

企业应用集成系统 V0.0.8*

软件工程1305班 田劲锋 201316920311

2015 年 4 月 17 日

目录

1	业务需求	1	5.2	用例描述	5
2	用户需求	2		用例 5.2.1: 用户无状态登录	5
3	系统需求	2		用例 5.2.2: 用户带状态登录	6
3.1	功能需求	2		用例 5.2.3: 业务请求	7
4	需求建模	3		用例 5.2.4: 记录日志	7
5	用例模型	5		用例 5.2.5: 同步业务请求	8
5.1	用例图	5		用例 5.2.6: 异步业务请求	8
				用例 5.2.7: 加入消息队列	9
				用例 5.2.8: 监控消息队列	9
				用例 5.2.9: 新增请求类型	10
				用例 5.2.10: 业务步骤分解	10
				用例 5.2.11: 业务规则校验	10

1 业务需求

企业应用集成（Enterprise Application Integration, EAI）是一个或多个系统的集合，并使两个以上的后台应用能够作特定数据的统一管理系统，并能保证这些后台应用同步良好。这样的系统可以为单个公司提供其所有信息的集中访问点。

通过应用集成，提高业务流程的自动化程度，获得如下好处：

- 增强业务处理的可靠性
- 缩短处理时间
- 减少人工干涉
- 保证数据一致性
- 获得更高质量的报表数据
- 降低运营成本

*本文档托管在 GitHub 上，PDF 文件：<https://github.com/kingfree/haut/raw/master/se/eai.pdf>，
TeX 源文件：<https://github.com/kingfree/haut/blob/master/se/eai.tex>。

本系统将来进行可以通过插件方式进行扩展，组合成功能全面的企业资源计划（Enterprise Resource Planning, ERP）系统。

2 用户需求

对于小型中型的商业公司，通常会使用多个软件系统来处理不同的事务。用户手工操作资金结算系统、发票管理系统、销售管理系统、客户服务系统，其数据都是手工完成输入的，工作时要打开多个软件，在不同的窗口之间复制、粘贴，甚至使用电子表格来处理软件系统没有提供的功能。

用户需要一个集成平台系统，尽可能来整合这些软件系统，从而减轻工作量，减少错误发生。

系统需要支持这些功能：

- 追踪单个请求处理状态
- 保存用户请求日志
- 生成各种审计报告
- 保存系统响应日志
- 以同步或异步的方式处理请求
- 回滚处理流程机制
- 搜索条目和内容
- 事务中断处理

由于并没有具体的底层平台，该系统应开发成一个可扩展的框架（Framework），提供具体的应用程序接口（Application Programming Interface, API）而不限于图形用户界面（Graphical User Interface, GUI）。

3 系统需求

3.1 功能需求

3.1.1 【平台通用】系统处理来自各方的请求，该机制不会限制用户或其他系统，也不限制开发环境、编程语言和操作系统。

3.1.2 【调用接口】系统提供一个不依赖于环境的交互接口，使用机器和人皆可读的 XML 格式可以提供一个这样的中间平台。

3.1.3 【日志记录】要求详细记录经过系统产生的一切处理过程，包括但不限于这些信息：编号、处理情况、提交和解决时间、处理方案和步骤。

3.1.4 【日志查询】提供查询记录日志的功能，用户可以按关键字查询、排序以及按条件查询。

3.1.5 【用户验证】提供用户注册和登录功能，为每个用户产生其需要的数据和界面。

- 3.1.6 **【用户权限】** 为不同的用户分配不同的权限和角色，保障系统安全性和业务稳定性。用户权限设置要灵活，然而根用户例外。
- 3.1.7 **【用户锁定】** 对于多次执行了非法操作的用户，如多次输错密码，系统将自动锁定用户一定时间，并有管理员决策是否解除锁定。
- 3.1.8 **【集成扩展】** 系统要可以实现新增功能和业务流程的插入和修改功能，实现集成系统的扩展，以便适应用户不断变化的业务需求。
- 3.1.9 **【业务组合】** 系统将多项任务组合成单个逻辑的用户请求，通过把若干相关的处理过程整合成为一个请求，用户可以将一系列相关操作组合成逻辑上的统一整体，并包含在单个请求中进行处理。
- 3.1.10 **【输入验证】** 系统要验证数据输入的合法性，这是保证系统能够正常运行的必要条件。通过操作进行之前的输入验证和修正，可以预防后期修正导致的一连串棘手问题。
- 3.1.11 **【业务处理】** 需要在请求处理过程中落实业务规则，验证用户通用的要求和自定义要求，根据不同情况处理状态和数据库，在此基础上制作数据报表和性能报表。
- 3.1.12 **【请求回滚】** 对于包含多个步骤的处理请求，有些可以容忍的错误可以记录在日志中，有些错误不可容忍，需要中止整个请求的处理，甚至需要恢复请求之前的数据系统状态。
- 3.1.13 **【管理报告】** 日志查询的升级版，提供图表形式的日志数据。通过对日志数据的分析，系统生成不同种类的报告提供给系统管理员分析和检查问题。
- 3.1.14 **【实时警报】** 对于严重的系统错误，影响到了用户利益，应当提供途径向相关人员发送警报信息，以便进一步处理。方式包括但不限于电子邮件、手机短信、即时通讯等。
- 3.1.15 **【灾难恢复】** 当系统运转过程中产生了不可避免的意外事故，如断电等，将导致处理结果不完整。有必要在分解请求的基础上，及时保存请求处理到数据库中，同时提供措施以便恢复用户提交而未能及时处理的业务。

4 需求建模

图 1 描述了系统最顶层的数据流。Web 服务器从用户处接收业务处理请求的消息，在记录如数据库的同时将组合业务逻辑转发给 EAI 后台，后台将业务分解到各个子系统中，并记录数据库。

顶级数据流包含两个主要的处理逻辑，这里将其描述为一级数据流程图。

图 2 描述了用户请求的数据流程，也就是 Web 服务器的处理流程。

图 3 描述了 EAI 系统的后端处理的数据流程。

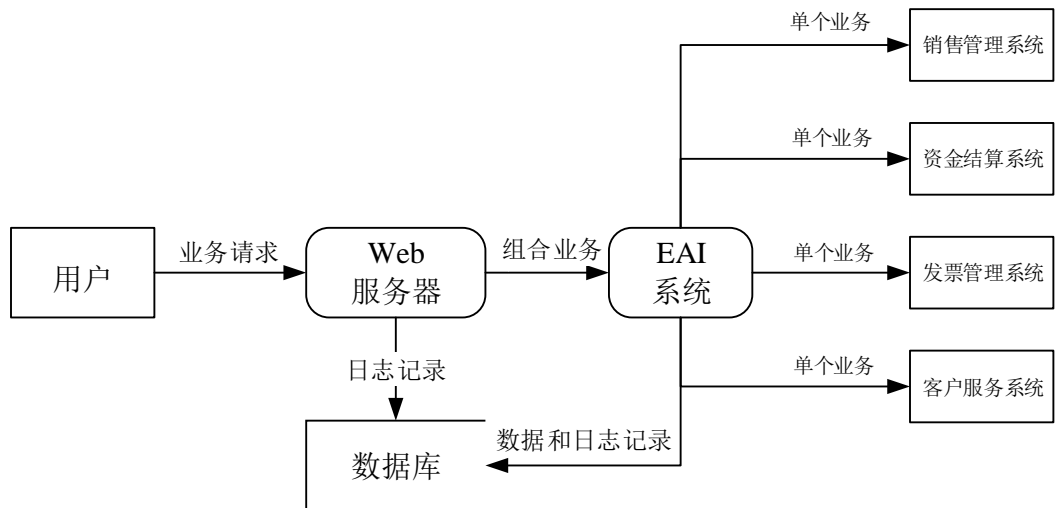


图 1: 系统顶级数据流图

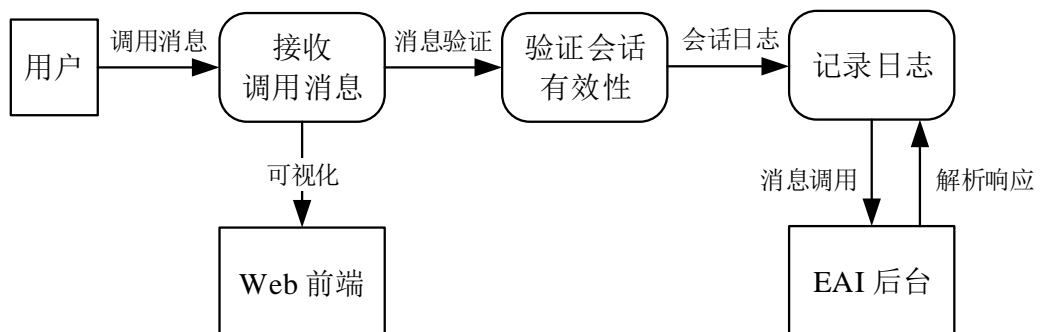


图 2: 一级数据流图：用户请求的处理

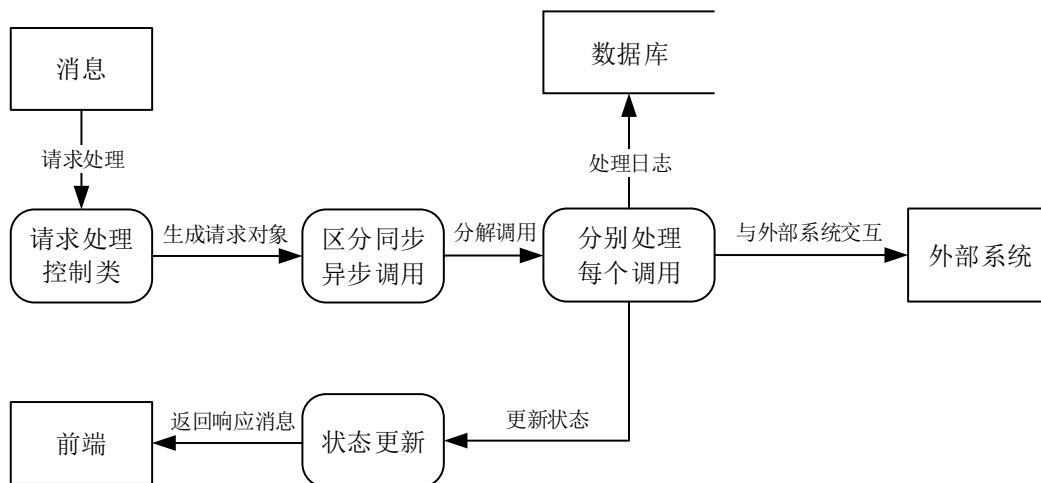


图 3: 一级数据流图：EAI 系统的后端处理

5 用例模型

5.1 用例图

图 4 是该系统的用例图，用户只有两个基本用例，登录和业务请求。注意这里我们将业务请求合并为了一个用例，因为实际上我们是可以用一个类来处理这项业务的。

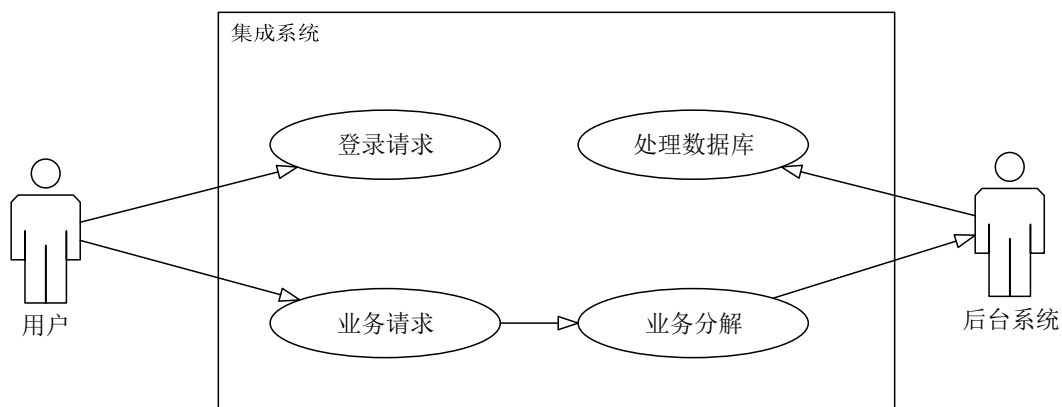


图 4: 系统用例图

5.2 用例描述

用例 5.2.1: 用户无状态登录

1. 目标

该用例允许用户登录，验证用户身份，并将用户状态保存在会话中。

2. 事件流

2.1 基本流程

当用户进入该系统登录页面，该用例开始执行。

- (1) 用户输入用户名和密码，按下 **登录** 按钮；
- (2) 系统将密码加密后，检查用户名存在且加密密码一致；
- (3) 生成新的会话编号并将其和当前时间记入数据库；
- (4) 返回并设置会话表示登录成功。

2.2 可选流程

(1) 用户不存在

在主流程中，如果系统中没有与输入用户名匹配的用户，返回错误消息，用例结束。

(2) 密码错误

在主流程中，加密后的密码与对应系统中用户的加密密码不一致，返回错误消息，用例结束。

3. 特殊需求

在输入过程中，用户选择了 下次自动登录 复选框，系统需要在登录验证成功后，将会话写入状态。

4. 前提条件

用户没有登录，且没有登录状态。

5. 后置条件

用户通过登录验证，其状态保存在会话中。

用例 5.2.2：用户带状态登录

1. 目标

该用例允许用户不输入用户名和密码直接登陆系统。

2. 事件流

2.1 基本流程

当用户启动该系统，进入该系统任意页面，该用例开始执行。

- (1) 检查用户是否有状态信息；
- (2) 检查用户状态，在数据库中找到对应的会话编号；
- (3) 返回并设置会话表示登录成功。

2.2 可选流程

(1) 没有状态

用户没有状态信息，跳转到用例 5.2.1：用户无状态登录，本用例结束。

(2) 无此会话

没有在数据库中找到对应的会话编号，返回错误消息，用例结束。

3. 特殊需求

无。

4. 前提条件

用户进入该页面时已经带有状态。

5. 后置条件

用户通过登录验证，其状态保存在会话中。

用例 5.2.3：业务请求

1. 目标

用户提交一个业务请求，系统分解处理该请求并返回给用户。

2. 事件流

2.1 基本流程

当用户进入指定页面填写业务表单后提交了该请求时，该用例开始执行。

- (1) 将用户的表单内容转换成中间格式 XML；
- (2) 系统解析 XML 语义生成请求对象；
- (3) 系统确定该业务是同步处理业务还是异步处理业务；
- (4) 将异步请求对象加入处理队列，为同步业务新建处理对象；
- (5) 系统处理该对象，调用用例 5.2.5：同步业务请求或用例 5.2.6：异步业务请求；
- (6) 根据处理结果，返回响应对象。

2.2 可选流程

(1) 解析错误

用户的输入存在错误，无法正确解析成 XML，返回出错消息，用例结束。

3. 特殊需求

无。

4. 前提条件

用户已登录且拥有处理该业务的权限。

5. 后置条件

业务处理并返回状态，记录日志。

用例 5.2.4：记录日志

1. 目标

对于用户操作和系统的处理等动作进行记录。

2. 事件流

2.1 基本流程

系统产生一个动作时进入该用例。

- (1) 系统获得当前动作的标记信息；
- (2) 系统区分该动作的日志级别；
- (3) 产生唯一日志编号，根据级别写入日志到数据库；
- (4) 系统读取并解析输出日志的参数；
- (5) 向指定位置输出日志消息和时间。

2.2 可选流程

- (1) 日志不可写

日志配置中指定的位置不可写入，产生警告并写入默认日志位置。

3. 特殊需求

无。

4. 前提条件

无。

5. 后置条件

日志写出成功，转入正常业务流程。

用例 5.2.5：同步业务请求

1. 目标

以同步的方式处理业务请求。

2. 事件流

2.1 基本流程

- (1) 检查预处理业务规则；
- (2) 解析出调用请求；
- (3) 调用对应的请求处理对象；
- (4) 检查请求类型的业务规则；
- (5) 请求处理对象执行必要的处理步骤；
- (6) 处理步骤记录必要的日志信息；
- (7) 处理业务后规则；
- (8) 生成并返回响应对象。

2.2 可选流程

(1) 处理步骤出错

处理请求处理对象时出现错误，在系统允许的情况下结束该业务，记录业务状态；需要回滚的业务要调用回滚处理用例。

3. 特殊需求

无。

4. 前提条件

系统接收到同步业务处理请求。

5. 后置条件

同步业务请求处理完成。

用例 5.2.6：异步业务请求

1. 目标

以异步的方式处理业务请求。

2. 事件流

2.1 基本流程

- (1) 将调用请求压入消息队列；
- (2) 记录必要的日志信息；
- (3) 生成并返回响应对象。

2.2 可选流程

- (1) 无。
- 3. 特殊需求
无。
- 4. 前提条件
系统接收到异步业务处理请求。
- 5. 后置条件
异步业务请求成功加入队列。

用例 5.2.7：加入消息队列

- 1. 目标
将异步处理业务请求加入消息队列
- 2. 事件流
 - 2.1 基本流程
 - (1) 将用户请求打包为格式消息；
 - (2) 将消息加入队列，消息队列接收消息并存储下来。
 - 2.2 可选流程
 - (1) 无。
- 3. 特殊需求
无。
- 4. 前提条件
异步业务已解析成为处理对象。
- 5. 后置条件
业务加入消息队列，等待处理。

用例 5.2.8：监控消息队列

- 1. 目标
监控程序监控消息队列并按顺序处理其中的请求对象。
- 2. 事件流
 - 2.1 基本流程
 - (1) 按照先进先出（First In First Out, FIFO）的顺序从队列中读取一个消息；
 - (2) 处理消息；
 - (3) 报告处理情况，记录日志和数据库。
 - 2.2 可选流程
 - (1) 队列为空
消息队列中没有消息请求，说明当前没有要处理的异步请求，用例结束。
- 3. 特殊需求
无。
- 4. 前提条件
消息队列已启动
- 5. 后置条件
继续处理下一个消息。

用例 5.2.9：新增请求类型

1. 目标

提供一个添加新的业务流程的通用接口。

2. 事件流

2.1 基本流程

- (1) 确定请求消息中的请求名称；
- (2) 通过继承公共请求类来新建处理类：
 - A. 为新的请求处理类新建一个类，声明其父类；
 - B. 重写处理方法；
 - C. 重写内部处理方法；
 - D. 重写回滚方法；
 - E. 编译测试新的处理类。
- (3) 在数据库中插入相应记录，标记此类可用。

2.2 可选流程

- (1) 新建类名冲突
用户要求新建的类名与已有类名冲突，系统提示更改名称以消除冲突；
- (2) 类编译测试失败
新建类编译测试失败，标记此类不可用，直至调试成功。

3. 特殊需求

提供调试功能。

4. 前提条件

系统已提供公共处理类。

5. 后置条件

请求类建立完成并可用。

用例 5.2.10：业务步骤分解

1. 目标

2. 事件流

2.1 基本流程

- (1) ttt

2.2 可选流程

- (1) sss

3. 特殊需求

4. 前提条件

5. 后置条件

用例 5.2.11：业务规则校验

1. 目标

2. 事件流

2.1 基本流程

- (1) ttt

2.2 可选流程

(1) sss

3. 特殊需求
4. 前提条件
5. 后置条件