Enhancing the CRM Middleware Example

Introduction

Business Partners in ERP have a field called "Customer classification (ABC analysis)" on their Sales Data. This field does not exist in CRM hence is not replicated between the two systems. In the following guide I explain how I added the ABC-classification to CRM and enhanced the CRM Middleware in order to replicate the field between the two systems.

From ERP to CRM

In the following chapter you'll find the developments done in the ERP and CRM system in order to send the ABC-classification of a customer from ERP to CRM.

General information

In transaction xd02/3 (Customer change/display) is the ABC-classification found under the Sales Organization Data and there under Pricing. The corresponding table is called KNVV and the field KLABC. The key consists of the customer number (KUNNR), sales organization (VKORG), distribution channel (VTWEG) and division (SPART).

Enhancement Steps in ERP

Customizing of User Exit

The CRM Middleware can be enhanced using User Exits which are triggered by certain Interface Processes. These processes are found in table TPS01 whereas the ones starting with "CRM" concern the middleware.

First however a so called Customer Product is needed which later on will be assigned to the user exit. A customer product is customized via sm30 of table TBE24. It consists of a product name, description, optionally of a RFC destination (if the called user exit is not in the same system) and an activation flag.

<table>
<thead>
<tr>
<th>Product</th>
<th>ZBPPRIE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>BUPA Pricing Extension</td>
</tr>
<tr>
<td>RFC-destination</td>
<td></td>
</tr>
<tr>
<td>Active</td>
<td>X</td>
</tr>
</tbody>
</table>

Now a User Exit can be customized via sm30 of table TPS34. The process "user-exit before data is sent" is the one to use (CRM0_200). A new function module needs to be created and the newly added customer product needs to be assigned (through its activation flag is the user-exit called or not).

<table>
<thead>
<tr>
<th>Process</th>
<th>CRM0_200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country</td>
<td></td>
</tr>
<tr>
<td>Appl.</td>
<td></td>
</tr>
<tr>
<td>Function module</td>
<td>Z_CRM_MW_BP_PRICING_OUT</td>
</tr>
<tr>
<td>Product</td>
<td>ZBPPRIE</td>
</tr>
</tbody>
</table>
Function for User Exit

Table **TPS01** contains sample function modules for all processes with the corresponding parameter interface. It is best to copy the sample function of the process chosen; in this case **SAMPLE_PROCESS_CRM0_200**.

The TABLES-parameter **T_BAPISTRUCT** contains all data (tables, structures and fields) which will be sent to CRM. The standard structures cannot have append structures therefore enhancements must be appended to that table as if they were a table themselves (at least is this the only way I could find).

**Function module Z_CRM_MW_BP_PRICING_OUT**

This is how I coded the adding of the ABP-classification to the CRM Middleware (the comments in the code should be explainable enough).

```
* Data definitions --------------------------------------------------------------*
TYPES: ty_knvv_t LIKE knvv OCCURS 0.
DATA: lt_complex TYPE busei_comExtern_t,
      ls_complex TYPE busei_comExtern,
      lt_not_processed TYPE busei_bapimtcs_t,
      ls_partner TYPE busei_comInstance,
      lv_customer LIKE kna1-kunnr,
      ls_salesarea TYPE busei_ccom_salesarea,
      lv_vkorg TYPE vkorg,
      lv_knvv_tab TYPE string,
BEGIN OF ls_mtcs_lite,
  currdno TYPE currdno,
  objkey TYPE crmobjkey,
  relkey TYPE crmrelkey,
END OF ls_mtcs_lite.
FIELD-SYMBOLS: <fs_knvv_t> TYPE ty_knvv_t,
                <fs_knvv> TYPE knvv.

* get current pricing table - KNVV
  lv_knvv_tab = '(SAPLCCCO)GT_KNVV[]'.
ASSIGN (lv_knvv_tab) TO <fs_knvv_t>.
CHECK sy-subrc = 0.

* map BAPIMTCS table to complex structure table (only the partners one)
CALL FUNCTION 'PI_BP_BAPIMTCS_TO_MAIN'
  EXPORTING
    it_bapimtcs = t_bapistuct[]
  IMPORTING
    et_bpExtern = lt_complex
    et_not_processed_bapimtcs = lt_not_processed.
LOOP AT lt_complex INTO ls_complex.
  CLEAR: ls_partner, lv_customer.

* set customer
  ls_partner = ls_complex-header-object_instance.
```
CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
  EXPORTING
    input = ls_partner-bpartner
  IMPORTING
    output = lv_customer.

* go through sales areas of customer
LOOP AT ls_complex-crm_data-salesarea INTO ls_salesarea.
  CLEAR: lv_vkorg, ls_mtcs_lite.

* set sales organization
  lv_vkorg = ls_salesarea-salesarea-sales_org(4).

* get current pricing data (ABC-classification)
  READ TABLE <fs_knvv_t> ASSIGNING <fs_knvv>
    WITH KEY kunnr = lv_customer
    vkorg = lv_vkorg
    vtweg = ls_salesarea-salesarea-channel
    spart = ls_salesarea-salesarea-division.
  CHECK sy-subrc = 0.

* add ABC-classification into BAPIMTCS
  DESCRIBE TABLE t_bapistruct[] LINES sy-tfill.
  READ TABLE t_bapistruct INDEX sy-tfill.
  ls_mtcs_lite-currdno = t_bapistruct-currdno + 1.
  ls_mtcs_lite-objkey = t_bapistruct-objkey.
  ls_mtcs_lite-relkey = t_bapistruct-relkey.

  CLEAR t_bapistruct.
  MOVE-CORRESPONDING ls_mtcs_lite TO t_bapistruct.
  t_bapistruct-tabname = 'ZMW_PRICING_EXT_S'.
  CONCATENATE lv_customer ls_salesarea-salesarea <fs_knvv>-klabc
    INTO t_bapistruct-data.
  APPEND t_bapistruct.
Enhancement Steps in CRM

Finding the right place for the "new" field

The ABC-classification field does not exist in CRM and must be created first. It should be together with the other customer sales - pricing (billing) data. On the WebUI would that be following:
Account -> Sales Area Data -> Billing

Technically are the pricing data in the Business Partner Set 0030:

<table>
<thead>
<tr>
<th>What</th>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link between Business Partner and Set 0030</td>
<td>Table</td>
<td>CRMM_BUT_LNK0031</td>
<td>Business Partner - Billing Rule</td>
</tr>
<tr>
<td>Data of Set 0030</td>
<td>Table</td>
<td>CRMM_BUT_SET0030</td>
<td>Business Partner: Billing Data</td>
</tr>
<tr>
<td>Data Structure of Set 0030</td>
<td>Structure</td>
<td>CRMT_BUS_SET0030</td>
<td>CRM BP Set Structure for Billing Data</td>
</tr>
</tbody>
</table>

The right place for the ABC-classification field is therefore an append structure of structure CRMT_BUS_SET0030.

Create "new" field

I created an append structure called ZMW_BP_PRICING_EXT_S and added it to CRMT_BUS_SET0030.

<table>
<thead>
<tr>
<th>Component</th>
<th>Component Type</th>
<th>Data Type</th>
<th>Length</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZZKLABC</td>
<td>KLABC</td>
<td>CHAR</td>
<td>2</td>
<td>Customer Classification (ABC analysis)</td>
</tr>
</tbody>
</table>

Customizing of User Exit

Again we're enhancing the CRM Middleware via User Exits. These exits are also triggered by processes which can be found in table TPS01. Unlike in ERP did I find that the processes starting with 'CRM' are not the ones to use but instead those starting with 'SMOUTIL'.

But first a Customer Product is needed - just like in ERP. It is also customized via sm30 of table TBE24.

<table>
<thead>
<tr>
<th>Product</th>
<th>ZBPPRIE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>BUPA pricing extension</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RFC-destination</th>
<th>Active</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

The actual User Exit is customized via sm30 of table TPS34. The process "user-exit for upload" is the one to use (SMOUTIL3). A new function module needs to be created and the newly added customer product needs to be assigned (through its activation flag is the user-exit called or not).

<table>
<thead>
<tr>
<th>Process</th>
<th>SMOUTIL3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country</td>
<td></td>
</tr>
</tbody>
</table>
Function for User Exit

Again it is best to copy the sample function from table TPS01 for the chosen process; in this case SAMPLE_PROCESS_SMOUTIL3. The TABLES-parameter TI_BAPIMTCS corresponds to T_BAPISTRUCT from ERP.

Function Z_CRM_MW_BP_PRICING_IN

*----------------------------------------------------------------------*
* Data definitions -----------------------------------------------------*
*----------------------------------------------------------------------*

data: lt_complex type bus_eiExtern_t,
      ls_complex type bus_eiExtern,
      ls_partner type bus_eiInstance,
      lv_customer type bu_partner,
      lv_partner_guid type bu_partner_guid,
      lv_sales_org type crmt_sales_org,
      lv_set_guid type crmt_bu_set_guid,
      ls_salesarea type crmt_bus_ei_salesarea,
      lv_klabc_data type string,
      begin of ls_klabc_data,
      knunnr type bu_partner.
      include structure crmt_bus_sales_area.
      data:   klabc type klabc,
      end   of ls_klabc_data.

*----------------------------------------------------------------------*
* Processing ----------------------------------------------------------*
*----------------------------------------------------------------------*

call function 'CRM_BUPA_BAPIMTCS_TO_MAIN'
importing
   et_bp_extern = lt_complex
   * et_not_processed_bapimtcs =
   * et_processed_partner =
   * et_error_table =
changing
   ct_bapimtcs = ti_bapimtcs[].

check sy-subrc = 0.

loop at lt_complex into ls_complex.
   clear: ls_partner, lv_customer.
* set customer
  ls_partner = ls_complex-header-object_instance.
call function 'CONVERSION_EXIT_ALPHA_INPUT'
  exporting
    input  = ls_partner-bpartner
  importing
    output = lv_customer.

* go through sales areas of customer
loop at ls_complex-crm_data-salesarea into ls_salesarea.
  ls_salesarea-salesarea-sales_org = ti_bapimtcs-data(4). " set sales org nr

* try to get ABC-classification
concatenate lv_customer ls_salesarea-salesarea '*' into lv_klabc_data.
loop at ti_bapimtcs where tabname = 'ZMW_PRICING_EXT_S'
  and data cp lv_klabc_data.
endloop.
check sy-subrc = 0.
move ti_bapimtcs-data to ls_klabc_data.

* get guid for customer SET 0030 (Pricing)
  lv_partner_guid = ls_partner-bpartnerguid.
call function 'Z_CRM_BP_GET_0030_SET_GUID'
  exporting
    iv_partner_guid = lv_partner_guid
    iv_sales_org    = lv_sales_org
    iv_channel      = ls_salesarea-salesarea-channel
    iv_division     = ls_salesarea-salesarea-division
  importing
    ev_set_guid     = lv_set_guid.
check lv_set_guid is not initial.

* update ABC-classification
update crmm_but_set0030
  set zzklabc = ls_klabc_data-klabc
  where set_guid = lv_set_guid.
endloop.
endloop.

**I'm not done yet... will continue later...**