

# 面向服务集成的自动化服务注册方法

李松犁 张型龙 肖俊超  
(中国科学院软件研究所 北京 100190)

**摘 要** 服务集成的目标是将多个服务集成到同一系统中以实现特定的业务需求。在集成过程中,由于工具、服务通常有着不同的数据标准、传输机制,导致了集成的困难。为了解决该问题,在服务集成系统中引入了企业服务总线 ESB(Enterprise Service Bus)。目前 ESB 系统的配置文件能够表达较为复杂的服务接入场景,但是根据服务的设计、架构不同,服务接入方法不同,也不支持集成系统的热部署功能。给出一种用于描述工具、服务信息的模型 SDModel(Service Description Model),并在该模型基础上提出一种面向服务集成的自动化服务注册方法。方法能够利用 SDModel,自动化地解析拟接入集成系统的第三方服务和工具,并实现热部署,供应用层使用。

**关键词** 服务集成 企业级服务总线 自动化注册

**中图分类号** TP3 **文献标识码** A **DOI**:10.3969/j.issn.1000-386x.2016.06.015

## SERVICE INTEGRATION-ORIENTED AUTOMATIC SERVICE REGISTRATION METHOD

Li Songli Zhang Xinglong Xiao Junchao  
(Institute of Software, Chinese Academy of Science, Beijing 100190, China)

**Abstract** The goal of service integration is to integrate multiple services into one system in order to implement specific business requirement. In the process of integration, since tools and services usually have different data standards and transfer mechanisms, these lead to the difficulty of integration. To solve the problem, we introduced the enterprise service bus (ESB) to services integration system. Currently the configuration files of ESB system can express some complicated services access scenario. However, according to the difference of services designs and architectures, the services access ways are different as well, they also not support the hot deployment function of the integration system. This paper introduces SDModel, it is a model for describing the information of tools and services. Based on this model, we propose a services integration-oriented automatic services registration method, it can makes use of SDModel, automatically parses the third party services and tools to be accessed to the integration system, and realises the hot deployment for the use in application layer.

**Keywords** Service integration Enterprise service bus Automatic registration

## 0 引 言

在软件开发中,通常会有多种第三方提供的工具、服务被使用。通过集成这些工具和服务,能够提升软件开发的质量和效率。然而由于工具、服务往往由不同开发商提供,集成中很难共享数据和统一管理。

服务集成的目标是将三方服务与工具集成到同一系统中使用<sup>[1]</sup>。在集成过程中,系统能够满足不同类型的工具、服务接入的需求,不需要更改自身的代码。集成后,系统能够复用三方工具的页面以及它所提供的服务,减少开发、集成的工作量。也可以根据特定的业务需求,将服务组合后提供给用户使用<sup>[2]</sup>。

进行系统集成时,由于工具、服务通常有着不同的系统设计、数据标准、传输机制等原因,其互相之间的通信、集成较为困难。为了解决这个问题,在服务集成系统中引入了企业服务总线 ESB<sup>[3]</sup>,为不同的工具、服务间提供一个通信的桥梁。

系统的集成分为服务的集成和工具的集成两种情况。当集成三方服务时,首先将服务接入 ESB,通过 ESB 提供的通信协议、数据格式转换功能,形成新的服务提供给系统使用。该过程

中,需要向 ESB 提供其能够识别、使用的配置文件。配置文件中包含了服务的输入输出、地址等信息。当集成三方工具时,除了将工具所提供的服务接入 ESB 外,还要将使用工具所需的必要信息如用户名密码、工具的地址等提交给系统,以便系统复用该工具的页面。

现有 ESB 系统所提供的集成方法并不能较好地满足服务集成系统的需求,具体问题如下:

- 1) ESB 配置文件的格式和内容复杂,对用户不友好。配置文件包含许多 ESB 内部组件,如数据格式转换模块等。这类组件和接入服务并不直接相关。另外如果配置错误,用户需要花费时间来检查和调试。
- 2) 由于工具、服务的种类繁多,集成中需要针对其不同的设计、架构,编写不同的集成模块,这将花费大量的时间。
- 3) 集成系统不支持热部署功能,导致在集成工具、服务过

收稿日期:2015-01-19。国家自然科学基金项目(91318301,91218302,61432001);国家高技术研究发展计划项目(2012AA011206)。李松犁,硕士生,主研领域:服务集成。张型龙,硕士生。肖俊超,副研究员。

程中,系统无法使用。只有在集成完成后,重新启动系统,才能够使用新的工具和服务。

针对上述问题,本文提出了面向服务集成的自动化服务注册方法。该方法定义了一个能够表达接入工具、服务信息的模型,利用该模型自动化的完成注册工作。

### 1 相关工作

一些研究提出了新的服务组合算法,解决了在 Web 服务注册到 ESB 后,如何利用 workflow 系统,根据需求选择服务,封装组合后形成新的服务提供给用户使用的问题<sup>[4,5]</sup>;一些研究关注于如何对 ESB 系统上的 Web 服务进行管理<sup>[6-8]</sup>;Tan 等人按照 Web 服务的用途分类后,使用 Petri 网计算,得到服务的组合方式<sup>[9]</sup>;也有的工作设计了一个分布式高可信 ESB 系统,在这个系统中解决了服务的组合问题<sup>[10]</sup>。上述研究重点解决了如何根据用户的需求,整合服务的问题。但是这些研究以服务已经注册到 ESB 为前提,没有把工具和服务的注册问题作为重点。

王路远提出了一个 ESB 管理系统<sup>[11]</sup>,该系统通过界面引导用户完成服务的配置后,自动生成 ESB 的配置文件,来完成服务的注册。该方法降低了用户注册和管理 ESB 的难度,但是没有关注集成系统如何使用三方工具的问题。

边小凡等人提出了一种快速集成工具、服务的系统模型<sup>[12]</sup>。该模型使用 ESB 来完成各部件之间的通信工作,并应用到了国家安全以及灾难响应的场景中。文中使用了 UDDI<sup>[13]</sup>来进行服务的发现和注册,但没有重点阐述集成系统如何使用三方工具的问题。

本文提出了一个面向服务集成的自动化服务注册方法,其能够解决上文描述的三个问题,方便接入新工具和服务。其包括下面两个步骤:

- 1) 从实际的工具、服务中抽象出一个较为通用的服务描述模型,其能够表达工具、服务的集成信息,以及它们的设计、架构差异;
- 2) 设计自动化注册方法,利用上述模型所含有的数据完成服务、工具的注册,并保证系统能够使用它们。

### 2 自动化服务注册方法

由于工具和服务的数量众多,为每一个都设计特定的注册方法将耗费大量的时间,因此需要根据实际使用的软件开发工具、服务,抽象出一个较为通用的服务描述模型 SDModel。

图 1 是注册方法的架构设计。首先根据需要集成的工具、服务的信息抽象,由用户填写或可视化模块生成 SDModel。模型解析模块会验证 SDModel 的合法性,如果不符合系统的要求,集成过程将不会被完成。此后 SDModel 会被解析成两部分数据,一是符合 ESB 规范的配置文件,二是存储后供查询的工具、服务信息。完成 SDModel 解析后,解析模块向服务管理模块发出通知,将生成的 ESB 配置文件交由 ESB 部署模块使用。通过使用热部署模块,文件被配置到 ESB 中。这个步骤中,由于诸多原因可能造成部署的失败,因此在部署前,状态保存模块会将正确运行的配置文件保存备份。一旦部署失败,服务管理模块会取出备份文件交给部署模块尝试重新部署。在部署服务中,

服务地址管理模块负责分配和回收地址供服务使用。

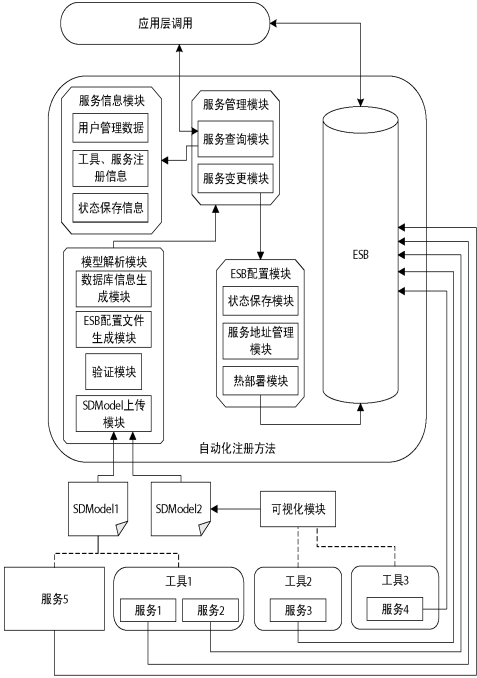


图 1 注册方法设计图

部署完成后,服务管理模块会通知应用层模块新的工具、服务接入完毕。应用层模块通过服务管理模块查找需要的服务,而后会使用接入总线服务所使用的协议和总线进行通信。到这里,ESB 就完成数据格式、通信协议的统一工作,将不能够通信的三方工具、服务连接到了一起。

为了让自动化注册方法实际工作,需要:

- 1) 定义 SDModel,描述待集成的工具和服务;
- 2) 模型解析模块,用于解析 SDModel,生成 ESB 能够识别、使用的配置文件;
- 3) 服务信息模块,保存工具、服务的信息以及 ESB 的配置文件备份;
- 4) ESB 配置模块,将生成的配置文件部署到运行中的 ESB 中,并保证 ESB 的正确运行;
- 5) 服务管理模块,向应用层提供已经集成的服务,同时控制 ESB 配置模块进行部署操作;
- 6) 可视化模块,帮助用户生成 SDModel。

#### 2.1 服务描述模型

本文参考了 WSDL(Web Service Description Language)<sup>[14]</sup>定义服务描述模型,该模型符合标准的 XML 格式( < key > value < /key > )。SD-Model 包含了拟接入的三方服务和工具信息。SDModel 的层次结构应该如图 2 所示。

首先使用 sdmodel 关键字

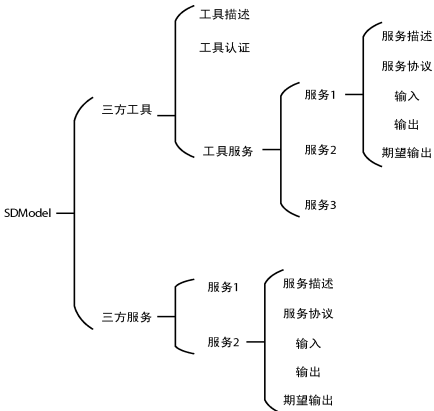


图 2 SDModel 层次结构

来定义模型。所有接入工具、服务等等信息都将写在这个关键字之中。

1) 三方服务

对于待集成的三方服务集合,使用服务集合 (services) 来描述。对于其中的某项具体服务,使用服务 (service) 来描述。三方服务包括了服务描述、服务协议、输入、输出、期望输出几部分。

• 服务描述:包括了服务名称 (name)、简要描述 (description) 和服务地址 (address) 三方面的信息。这里地址应当包括协议和地址两部分。如: `http://192.168.10.10/xmltojson`。其中 `http` 为协议,后面部分为具体的地址。

• 服务协议:使用 `protocol` 关键字。服务的数据经过 ESB 处理后,需要再次提供给其他模块使用,如果用户想在这里改变数据传输的协议,应该使用 `protocol` 关键字。如果没有,则默认为服务地址中的协议。

• 输入:使用 `input` 关键字,描述服务的输入参数。由于输入中可能包含了若干的参数,定义 `param` 关键字来描述这些参数。每个参数由参数名称 (name), 参数格式 (param-format) 和参数取值 (param-value) 几部分组成。参数格式取值如 XML、Object 等。如果参数取值为空,则表示该参数将在 ESB 运行中传入。

• 输出:使用 `output` 关键字,描述服务的输出结果。由 `param` 关键字描述,但是不包含参数取值。

• 期望输出:使用 `expect-output` 关键字,描述用户希望通过 ESB 处理后输出的结果。由 `param` 关键字描述,但是不包含参数取值。

2) 三方工具

三方工具由两部分信息组成:一是工具自身的信息,如账户、工具地址等;二是工具提供服务的信息。使用工具 (tool) 描述用具定义的开始和结尾。

• 工具描述:包括了工具名称 (name)、简要描述 (description) 和工具地址 (address) 三方面的信息。

• 工具认证:用 `tool-access` 关键字描述,定义三方工具账户信息,其中可以包含多个账户 (account)。每个账户中提供工具的用户名、密码。和服务的输入输出类似,账户中使用参数 (param) 来描述具体的用户名和密码。此外,对于 B/S 架构的工具<sup>[15]</sup>,还可以添加工具的登录信息,如工具的登录的地址,来帮系统进行单点登录操作。

• 工具服务:表示由工具提供的若干服务。该部分和三方服务集合 (services) 类似。不同的是这些服务可能没有具体的地址,故可以不配置地址 (address)。

3) 示例

下文展示了一个详细的 SDModel,其中包含了缺陷管理工具 Bugzilla 的工具地址,同时提供一个用户名为 `123@example.com`,密码为 `123` 的账号供集成系统使用。此外,SDModel 中还定义了 Bugzilla 提供的 `getBugById` 服务。该服务的输入为 `int` 格式的 `bugId`,而输出为 `xml` 格式的具体 `bug`。

```
<sdmodel>
<tool>
  <name> Bugzilla </name>
  <address> http://192.168.10.10/bugzilla </address>
  <tool-access>
    <account>
      <param>
        <name> username </name>
        <param-format> string </param-format>
```

```
      <param-value> 123@example.com </param-value>
      </param>
    </account>
  </tool-access>
</services>
<service>
  <name> getBugById </name>
  <protocol> http </protocol>
  <input>
    <param>
      <name> bugId </name>
      <param-format> int </param-format>
    </param>
  </input>
  <output>
    <param>
      <name> bug </name>
      <param-format> xml </param-format>
    </param>
  </output>
</service>
</services>
</tool>
</sdmodel>
```

通过 SDModel 的定义,其能够较为通用地表达三方工具、三方服务的信息。

2.2 注册方法模块实现

1) 模型解析模块

解析模块首先会接收用户上传的 SDModel,然后会验证其正确性。

为了便于 SDModel 的解析和验证,这使用了 JAXP (Java API for XML Processing)<sup>[16]</sup>。JAXP 是一套用来处理 XML 文档的 API。其中包括了 `javax.xml.validation`、`javax.xml.parsers` 等多个包 (package)。

验证分为两个阶段:第一阶段通过 `javax.xml.validation` 包来验证 SDModel 是否符合标准 XML 文件的格式;第二阶段验证 SDModel 的完整性,如 `<tool>` 中是否含有 `<address>` 信息, `<param>` 中是否遗漏了 `<param-format>` 等。如果模块发现 SDModel 有错误,则模块不会进行下一步的 ESB 配置文件生成工作。同时,会将具体的错误信息返回给用户。该模块避免了用户错误填写配置文件后,系统的不能正常运行问题,也减少了进行错误的排查修正的时间。

而后,利用 `javax.xml.parsers` 包解析 SDModel。将解析出的数据写成 ESB 的标准配置文件。根据 ESB 的不同,其配置文件的格式也不同。为了解决这个问题,模块对常见的 ESB 配置文件进行单独处理,并且维护其配置文件的格式信息。如 `mule` ESB 的配置文件开始和结束的标识为 `<mule>` `</mule>`,且通过 `xmlns` 来关联需要的其他 XML 文件<sup>[17]</sup>。随后,该文件将会发送至 ESB 配置模块。另外,SDModel 还包含的工具、服务信息,如服务所需要的输入输出、是否需要格式的转换等,将会被储存

起来,供服务管理模块查询使用。

#### 2) ESB 配置模块

为了达到集成系统的热部署功能,同时保证 ESB 的正常运行,ESB 处理模块会有三个主要的功能:

(1) 服务地址管理模块。当新的服务接入到集成系统中后,需要为其分配相应的服务地址。该模块维护一个地址池,该地址池由集成系统管理员根据拥有的 IP 段进行分配。地址池按顺序为新的服务和工具分配 IP,同时回收不再使用、注销的服务的 IP。

(2) 热部署模块。ESB 配置文件生成后,需要把该文件部署到 ESB 运行环境中。对于一些 ESB,如 Synapse,只需将该文件放入运行环境下;而对于 mule,需要将配置文件打包为发布版后,再放入运行环境。随后重启 ESB 系统(根据 ESB 的不同,处理方式稍有差别,如 mule ESB 不用重启便可完成新服务的部署,而 synapse 需要重启),使新的服务和工具生效。这里使用脚本来控制 ESB 的启动、停止等工作。模块首先会判断用户所使用的操作系统。在 Linux/Mac 系统下采用 shell 脚本;在 Windows 系统下则采用 bat 脚本。另外,由于可能出现的网络异常、配置文件错误、ESB 系统异常等原因,部署可能会失败。因而每次部署前,应当保存当前 ESB 上正常运行的配置文件。部署失败时,向状态保存模块请求恢复之前的状态。

(3) 状态保存模块。由于 ESB 在部署过程中可能出现的异常情况,在每次热部署之前,该模块会记录下当前 ESB 系统中的配置信息。如果系统发生了错误,本模块会负责取回最近一次的信息进行恢复。同时,还应该提醒用户部署发生错误,以及错误原因。

通过 ESB 配置模块,当集成系统接入新的工具和服务时,不再需要重启整个系统,保证了用户的使用。

#### 3) 服务信息模块

服务信息模块将会主要保存三部分信息:

(1) 用户管理数据,用于记录用户的信息,如用户名、密码等。

(2) 工具、服务的注册数据。这部分信息将会用于服务管理模块的查询和变更工作。当集成系统需要多个工具的单点登录功能时,该部分还会记录工具的登录地址等信息,以便集成系统模拟登录取回工具的 cookie。

(3) 状态保存信息。该模块负责维护 ESB 的配置文件,以及 ESB 是否正常工作等数据。通过设置时间间隔,模块将定时从 ESB 抓取当前系统的运行信息。部署失败时,模块将取出最近一次正常工作的配置文件,以便 ESB 系统恢复运行。

#### 4) 服务管理模块

该模块主要包含了两个功能:

(1) 服务查询模块。当应用层模块需要调用 ESB 上的服务时,会首先向该模块发出申请。模块查询 ESB 是否存在该服务,以及该服务运行是否正常,然后把查询的结果返回给应用层模块。最后应用层模块会根据该结果决定是否对服务发起 http、或是其他协议的请求。例如前文中 Bugzilla 提供的 getBugById 服务,由于该服务的协议是 http,应用层将会对该服务的地址发起 http 请求。在请求中包含有服务需要的参数 bugId。ESB 响应请求后,从 Bugzilla 查找 bug 数据,最后把该 bug 返回给应用层。此外,每次部署新工具、服务成功后,模块会向应用层模块发出消息,通知其可以使用这些工具和服务。

(2) 服务变更模块。当用户不再需要某个服务或是想对服

务做出修改时,该模块会对保存的服务信息做出相应的变更。如注销 getBugById 服务,该服务的数据则会被删除。同时模块会定位该服务所属的 ESB 配置文件,在该文件中去除不需要的服务项,最后通知 ESB 配置模块,重新部署更新后的 ESB 配置文件。

#### 5) 可视化模块

对于用户来说,手动编写 SDModel 仍有一定的难度,因此可以根据具体集成系统的需求,为用户提供两种方式完成注册:

(1) 对话输入框的方式,引导用户填写注册信息。

(2) 直接上传包含有接入工具、服务信息的 SDModel。此后所有的工作由模块自动完成。该界面将会在下一章的案例分析中给出示例。

## 3 案例分析

### 3.1 软件开发可信证据采集平台介绍

本文所提出的自动化服务注册方法已经应用在软件开发可信证据采集平台上。该平台是一个 Web 应用工具集成系统,主要用于将软件生命周期的相关工具,如需求管理工具 Jira、缺陷管理工具 Bugzilla、测试工具 TestLink 等,集成到同一系统后提供给用户使用。

在用户使用的过程中,这些接入的工具将产生许多和软件生产相关的数据。平台可以利用这些数据,按照可信证据的定义计算出该软件的可信度。此外,在工具集成的基础上,平台还允许用户在为该平台定制的工作流规则下组合工具所提供的服务。例如同类工具之间的数据传输,将 Bugzilla 中的缺陷数据同步到 Mantis 中。该平台的简单架构图如图 3 所示。

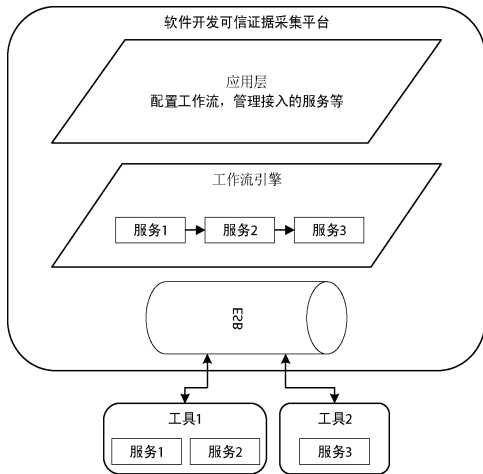


图3 采集平台架构

在使用平台之前,用户需要在平台注册一系列工具和服务。在使用中,用户可以选择从平台进入某个工具操作。也可以利用平台的工作流系统定制出一套符合自己业务需求的流程,如通过项目中需求和测试的差值得黑盒覆盖率。最后由工作流引擎调用 ESB 上所接入的服务,返回结果给用户。在使用平台前,需要接入软件生命周期中的各项工具和服务。传统的注册方法需要平台针对每一个接入的工具、服务都开发一套通信组件,这增大了平台开发的负担。另一方面,传统的集成方法不能够解决集成的工具之间互相的通信问题。例如希望 Bugzilla 和 Jira 通信,则需要为两个工具单独的开发通信组件。那么当 Bugzilla 希望和另外 10 个工具通信时,开发工作量和难度都将增加。在引入 ESB 后,所有工具、服务间的通信都通过总线来

完成,只需要为它们开发和 ESB 通信的组件即可,大大节省了集成的工作量。

这时,本文提出的注册方法就能够解决上文所述的集成场景。基于以上原因,平台使用了本文所提出的自动化注册方法,以支持平台访问工具,以及工作流引擎调用服务的需求。

3.2 平台注册方法

在该平台下,用户可以选择两种方式来接入三方工具和服务:一是直接上传 SDModel;二是通过可视化界面的引导,填写信息后,由平台生成 SDModel 供自动化注册方法使用。平台使用了 mule ESB,该 ESB 提供了为接入工具而设计的 connector 机制。用户可以将三方工具提供的 API 封装为 connector,接入到 ESB 后提供 Web 服务。因此当用户需要为平台接入新工具,且该工具没有提供现成的服务时,还需要上传封装好的 connector。

3.3 可视化界面和单点登录

为了进一步降低用户在接入工具、服务时的困难,平台设计了一套友好的可视化界面来引导用户完成注册工作,如图 4 所示。

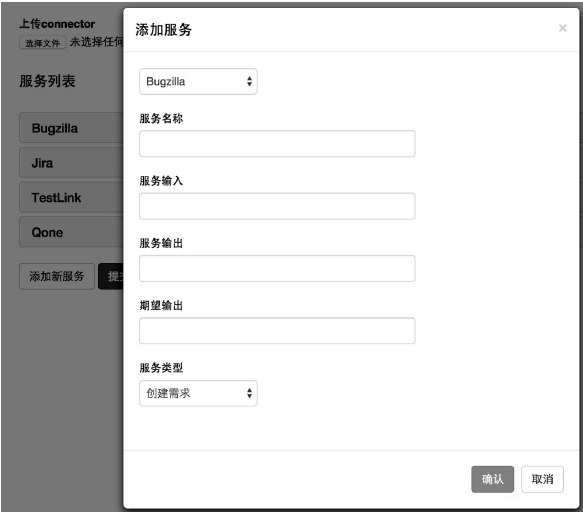


图 4 服务注册界面

用户在界面上填写工具、服务的信息后,平台会生成 SDModel 文件,供注册方法使用。

另一方面,用户还需要直接在登录平台的同时登录接入的三方工具,因此这里需要一个单点登录服务。对于 B/S 架构的工具,当用户登录时,工具服务器会生成若干 cookie 来标记用户的信息,如加密后的用户名、密码,以及服务器 sessionId。根据这些 cookie,服务就能够判断用户是否登录。为了让平台模拟用户登录的动作,需要在 SDModel 的 tool - access 关键字中添加工具的登录地址和登录表单用户名、密码的 ID,这里的数据还是通过 param 关键字来定义。

此外,平台和工具的地址通常不在同一个域中,为了能够让平台和工具跨域通信,将所有三方工具通过 apache 的反向代理设置在了同一个域下。

3.4 定制业务流程

当平台接入了服务后,用户能够利用这些服务定制业务流程,并使用 jBPM 工作流引擎<sup>[18]</sup>来驱动业务流程的执行。为了方便制定过程,平台提供了一个可以拖拽的可视化界面,如图 5 所示。图中的每一个元素均代表一项接入的服务或工具,在定制界面下方可设定三方服务或者三方工具提供的服务,这里的

每一项服务,都由 ESB 所提供。

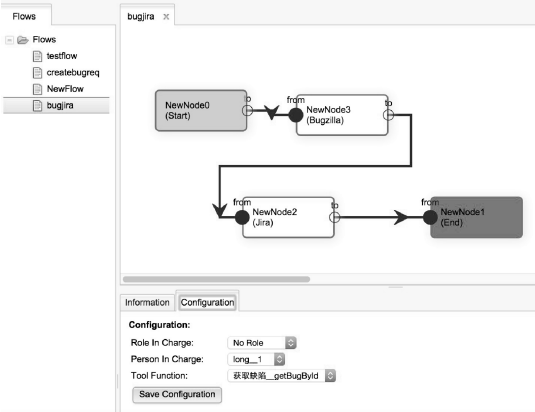


图 5 业务流程定制

完成流程的定制后,平台将验证其是否符合执行规范。如测试服务节点必须处于需求服务节点之后。而后工作流引擎调用平台的通信模块,发送 http 或其他请求和 ESB 进行通信,即可运行该工作流。

最后,平台利用自动化注册方法完成新工具、服务的接入工作,减少了集成所花费的时间,提供了热部署功能,满足了应用层定制业务流程的需求。

4 结 语

本文通过分析发现当前服务集成系统注册新工具、服务所存在的几个问题:1) ESB 配置文件较复杂对用户不友好;2) 工具、服务种类众多,开发相应的集成方法需要大量时间;3) 不支持热部署导致重新部署系统时用户无法使用。为了解决这些问题,首先定义了服务描述模型 SDModel,利用该模型,可以较为完整地描述出用户希望接入服务或工具的信息。在此基础上,设计了一个基于 SDModel 的自动化注册方法。该方法能够读取、解析 SDModel,并将其转换为对应 ESB 系统的配置文件,通过热部署的方式使其生效。由于 SDModel 能够较为通用表达服务、工具的信息,该方法具有一定的普适性。此外,方法提供了错误检测和恢复机制,保证了集成系统可以从错误状态中恢复。该方法已经在软件开发可信证据采集平台中实现,并取得了良好效果。

本文提出的自动化注册方法在面对分布式部署的 ESB 总线时,还有一定的缺陷,不能够根据 ESB 当前的负载,将服务注册到空闲的节点上。因此这是需要进一步研究的方向。

参 考 文 献

[ 1 ] Papazoglou M. Service-oriented computing: Concepts, characteristics and directions [ C ]//Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on. IEEE,2003;3 - 12.

[ 2 ] Krafzig D,Banke K,Slama D. Enterprise SOA;service-oriented architecture best practices[ M]. Prentice Hall Professional,2005.

[ 3 ] Chappell, David. Enterprise service bus [ M ]. O ' Reilly Media, Inc. 2004.

[ 4 ] Tang X,Sun S,Yuan X,et al. Automated Web Service Composition System on Enterprise Service Bus[ C ]//Secure Software Integration and Reliability Improvement,2009;9 - 13.

实例库中检索出与当前任务相关的零件实例知识。然后通过实例知识与当前知识进行相似度匹配算法分析,将相似的实例知识以列表的形式推送给用户,推送结果如图 8 所示。

序号	零件名称	加工工艺	图例	图例人
1	活木螺钉	压制 下料 加工工艺 压制 下料	广西*****公司	
2	活木螺钉	压制 下料 加工工艺 压制 下料	南宁*****公司	
3	活木螺钉	压制 下料 加工工艺 压制 下料	南宁*****公司	
4	轴瓦衬套	压制 下料 加工工艺 压制 下料	广西*****公司	
5	轴瓦衬套	压制 下料 加工工艺 压制 下料	广西*****公司	
6	轴瓦衬套	压制 下料 加工工艺 压制 下料	广西*****公司	
7	轴瓦衬套	压制 下料 加工工艺 压制 下料	广西*****公司	

图 8 关联知识推送结果

## 5 结 语

知识服务是提高知识获取能力的有效服务手段,本文结合零件知识特点,提出了基于零件实例库的关联知识服务模型。该模型是一种面向知识内容和解决实际生产问题的服务模型。研究了三种类型的关联知识检索方法,构建了关联知识检索体系,便于查找与任务相关的知识。通过文本相似度算法进行知识匹配,设计了关联知识推送方法,实现了实例库中的知识与当前知识的匹配并主动推送给用户。最后,通过应用实例说明了关联知识服务在机械非标零件设计知识服务平台中应用的可行性和实用性。通过对机械零件知识的科学管理,整合企业中各种形式的分散的知识资源,促进了知识的获取和重用,提高了设计人员的工作效率,降低了用户对知识获取的盲目性。充分利用已有的知识成果,避免或减少重复劳动,能够为用户提供高效率、个性化的知识服务。今后的工作重点是加强关联知识检索与关联知识推送的技术创新,结合企业知识需求的变化,为企业知识服务和信息化做进一步研究。

## 参 考 文 献

- [1] Mukherjee S. Information retrieval and knowledge discovery utilizing a biomedical Semantic Web [J]. Briefings in Bioinformatics, 2005, 6 (3): 252 - 262.
- [2] 刘品阳,孙滔. 知识服务平台的研究与设计——陕西省主导产业科技服务平台建设[J]. 计算机技术与发展, 2009(12): 248 - 250.
- [3] 张晓林. 走向知识服务:寻找新世纪图书情报工作的生长点[J]. 中国图书馆学报, 2000, 26(5): 32 - 37.
- [4] 熊大红. 知识服务模型和农业知识服务平台设计研究[J]. 湖南农业科学, 2009(6): 127 - 130.
- [5] 李向前,杨海成,敬石开,等. 面向集团企业云制造的知识服务建模[J]. 计算机集成制造系统, 2012, 18(8): 1869 - 1880.
- [6] Cuzzocrea A, Mastroianni C. Pushing Knowledge Management in Web Information Systems Engineering [C]//Proceedings of the International Database Engineering and Applications Symposium (IDEAS' 04), 2004: 480 - 485.
- [7] Fu C H, Trappey A J C, Hou J L, et al. Develop a Multi-Channel Legal Knowledge Service Center with Knowledge Mining Capability [J]. International Journal of Electronic Business Management, 2004, 2(2): 92 - 99.
- [8] 黄沈权,顾新建,祁国宁,等. 基于 Web 2.0 的零件库管理系统[J]. 计算机集成制造系统, 2009, 15(10): 2057 - 2062.
- [9] 王玉,刘钊. 复杂机械产品设计知识推理方法及软件系统[J]. 南

京工程学院学报:自然科学版, 2013(3): 48 - 52.

- [10] 孙长乐,宁大勇,熊伟,等. 基于特征的工程领域 CAD 模型检索技术[J]. 计算机集成制造系统, 2014, 20(4): 747 - 754.
- [11] 王家乐,姜波,黄逸民. 机械零件模型局部形状相似性检索算法研究[J]. 中国机械工程, 2012, 23(11): 1293 - 1296.
- [12] 姜永常. 论知识服务与信息服务[J]. 情报学报, 2001(5): 572 - 578.
- [13] 潘星,王君,刘鲁. 一种基于 Web 知识服务的知识管理系统架构[J]. 计算机集成制造系统, 2006, 12(8): 1293 - 1299.
- [14] 周宁,张玉峰,张李义. 信息可视化与知识检索[M]. 北京:科学出版社, 2005.
- [15] 郭忠南. 利用 TreeView 实现层次结构数据导航查询[J]. 电脑开发与应用, 2012(8): 81 - 83.
- [16] 王生发,顾新建,郭剑锋,等. 面向产品设计的知识主动推送研究[J]. 计算机集成制造系统, 2007, 13(2): 234 - 239.

## (上接第 63 页)

- [5] Li X, Tang X, Song Z, et al. AFlow: An Automated Web Services composition system based on the AI planning and workflow [C]//Progress in Informatics and Computing (PIC), 2010 IEEE International Conference on. IEEE, 2010: 1067 - 1071.
- [6] Vrba P, Fuksa M, Klima M. JADE-JBossESB gateway: Integration of multi-agent system with enterprise service bus [C]//Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on. IEEE, 2014: 3663 - 3668.
- [7] Hübner J, Stoll C. SCENE admin: A component-based integrated management framework for web service platforms [C]//IT in Medicine and Education (ITME), 2011 International Symposium on. Vol. 2, IEEE, 2011: 77 - 81.
- [8] Yu H, Ma J, Yang X. Research on Application Integration in Digital Campus Based on JBoss ESB Platform [M]. Advances in Future Computer and Control Systems. Springer Berlin Heidelberg, 2012: 319 - 324.
- [9] Tan W, Fan Y, Zhou M, et al. Data-driven service composition in enterprise SOA solutions: a Petri net approach [J]. Automation Science and Engineering, IEEE Transactions on, 2010, 7(3): 686 - 694.
- [10] Yin J, Chen H, Deng S, et al. A dependable esb framework for service integration [J]. Internet Computing, IEEE, 2009, 13(2): 26 - 34.
- [11] 王路远. 基于 ESB 的 Web 服务注册和管理的研究与实现 [D]. 北京邮电大学, 2011.
- [12] 边小凡,代艳红,马兵会. 基于 ESB 的企业内部构件与 Web 服务的集成[J]. 计算机工程与设计, 2009(4): 918 - 920.
- [13] Curbera F, Duftler M, Khalaf R, et al. Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI [J]. IEEE Internet Computing, 2002, 6(2): 86 - 93.
- [14] Weerawarana S, Curbera F, Leymann F, et al. Web services platform architecture: SOAP, WSDL, WS-policy, WS-addressing, WS-BPEL, WS-reliable messaging and more [M]. Prentice Hall PTR, 2005.
- [15] Morgan J, Venkatraman C. Embedding web access mechanism in an appliance for user interface functions including a web server and web browser [P]. U. S. Patent No. 5,956,487. 21 Sep. 1999.
- [16] Java, API for XML processing (JAXP) [OL]. [2006]. Sun Microsystems, <http://java.sun.com/webservices/jaxp/>.
- [17] Mule ESB [OL]. [2014]. MuleSoft Inc. [www.mulesoft.com](http://www.mulesoft.com).
- [18] Cumberlidge M. Business Process Management with JBoss jBPM [M]. Packt Publishing Ltd, 2007.