

# SAP BDT 业务数据工具集的开发原理及应用实例

朱羽冰

(中国惠普有限公司)

**摘要:** 介绍 SAP BDT (业务数据工具集) 这一集成的封装式开发模型的原理和开发方法。该模型是在 SAP CRM 的集成业务伙伴的实现中研制出来的, 在国内 SAP 实施项目中才刚刚开始被使用。给出的实际案例描述了高效深入地使用 BDT 技术的具体步骤, 集成了 BDT 和自己的程序。为 SAP 的用户和实施伙伴进行客户和行业定制化开发提供指导作用。

**关键词:** 业务数据工具集; 业务伙伴; 简易增强工作台

**中图分类号:** TP392 **文献标识码:** B

## Development principle and application of SAP BDT

ZHU Yu Bing

(China Hewlett-Packard Co., Ltd)

**Abstract:** The paper introduces the concept and theory of SAP BDT (Business Data Toolset) which is originated in the central business partner project in several years. The given practical example describes the steps of BDT development in the case of complicated project. It will offer SAP users specific requirements and industrial solution design.

**Key words:** BDT; BUPA; EEWB

业务数据工具集 BDT(Business Data Toolset)是一套集成的封装式的开发模型, 主要用于维护复杂的主数据或简单的事务型数据。BDT除了包含对话框的操作, 同时还支持直接数据录入和函数操作。BDT集成了SAP的其他技术功能, 如变更管理、基于字段组的输出控制、权限控制、存档功能<sup>[1]</sup>。BDT还集成了与BW的接口程序, BDOC类型的接口程序、XIF类型的接口函数。BDT具有一整套的配置工具和核心控制程序, 然后配合具有规范接口的基于事件的函数集合, 允许快速规范开发自己的整套应用或者对现有的应用进行不同层次的配置和增强, 这些增强不会涉及直接更改系统核心源代码, 因而可以高效地利用这套模型的强大功能。

BDT是在SAP CRM产品的集成业务伙伴对象和业务伙伴关系对象的实现过程中研制出来的, 并完善了VCT和EEWB两个工具。可视化配置工具VCT(Visual Configuration Tool)实现了类似所见及所得的屏幕配置。因为该工具比较简单也有一定的局限性, 本文不展开。简易增强工作台EEWB(Easy Enhancement Workbench)有增加新字段

和创建新透明表两种模式。通过一个向导, 在回答完所有问题后, 系统会自动产生一系列的程序代码。可以对产生的程序做一些变更和完善, 以满足个性化需要。

BDT代表了SAP规范化开发的一个方向。BDT成型后, 在SAP内部被一些行业解决方案和主要合作伙伴积极采用, 如银行业、保险业、房地产业、校园管理等领域的解决方案, 所以应用BDT的机会越来越多了。但是目前国内的用户还很少, 并且大多满足于使用其标准功能, 而敢于采用BDT开发整套应用的还没有。

BDT的整套封装式模型包含大量的配置(IMG)、用户出口、程序、函数和工具。本文以核心概念和应用实例为主, 具体的细节可以参照SAP手册<sup>[2]</sup>。

### 1 基本概念

#### 1.1 基本概念

一个主数据由一个关键字来惟一地识别它。但是还有一些其他的标准来区分和约束它, 这就是差异类型(Differentiation)。差异类型可以理解为一个组织架构或者其他决定因素。它在创建该主数据的初始屏幕

时就要确定, 决定了后续的维护界面和内容<sup>[2-3]</sup>。

应用对象是一类数据的抽象名, 如业务伙伴 (BUPA)、业务伙伴关系 (BUPR)、银行账号 (BKKA)、合同账号 (FICA)。它被分配给差异类型。有些具体对象允许有多个差异类型实例, 如业务伙伴有多个角色; 有些具体对象只允许有一个差异类型实例。应用对象的操作模式如创建、修改、显示等以及其他的控制配置均和应用对象联系起来。需要定义应用对象关系之间允许的差异类型和缺省的差异类型<sup>[4]</sup>。

应用是指某应用对象的一种具体的操作和功能。

## 1.2 交互控制

### 1.2.1 数据结构

BDT的主数据可以保存在很多个有相似关键字的透明表 (数据库表) 中。每个表都有相对应的读取和保存函数。同时有一批数据结构用于接口定义和屏幕输出。通过数据字典、检查表和搜索帮助可以方便地实现在线输入帮助和检查。

### 1.2.2 屏幕控制

屏幕显示的基本要素是视图 (View), 实际上就是子屏幕。通过调用函数 BUS\_PBO 和 BUS\_PA1 实现 PBO 和 PA1, 并规范 PBO 和 PA1 的程序。主要的流程控制、屏幕显示以及各种检查和缺省都在视图的程序中。通常一个视图内的字段是一个字段组, 通过基于字段组的输出



图1 视图结构

控制配置功能实现基于不同条件下的屏幕输出控制。

几个视图组成一个块 (Section), 块内也可以有流程控制。几个视图组成一个屏幕, 如图1所示。

屏幕次序定义了屏幕出现的次序, 如图2所示。屏幕次序可以分成主屏幕次序和辅助屏幕次序。屏幕次序可以分给不同种类的对象, 如不同角色的业务伙伴。

每个屏幕和块都有标题, 每个屏幕也对应一个功能码。BDT有标准的菜单, 用户也可以增加功能码到菜单中。

### 1.2.3 程序逻辑

BDT的交互控制程序定义了大量的事件函数, 流程图如图3所示, 每一个事件对应一个函数, 开发人员可以通过更改这些函数来控制交互过程。这些函数有明确的命名规范, 为<应用>\_<应用对象>\_EVENT\_<事件名>。这些函数被BDT的核心程序调用。

### 1.2.4 其他

数据分割是BDT的一个重要概念。每一个应用对象都包含很多可以维护的数据, 但并不是每一个主数据都允许显示和维护所有的字段。数据分割就是进行这样的区分。业务伙伴的角色就是数据分割, 不同的角色有不同的主数据, 同一个对象允许是多个角色。数据分割的方法是由应用对象决定的。

BDT中数据的保存并不是以屏幕为单位的, 而是把数据保存在全局变量中, 最后在DSAVE事件中保存。这样的机制甚至允许多个有关系的应用对象按一定的次序共同创建或共同取消<sup>[5]</sup>。

## 2 客户项目中BDT的应用实例

某客需要对于业务伙伴的定义进行增强。首先创建角色最终用户, 并进行字段增强。业务伙伴字段的增强采用EEWB的工具产生BDT的各种程序和函数, 接着做一些简单的修改, 然后通过业务伙伴的配置把它们合理地显示在屏幕上。

### 2.1 运行EEWB和后续调整

运行事务代码EEWB, 进入简易增强工作台。创建针对业务伙伴的增强项目ZFWW\_BUPA\_01。针对定义的每一个视图创建一个子对象, 运行一次向导。每一个视

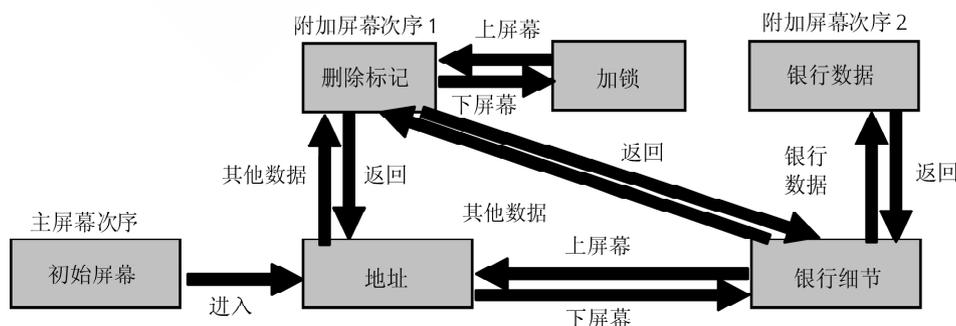
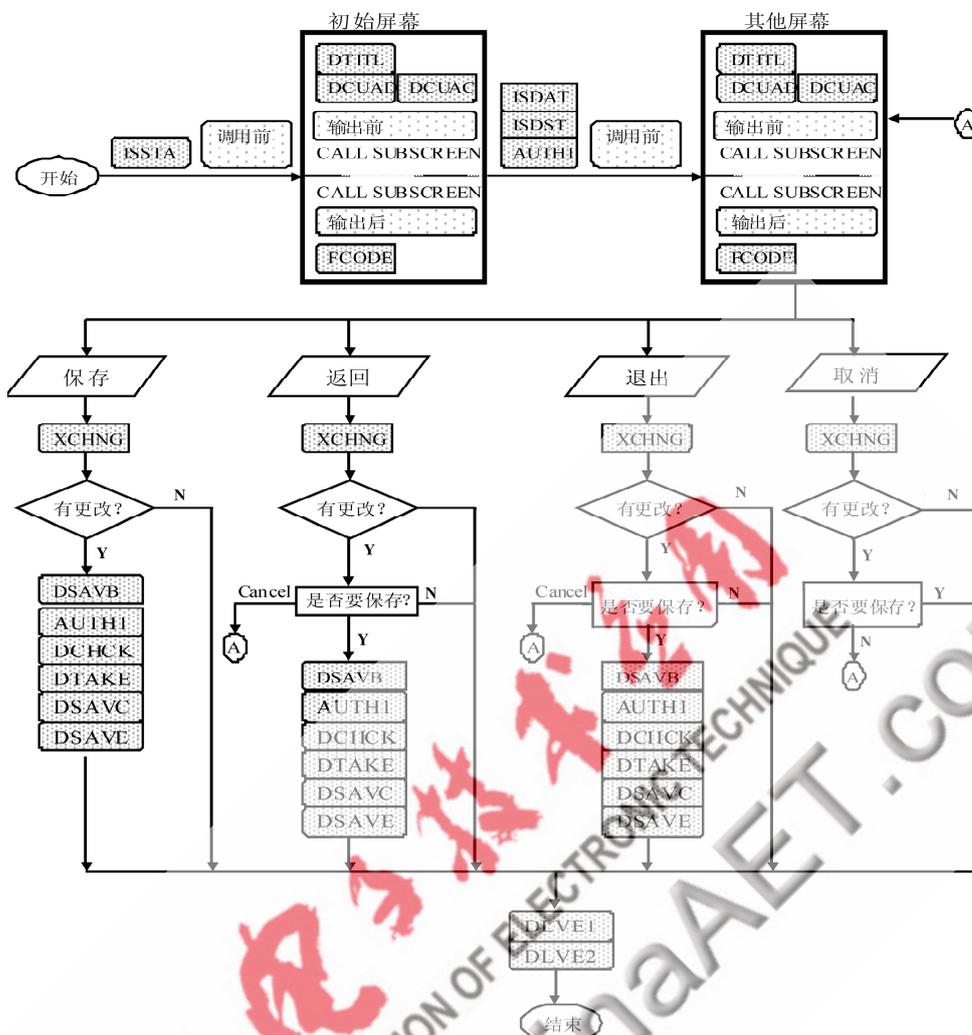


图2 屏幕次序



屏幕程序中的事件流

图3 流程图

ISSTA: 初始化;

ISDAT: 由主应用从数据库读取数据;

ISDST: 分配数据到其他使用该数据的应用中

DTITL: 更改屏幕标题

DCUAD: 决定菜单

DCUAC: 更改菜单中功能码的状态等

AUTH1: 权限控制

FCODE: 触发自定义的功能码后的操作

XCHNG: 检查屏幕输入字段是否改变

DSAVB: 收集本应用的数据

DCHCK: 数据有效性检查

DTAKE: 数据汇合后保存到内存

DSAVC: 完成待保存入数据库的数据

DSAVE: 数据库保存操作

DLVE1: 初始化本地内存

DLVE2: 初始化全局内存

图对应一个字段集合和权限对象等, 分得细有利于控制。至于屏幕的显示可以通过配置 (IMG) 来调整。

在本案例中, 针对单选项或简单输入项, 采用对表 BUT000 添加字段的方法。举例客户付款方式。在向导中定义字段名、字段类型、检查表名、抬头名、不需要有效期控制、需要 BW 集成, 完成后系统会自动产生数据字典、数据结构、检查表、子屏幕(视图)、子屏幕的 PBO 和 PAI、针对于事件 DLVE1、DSAVB、ISDST、ISSTA and

XCHNG 的函数、直接操作的 BAPI 函数、现有 BDOC 自动扩展、在 BW 的数据源 OBPARTNER 增加字段。产生的程序属于自开发程序。

EEWB 结束后, 首先激活检查表的维护, 然后使用事务代码 SM30 维护可选项。例如维护客户付款方式: 00—其他; 01—分期付款; 02—全款; 03—租赁购买; 这样就形成了如图 4 的单选项。针对复杂的运用, 可能需要修改和创建针对特定事件的函数, 如扩展性检查等。



图4 客户付款方式选择界面

在本案例中希望联系方式和方便联系时间是多选项，所以采用创建新透明表的方法。在向导中定义字段名、字段类型、检查表名、抬头名、不需要有效期、需要BW集成、需要BDOC集成，完成后系统会自动产生数据字典、数据结构、检查表、子屏幕（视图）、子屏幕的PBO和PAI、针对于事件DLVE1、DLVE2、DCHK、CHGD1、CHGD4、DSAVC、DSAVE、DTAKE、FCODE、ISDAT、ISSTA和XCHNG的函数，直接操作的BAPI函数、BDOC接口函数，归档函数、变更记录管理函数、在BW的数据源OBPARTNER增加字段。

系统产生的希望联系方式和方便联系时间只有一列，没有文本显示，通常需要修改系统产生的程序以改进显示效果。以希望联系方式为例说明如下。

首先在产生的屏幕结构 gty\_dynpstruc 中增加字段 TEXT like ZTB0000YQJXCOT-text。

#### 修改视图 屏幕顺序 -> 屏幕: 总览



图5 屏幕配置

在产生的函数组的屏幕 0100 中增加输出元素: ZTB0000YQJXCOT-text。这样就调整好了子屏幕。接着在 PAI 中, 当用户选择了希望访问方式后读取文本。

```
* 希望访问方式描述 BEGIN-----*
```

```
SELECT SINGLE TEXT
```

```
INTO GT_ZBUT0000C7J51V_D-TEXT
```

```
FROM ZTB0000YQJXCOT
```

```
WHERE ZZTFLDRUJ81V = GT_ZBUT0000C7J51V_D-
```

```
ZZTFLDRUJ81V .
```

```
* 希望访问方式描述 END -----*
```

最后在函数 ZZO6J\_BUPA\_PBO\_ZO6J0100 中添加类例程序。

对于复杂的应用, 需要修改和创建针对特定事件的函数, 并且调整会比较多, 例如和其他数据库表关联后读取数据和保存数据。如果要使用 BW 和 BDOC, 也需要根据 SAP 的手册做一些配置和激活<sup>[4]</sup>。

#### 2.2 区域菜单 BUPT 内的配置

区域菜单 BUPT 包含了所有 BDT 业务对象中业务伙伴和业务伙伴关系的配置, 所以本文的路径都在区域菜单 BUPT 内。本案例的配置不牵涉到创建新的应用对象、应用和分割, 但是所有配置菜单都在 BUPT 内。首先是配置屏幕的输出效果、参考和复制屏幕次序 BUP001 及其屏幕, 创建自己的屏幕次序, 如图 5 所示,

其中包含系统标准的屏幕和自定义及修改后的屏幕。SAP 菜单路径为业务合作伙伴→控制→屏幕布局→部分, 业务合作伙伴→控制→屏幕布局→屏幕, 业务合作伙伴→控制→屏幕顺序。

定义业务伙伴视图: 最终用户。其中定义了数据集和屏幕次序。SAP 菜单路径为业务合作伙伴→控制→可分割→业务伙伴视图。

定义角色的基于字段组的输出控制。SAP 菜单路径为 IMG→交叉应用组件→SAP 业务伙伴→业务合作伙伴→基本设置→字段分组→配置每个商业伙伴角色的字段属性。

定义角色最终用户的专用事务代码。这样可以提高最终用户的创建、修改、显示的速度。SAP 菜单路径为业务合作伙伴→控制→应用程序事务。

### 3 校园管理解决方案中 BDT 应用实例

在校园管理的解决方案中, 学生主数据也是一种业务伙伴。采取案例 1 中的方法产生和配置好业务伙伴学生后, 根据行业特点做一些较深入的开发。

#### 3.1 更改初始化屏幕

业务伙伴创建时需要指定角色、种类、组、类型<sup>[4]</sup>。但是对于学生主数据来说, 这些概念有一定的扩展和转换。需要根据其他的一些要素来自动决定上述关键信息。于是需要定义一些配置表, 开发函数 PMIQ\_BUPA\_CREATE\_STUDENT\_BUF 在 BDT 运行开始时被调用。屏幕次序中的初始化屏幕也需要改变。该函数的定义如下:

输入条件: 学生的 ID, 操作类型 (创建、修改、显示), 有效期。

程序运行的结果是 BDT 的实例被创建, 开始进入后续界面。

程序流程:

\* BDT\_CLASS\_INIT 初始化 BDT 的分类

\* BDT\_INSTANCE\_CREATE 创建 BDT 的实例

\* BDT\_INSTANCE\_ATTRIBUTES\_SET 定义操作类型和显示控制参数

\* BDT\_HEADER\_CHECK 根据自定义的配置表和其他约束条件, 推导出业务伙伴的角色、种类、组、类型

\* BUS\_VALDT\_SET 进行有效期控制。创建时一定是当前日期, 修改时可以定义当前日期或未来的日期, 显示时任何日期都允许

\* BDT\_DATA\_READ 读出相关的数据。在创建时作为缺省数据。在修改和显示时, 是系统内一致的数据。

#### 3.2 创建自己的辅助屏幕次序

需要根据一批已经创建完成的屏幕组成一个辅助屏幕次序, 这些屏幕采用制表页的形式出现, 并增加了

一些自己的逻辑, 所以采用自编程序调用 BDT 的程序的方法。函数 PMIQ\_BUPA\_BDT\_SCREEN\_NAVIGATE 用来控制屏幕的切换。这个函数调用了函数 BDT\_FUNCTION\_NAVIGATE: 它指定了屏幕号、屏幕类型、光标位置。BDT\_SCREEN\_ATTRIBUTES\_GET: 使用它激活了一些按钮等。这个函数用于屏幕的 PBO。

创建函数 PMIQ\_BUPA\_BDT\_SCREEN\_PBO 来控制 PBO, 该函数调用了函数 BDT\_SCREEN\_PBO\_PROCESS。

创建函数 PMIQ\_BUPA\_BDT\_SCREEN\_SEL\_PBO, 在 PBO 中它调用函数 BDT\_SCREEN\_SELECT\_PBO 得到 BDT 子屏幕号, 然后再调用函数组 BUSS 中对应屏幕的 PBO 函数。

创建函数 PMIQ\_BUPA\_OK\_CODE\_SET, 由这个函数把 OK\_CODE 传递给 BDT。

创建函数 PMIQ\_BUPA\_BDT\_SCREEN\_SEL\_PA1, 在 PA1 中调用函数 BDT\_SCREEN\_SELECT\_PA1 得到 BDT 子屏幕号, 然后再调用函数组 BUSS 中对应屏幕的 PA1 函数。

#### 3.3 函数 PMIQ\_BUPA\_BDT\_SUBSCREEN\_PA1

这是主要的 PA1 函数。它首先得到了光标的位置, 处理完 BDT 的 PA1 函数后处理其他的功能码。用户退出 BDT 时, 停止 BDT 实例。

#### 3.4 主数据保存

因为在学生主数据保存的同时, 需要创建学生财务账号。所以在 BDT 事件 DSAVB、DSAVC、DSAVE 完成后, 再调用一个函数去创建学生财务账号。

BDT 是一个封装式的应用开发模型。它包含大量的配置 (IMG)、用户出口、程序、函数和工具, 并且单纯的开发可能看不到效果或者对其他逻辑产生影响, BDT 确实是一个有效的、复杂的工具。从 SAP 正在开发和完善的行业解决方案中大量采用 BDT 可见一斑。有些企业和 SAP 合作伙伴采用 SAP 系统平台开发自己特有的局部解决方案, BDT 是一个可以考虑的工具。对于其他应用开发平台供应商而言, 它也是一个可以借鉴的开发模型。

#### 参考文献

- [1] 托马斯·H·达文波特, ERP 必备指南.北京: 北京机械工业出版社,2003.
- [2] SAP AG. 在线帮助 CRM5.0 BDT 部分. CRM5.0 Online help BDT part.
- [3] SAP AG. CRM 培训教材 CR590[A]. CRM training material CR590 BDT.
- [4] 关宇平. SAP R3 系统客户接口程序的开发方法及实现[J]. 计算机系统应用, 2007, 13(2):47-50.
- [5] 施权. 利用 SAP ABAP 语言解决客户特制要求的方法[J]. 计算机工程, 2004, 30(S1):84-86.

(收稿日期: 2008-12-15)