
- 集成框架为工具提供通信服务
 - ▶ 多种通信方式
 - ▶ 本地访问
 - ▶ 远程访问



集成框架需要定义
统一的软件工具通信模型

简介

软件生产线集成框架概述

参考模型

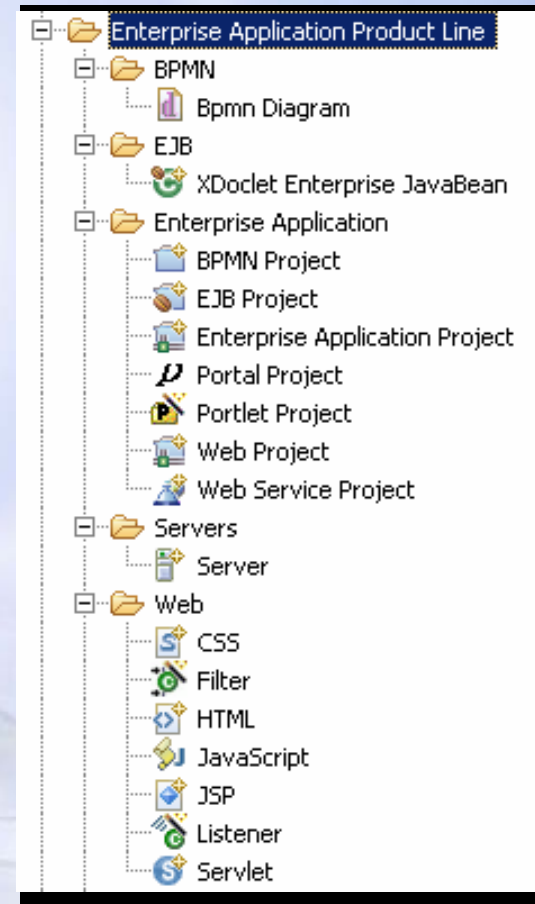
后续工作

软件生产线集成框架中参考模型

- 软件制品模型
 - ▶ 软件制品数据模型
 - ▶ 软件制品操纵与管理模型
- 软件工具模型
 - ▶ 软件工具表示模型
 - ▶ 软件工具管理模型
- 通信模型

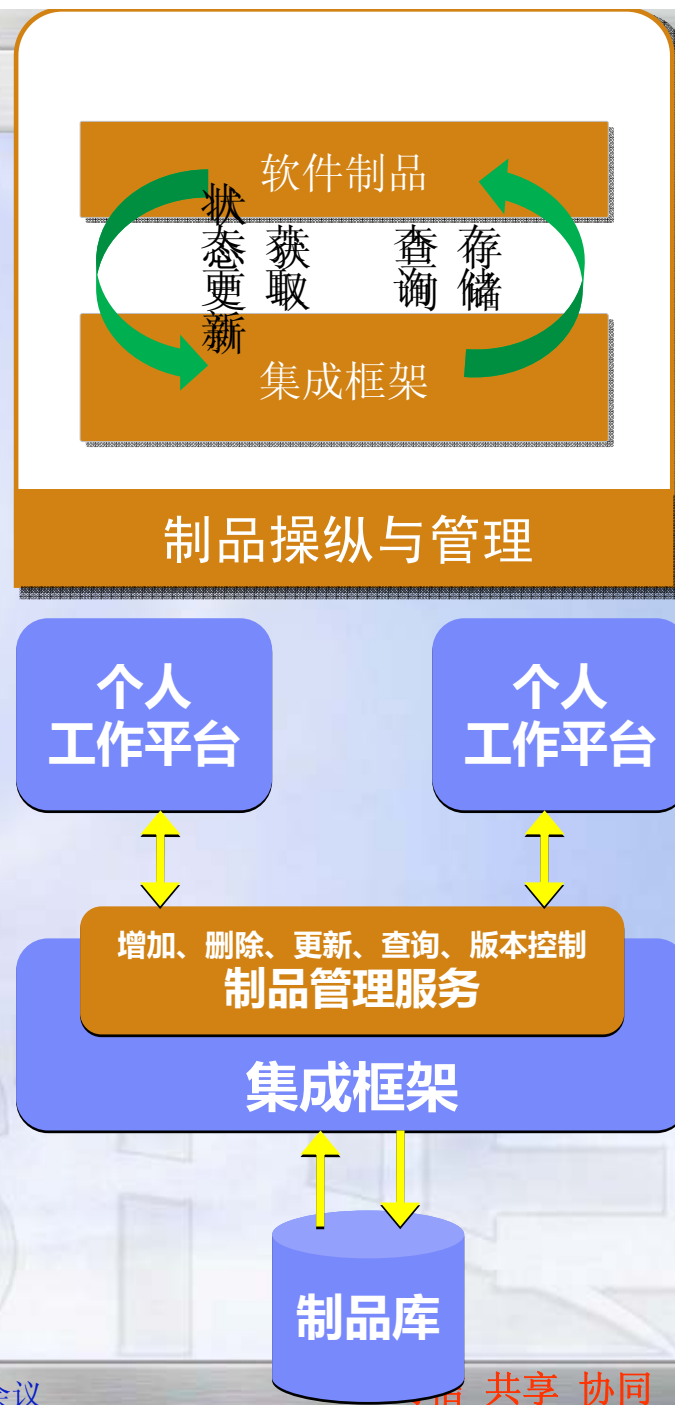
软件制品数据模型

- 生产线中表现为软件生产线工程模型
 - ▶ 工程模型是开发过程中所有软件制品的**组织与描述模型**。
 - ▶ 包含工程模型**元模型**，刻画工程模型的基本元素及其关系
 - ▶ 软件制品**描述模型**，刻画软件制品的特征
- 根据工程模型元模型，可以生成领域特定的软件生产线工程模型



软件制品操纵与管理模型

- 集成框架提供软件制品管理
 - ▶ 查询
 - ▶ 访问
 - ▶ 更新
 - ▶ 存储
 - ▶ . . .
- 软件制品通过集成框架通知相关软件工具软件制品状态
- 软件工具使用集成框架提供的软件制品管理和操纵服务来使用软件制品



软件工具表示模型

- 集成需求要求软件工具必须要符合一定的设计规范
 - ▶ 软件工具以插件形态存在，满足工具结构模块化的要求
 - ▶ 软件工具功能以服务形式进行提供，满足工具功能服务化的要求
 - ▶ 软件工具在表现形式上为JAR包
 - ▶ 通过软件工具配置描述文件定义软件工具相关服务接口



软件工具JAR包结构

```

Manifest-Version: 1.0
Bundle-ManifestVersion: 2
Bundle-Name: Log Plug-in
Bundle-SymbolicName: com.example.log
Bundle-Version: 1.0.0
Bundle-Activator: com.example.log.internal.Activator
Bundle-Vendor: EXAMPLE
Import-Package: org.osgi.framework;version="1.3.0"
Export-Package: com.example.service
    
```

软件工具配置文件

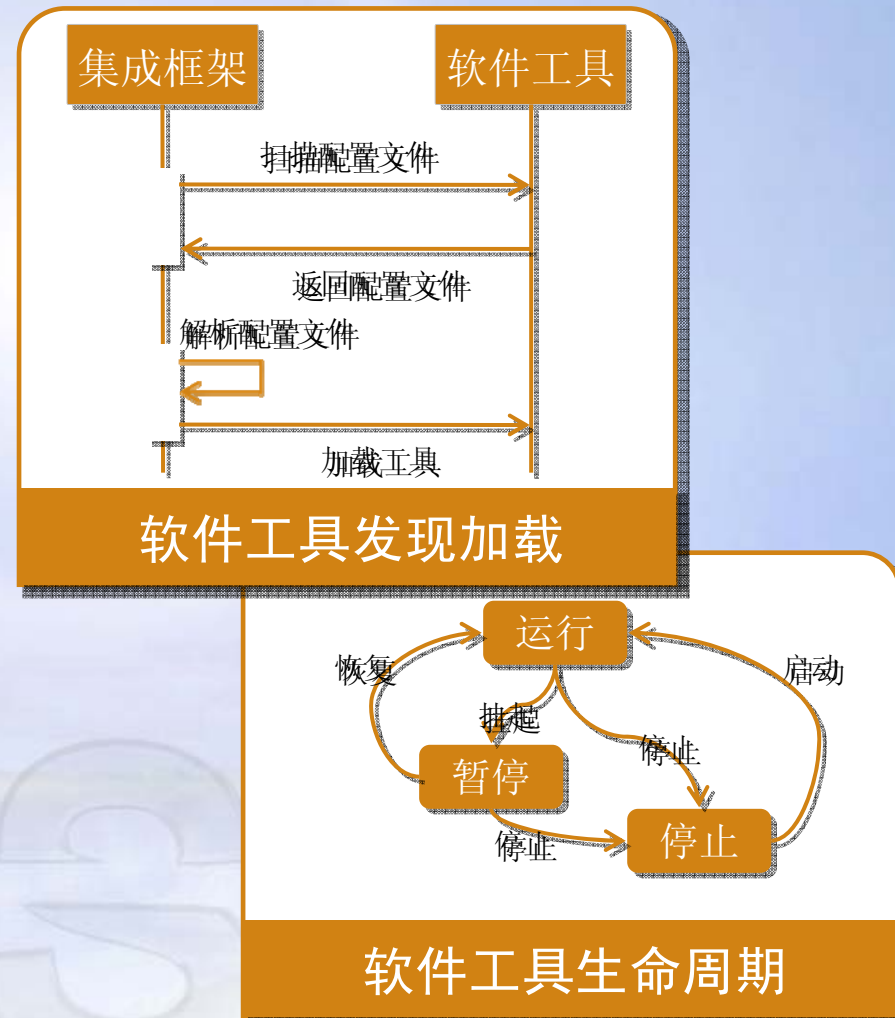
软件工具管理模型

■ 软件生产线构建

- ▶ 集成框架发现软件工具
- ▶ 集成框架加载软件工具

■ 软件生产线运行

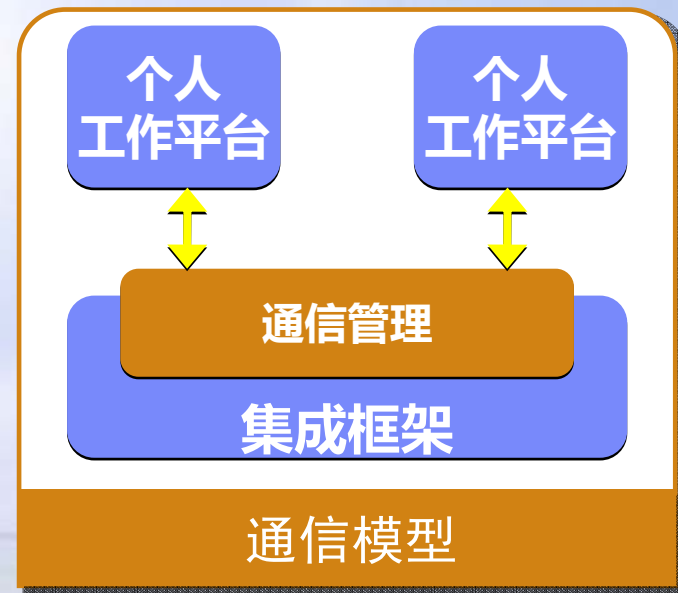
- ▶ 生产线集成框架可以通过软件工具的管理接口对软件工具的进行生命周期管理



通信模型

■ 两种通信模式

- ▶ 软件生产线集成框架加载与开发人员角色相符合的若干软件工具，在开发者的机器上构成**个人工作平台**
- ▶ 软件生产线中建立个人开发平台后，进行开发活动协同，在团队中形成**团队工作平台**



集成框架参考体系架构

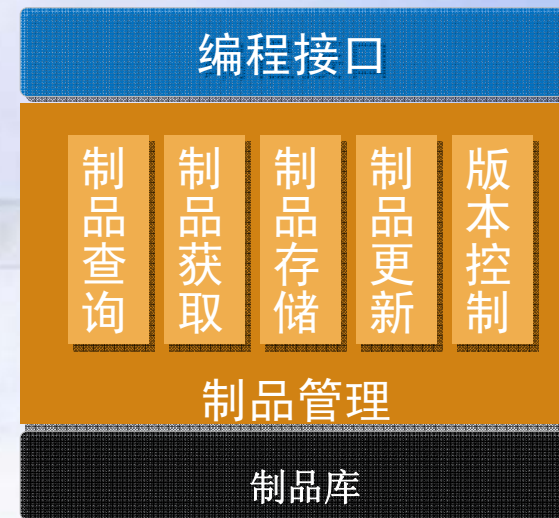


基于OSGi的微内核集成总线

- “微内核+扩展体”的插件体系结构
 - ▶ 模块化
 - ▶ 可扩展
 - ▶ 微内核组织、管理、运行插件
 - ▶ 插件作为外围设备运行在微内核之上

软件制品管理服务

- 软件制品管理服务由一个软件制品库和若干编程接口组成
- 集成框架的软件制品管理服务通过编程接口提供一系列的软件制品管理与操作能力
 - ▶ 软件制品查询服务
 - ▶ 软件制品获取服务
 - ▶ 软件制品存储服务
 - ▶ 软件制品更新服务
 - ▶ 软件制品版本控制服务



软件工具管理服务

- 软件工具生命周期两大阶段
 - ▶ 静态的插件由一组资源文件构成
 - ▶ 运行时插件为集成框架中的实例
- 使用集成框架框架构建生产线
 - ▶ 发现静态工具插件
 - ▶ 加载静态工具插件
 - ▶ 通过版本控制，可以实现高度松耦合的系统
- 生产线启动后产生相应的运行时软件工具插件实例
 - ▶ 为每一个运行时软件工具插件分配一个工具上下文
 - ▶ 每个软件工具插件都可以使用该上下文获取框架提供的各种信息和功能
 - 框架内部发布的事件
 - 其他软件工具插件发布的服务等

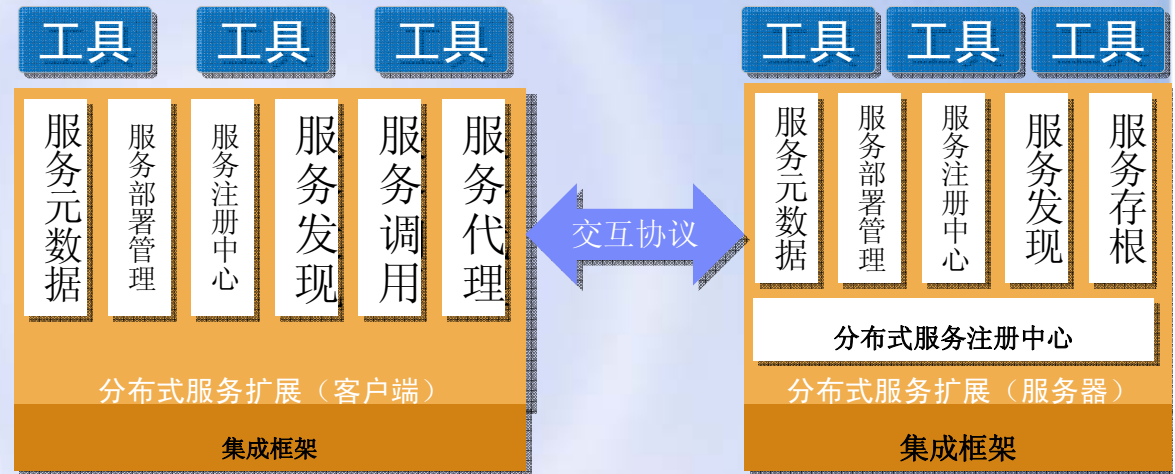


通信服务

- 通信服务分为六个模块

- ▶ 服务部署管理
- ▶ 服务注册中心
- ▶ 服务发现
- ▶ 服务调用
- ▶ 服务代理和存根
- ▶ 分布式服务注册中心

- 分布式工具服务注册和本地工具服务注册保持一致性
- 工具服务发现和调用也与本地工具服务调用保持一致性
- 通信管理服务支持多种通讯协议，实现工具服务交互



规范进一步完善工作

■ 参考实现（2009年6月）

- ▶ 通过参考实现，研制API级的系列子规范

■ 子规范

- ▶ 《软件生产线工具规范》（2009年3月）
- ▶ 《软件生产线集成框架基础服务规范》（2009年6月）
- ▶ 《软件生产线制品规范》（2009年9月）

规范推广工作

- 发布参考实现
- 技术咨询与支持
- 规范执行要求
 - ▶ 在工具规范发布后，按照规范开发、提供工具
 - ▶ 在基础服务规范发布后，使用集成框架原型及其服务进行工具集成
 - ▶ 根据生产线特点提供相应生产线的工程模型

谢谢!

TRUSTIE