EXAMPLES

FOR THE

DEFINITIONS AND COUNTING GUIDELINES

FOR THE APPLICATION

OF FUNCTION POINT ANALYSIS

Version 2.3

EDITION 2022.1

nesma.org

Contents

Foreword

The definitions and guidelines for the application of function point analysis are laid down in the ISO/IEC 24570:2018 standard and version 2.3 of the Nesma definition and counting guidelines that are substantively identical. In practice there is a need for practical examples that show how the definitions and counting guidelines must be applied in concrete situations. This document contains a number of concrete examples of situations that a function point analyst might be confronted with. They also show how the counting guidelines should be applied.

The examples focus mainly on showing how the functions to be counted can be identified or recognized and, when applicable, state how data element types should be counted.

Each practical situation contains:

- A **Problem description** to be solved
- A **Discussion** of the problems for the function point analyst
- The correct **Solution**
- **References** to the sections and counting guidelines involved

These examples were composed with the utmost care and reviewed by experts of Nesma as a further elaboration of the ISO/IEC 24570:2018 standard. If in practice there seems to be a contradiction between this example and the guideline, the standard prevails. To give absolute clarity that these examples are not a part of the standard, these examples are included in a separate document.

The Counting Practices Committee is developing new examples. As soon as they are ready, a new edition of this document can be released, without the need to release a new standard.

Changes with respect to edition 2021.1

In this edition one new example has been added: 41. Next to that the reference matrix has been extended to cover two additional rules. This reference matrix is also available as separate document.

Authors

These examples are compiled by the Counting Practices Committee of Nesma that consists of:

Adri Timp
Alexander Vermeulen
Eddy Schooneveld
Frank Vogelezang
Jacques van der Knaap
Jolijn Onvlee
Martin Jacobs
Vincent Barth
Wim Visser

1 STANDARD AUTHORIZATION FUNCTIONS

Problem description

A user is granted access to a computer system by the input of a computer system identification, a user identification, and a password. This logon procedure is the same for all applications that run on the system. In order to obtain access to a specific application, the user must type in the application-identification concerned, a user identification and a password. Having done this, he is authorized to carry out certain transactions of the application. The passwords are stored in a database table within the application. The system administrator can change the passwords and indicate which transactions are permitted.

Is this logon procedure counted or not? How are the authorization table and the maintenance functions counted for this?

Discussion

Do not count a function for the logon procedure. The authorization table (that contains the passwords) is an FPA table and is included as a record type in the FPA tables ILF when logical files are counted. Changing the authorization table is not counted as a separate function because one external input, one external output, and one external inquiry as a rule is counted for the FPA tables ILF.

Solution

Consider the authorization table as an FPA table and do not count a function for the logon procedure.

References to the standard

4.9, 4.20 and 5.2.k

2 SPECIFIC AUTHORIZATION FUNCTIONS

Problem description

The file *Employee* in a time registration and planning system contains personal data and indicates whether someone is a project leader, a supervisor, or an employee. An employee can be authorized to fulfill one or several of these roles. The combination of these roles determines which transactions the user can carry out. For example, only the project leader can add activities to a project, whereas other project members are not authorized to do this.

Should the file *Employee* be counted when determining the complexity of the transaction *Add activities*? After all, is this not a form of authorization?

Discussion

In order to be able to determine whether a user is allowed to carry out a certain transaction, the file *Employee* must be read. This is an internal logical file (not an FPA table) and should therefore be included in the count when determining the complexity of the transaction.

Solution

Include the file Employee as a referenced internal logical file when determining the complexity of the external input *Add activities*.

References to the standard

4.9, 4.20 and 7.3.h

3 REPORT GENERATOR AND QUERY FACILITY

Problem description

An application has been developed in a 4th generation environment. This 4GL also provides an interactive Query facility. The user can make whatever ad hoc queries he wants and produce his own reports with the help of this computerized tool.

Should this Query facility and report generator be expressed in function points and, if so, in what way?

Discussion

The query facility and the report generator fall outside the boundaries of the system to be developed. They are not included in the functional size when determining the project functional size or the application functional size.

Solution

The Query facility and the report generator are not included in the functional size when determining the application functional size or the project functional size of the application to be developed.

References to the standard

4.11 and 9.2.d

4 HELP FUNCTIONS

Problem description

A number of help screens are to be implemented in an application to be developed.

General information about the application can be obtained by clicking the menu-item *Application* of the ?-icon in the menu bar, e.g., information about which modules exist and about the relationship between them. Specific information about a particular transaction can be retrieved by clicking the menu-item *This screen* of the ?-icon, e.g., information about which fields must be filled in and the value range of the different fields.

The user cannot maintain these help texts.

How is this help facility counted?

Discussion

According to the guidelines, help screens are valued as external inquiries. The number of types of help information determines the number of functions. In this situation, there are two kinds of help information because the menu-item *This screen* provides help information at screen level and the menu-item *Application* provides help information about the application.

The complexity of such external inquiries is valued as *low* in a detailed function point analysis and as *average* in a high level function point analysis.

Solution

Count this help facility as two external inquiries.

Reference to the standard

4.13

5 ERROR MESSAGES

Problem description

A number of checks are carried out when a user enters customer data. If the user enters a customer number that already exists, the application displays the error message "customer already exists". When the user enters letters instead of numbers, the application displays the error message "customer number must consist of numbers".

Should each different error message be counted as a separate external output or as a separate external inquiry?

Discussion

The different error messages are not seen as separate functions, but as part of the external input, output, or inquiry involved.

The field where the error message is displayed must be counted as a data element type of the function. Therefore, do not count the number of different messages!

Solution

No additional functions are counted for error messages.

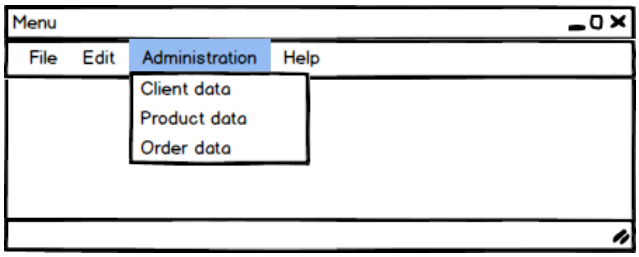
References to the standard

4.14 and 8.2.n

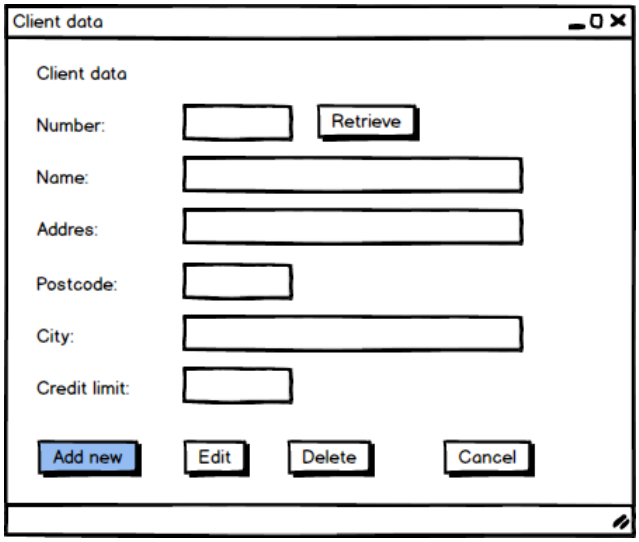
6 MENU STRUCTURES

Problem description

An application has the following menu structure.



Clicking *Client data* brings up:



How should the activation of the several transactions, like *Add new client*, through the dropdown menus be counted?

Discussion

The stepwise activation, menu structure, of a transaction (first Administration, then Client data) is not counted as a user transaction. The full menu path however, is counted as one additional data element type for the underlying functions (Retrieve, Add new, Edit, Delete).

Solution

Do not count the menu structure.

References to the standard

4.15, 4.23, 7.2.m and 7.3.b

7 FPA-TABLES

Problem description

For a sales system that records and supports sales activities, the following entity types have been defined as part of a data model in third normal-form:

Product:	product number (consists of: product group number, sequence number)
	description
	country of origin (code)
	buyer number
	price
	VAT code
Country:	country code
	name of country
VAT Rate:	VAT code
	VAT rate
	effective date
Buyer:	buyer number
	buyer name

Functions are available for each of the entity types in order to add, change, delete, and query data. Additionally, a report with all the occurrences or specimens of data can be printed for each entity type.

Should these files be considered internal logical files? And is an FPA tables ILF or an FPA tables ELF present here? If so, what is its complexity?

Discussion

Within the framework of section 4.20, the entity type *VAT Rate* is not an FPA table, but an individual internal logical file. Product is also an individual internal logical file.

Because the entity types *Country* and *Buyer* are used only for decoding the codes and numbers used (i.e., they fulfill a secondary function), they should be considered an FPA table. No additional information, for example, is maintained about the buyers.

There is an FPA tables ILF because all the entity types can be maintained. Its complexity is determined as follows: The total number of entity types (two: *Country* and *Buyer*) determines the number of record types of the FPA tables ILF. The total number of data element types (four in all) of the different entity types of the FPA table type makes up the number of data element types of the FPA tables ILF. Via the complexity matrix for internal logical files, the complexity of the FPA tables ILF can be determined (low).

Count one external input, one external output, and one external inquiry for the FPA tables ILF, regardless of the number of entity types of which the FPA tables ILF consist.

Solution

Count three internal logical files:

- *Product*: consists of one record type and seven data element types. The complexity is therefore *low*.
- *VAT Rate*: consists of one record type and three data element types, so that the complexity is *low*.
- One internal logical file for the FPA tables. There are four data element types (country code, name of country, buyer number, buyer name) and two record types (the entity types *Country* and *Buyer*). Complexity is therefore *low*.

References to the standard

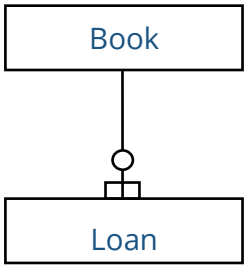
4.20, 4.21, 5.2.a and 5.2.k

8 DENORMALIZATION

In this illustration, three examples of denormalization are given for a situation in which a 1:(N) relationship exists between two entity types. The situations 1:N, (1):N, (1):(N), 1:(1), and (1):(1) speak for themselves. (See section 4.21.3 for information about the notation method.)

8.1 1:(N) with independent existence

Problem description



The normalized data model of a library application shows that there is a 1:(N) relationship between the entity types *Book* and *Loan*. A book does not have to be loaned out and, so, optionality is a factor in the relationship. If a loan is made, it always relates to one *Book* (no optionality).

The library's business rule is that a *Book* can be deleted only if a *Loan* is no longer linked to it.

Does this case involve one or two logical files?

Discussion

Book and *Loan* have a 1:(N) relationship. According to the table in section 4.21.5, the number of logical files is determined on the basis of entity dependence. Because the library may delete a *Book* only when a *Loan* is no longer linked to it, we can conclude that *Loan* also has a separate significance to the application apart from *Book* and, therefore, is entity independent with respect to *Book*. (See situation 2 in the discussion about (in)dependence in a 1:(N) relationship in section 4.21.4.) There are, then, two logical files.

Solution

Count two internal logical files.

References to the standard

4.21 and 5.2.a

8.2 1:(N) with dependent existence

Problem description

The normalized data model of a library application shows that there is a 1:(N) relationship between the entity types *Book* and *Loan*. A book does not have to be loaned out and, so, optionality is a factor in the relationship. If a *Loan* is made, it always relates to one *Book* (no optionality).

The business rule of this library, however, is that if *Book* is taken from the collection (is deleted), the library is no longer interested in *Loan* and, therefore, it may be deleted automatically when *Book* is deleted.

How many logical files must be identified in this case?

Discussion

A 1:(N) relationship exists between *Book* and *Loan*. According to the table in section 4.21.5, the number of logical files is determined on the basis of the entity dependence. Because a *Book* can always be deleted, and because any *Loan* linked to a *Book* may be deleted automatically with that *Book*, we can conclude that *Loan* is not significant to the application when separated from *Book*. Therefore, *Loan* is entity dependent with respect to *Book*. (See situation 1 in the discussion about (in)dependence in a 1:(N) relationship in section 4.21.4.) This means that there is only one logical file.

Solution

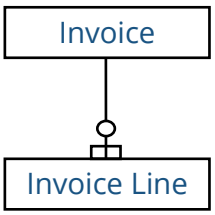
Count one logical file with two record types.

References to the standard

4.21 and 5.2.a

8.3 1:(N) with dependent existence

Problem description



The normalized data model of an invoicing system indicates that a 1:(N) relationship exists between *Invoice Header* and *Invoice Line*. The application allows users to create an *Invoice Header* first to which lines can be added later on; hence, the optionality. If users decide at a given moment to delete the *Invoice Header*, the *Invoice Lines* are also automatically deleted.

How many logical files should be distinguished here?

Discussion

Invoice Header and *Invoice Line* have a 1:(N) relationship. According to the table in section 4.21.5, the number of logical files is determined based on entity dependence. Because of the business rule that any *Invoice Lines* linked to the *Invoice Header* are deleted automatically when the *Invoice Header* is deleted, we can conclude that *Invoice Line* is entity dependent with respect to *Invoice Header*. (See situation 1 in the discussion about (in)dependence in a 1:(N) relationship in section 4.21.5) There is, then, one logical unit called *Invoice* that contains the entity types *Invoice Header* and *Invoice Line*.

Solution

Count one logical file with two record types.

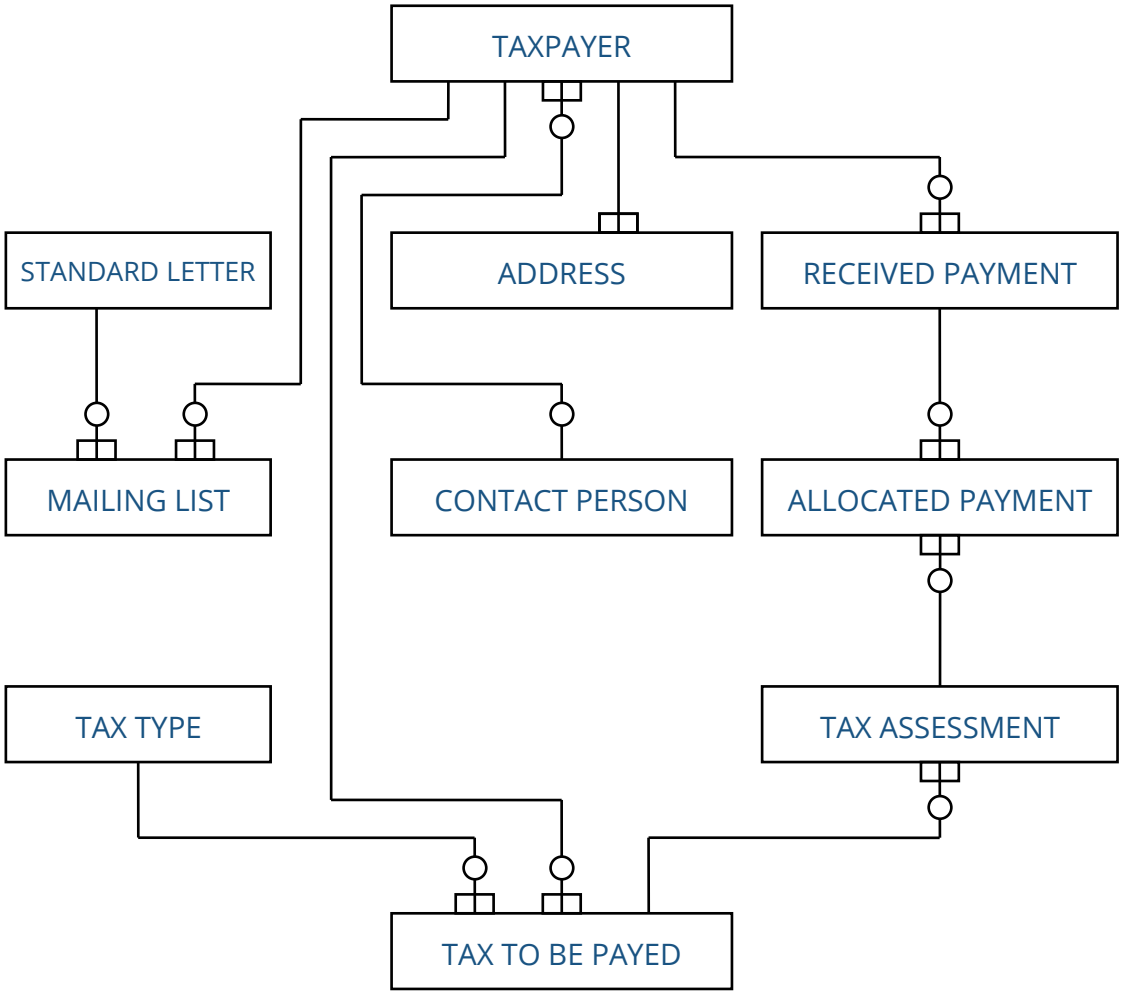
References to the standard

4.21 and 5.2.a

9 COUNTING LOGICAL FILES (DATA FUNCTIONS)

Problem description

Below a part of a normalized data model is illustrated.



This data model was made on the basis of the following user specifications.

All of the entity types named are maintained by the application.

The entity type *Taxpayer* contains the taxpayer identification number, the name, the date of birth, and some personal information about a taxpayer.

A taxpayer can have several addresses. For example, in addition to a home address (the minimum that must be present), an invoice address and/or a post office box number may also be identified.

The entity type *Standard Letter* consists of a unique letter number and of a fixed text belonging to the letter.

The entity type *Mailing List* contains only reference keys and indicates which letter is sent to which taxpayer.

The entity type *Tax Type* contains the different kinds of taxes that can be charged. The composition of *Tax Type* is as follows: code, description, and tax amount per month. (In this particular case, the tax pertains to fixed assessments that are the same for every taxpayer.)

The entity type *Tax To Be Paid* records which taxes must be paid by which taxpayer. In addition to reference keys, it also contains the date on which the tax obligation becomes effective and the date on which this obligation ends (expiration date). (The latter is usually not known when the obligation becomes effective, but is recorded later.)

The entity type *Tax Assessment* contains an amount, the final payment date, and the applicable tax period. An assessment always covers a fixed period: a year, half-year, quarter, or month.

The entity type *Received Payment* contains the amount received, the date on which the payment is received, and the amount that has still not been allocated to a *Tax Assessment*.

The entity type *Allocated Payment* contains reference keys to *Tax Assessment* and *Received Payment*, but also contains the part of the *Received Payment* that has been allocated for payment of the linked *Tax Assessment*.

The entity type *Contact Person* contains the names and some supplementary data about the employees of the tax department who can act as a contact person for a taxpayer. A particular contact person is assigned only when a taxpayer asks for advice. From that moment on, the taxpayer is always spoken to by the same person.

In principle, a taxpayer is entered into the system only when he is required to pay one or more kinds of tax. The taxpayer can be deleted as soon as he is no longer registered for a *Tax Type* (i.e., all the expiration dates in the linked entities of *Tax To Be Paid* have elapsed or, in other words, the taxpayer is no longer obliged to pay the tax) and no *Received Payments* are linked to the taxpayer anymore. When deleting the *Taxpayer* the linked occurrences in *Mailing List* will be automatically deleted. *Taxes To Be Paid* will also be deleted automatically when deleting the taxpayer, provided that no *Tax Assessments* are linked to it still.

A *Tax Assessment* is archived via a batch function one year after it has been paid in full. The archive file created contains the taxpayer identification number, the type of tax involved, the period the tax covers, the amount of the tax, the date on which the assessment was sent, and the date on which the assessment was paid in full. When the data is recorded in the archive, the *Tax Assessment* is deleted immediately together with the *Allocated Payments* linked to it.

A *Received Payment* can be deleted only if the full amount has been allocated and *Allocated Payments* are no longer linked to it.

Finally, a *Tax Type* may be deleted only if it does not have any *Tax To Be Paid* still linked to it.

How many logical files are present in this normalized data model? Are there any historical files?

Discussion

To analyze this data model, you should assume the denormalization rules given in section 4.21. The first question that must then be posed is whether any FPA tables are present. The description of the entity types shows that only the entity type Standard Letter meets the criteria for an FPA table. The only entity type whose status is ambiguous and can be discussed in this regard is *Tax Type* because it contains an amount, in addition to a code and a description. This means that it contains dissimilar kinds of data; i.e., it is not just meant for the translation of the code.

In keeping with the denormalization rules the next question that should be asked is "which entity types contain only key data"? These entity types do not count according to the guideline; however referral attributes are being counted as data element type in both logical data files associated with the present key-key entity.

At first glance, in this data model, it appears that it is only the *Mailing List* entity type. However, the taxpayer's key should then be counted on the *Standard Letter* and thus the entity type *Standard Letter* (which was referred to as the FPA table) would lose the character of the FPA table. In fact, *Mailing List* is no longer a key-key entity but the result of normalizing a recurring attribute in *Taxpayer*. Therefore, the conclusion is that *Mailing List* is not covered by the concept of "key-key entity" and still "simply" is counted.

The *Allocated Payment* entity type contains, in addition to the referring keys, the amount paid to a *Tax Assessment* and does not meet the requirements at this point. The entity type *Tax To Be Paid* also contains more data than just key data.

The other nine entity types must be examined as to how many internal logical files they represent. This is done on the basis of cardinality, optionality, and entity independence. Each pair of entity types linked via a relationship is looked at to see whether they should be included in one logical file.

The relationship between *Taxpayer* and *Contact Person* is bilaterally optional. Within the context of the guidelines, then, they are independent logical files. Additionally, *Contact Person* does not have any relationships with other entity types and is therefore one internal logical file with one record type.

The relationship between *Taxpayer* and *Address* is a bilateral-mandatory 1:N relationship. In keeping with the denormalization rules, these two entity types belong to the same internal logical file. In order to determine whether any other entity types should be included in this internal logical file, the remaining relationships of the entity type *Taxpayer* must be investigated.

The relationship between the entity type *Taxpayer* and *Mailing List* is a 1: (N) relationship. The problem description shows that the occurrences of *Mailing List* associated with a *Taxpayer* to be removed are automatically deleted. *Mailing List* is thus dependent and is therefore included in the same internal logical file.

The relationship between the entity type *Taxpayer* and *Received Payment* is a 1:(N) relationship in which *Taxpayer* may not be deleted as long as a *Received Payment* is still linked to it. This means that *Received Payment* is entity independent in relation to *Taxpayer* and does not belong to the same internal logical file as *Taxpayer* and *Address*.

The next relationship of *Taxpayer* that must be examined is its 1:(N) relationship to *Tax To Be Paid*. Here when a *Taxpayer* is deleted, the entities *Tax To Be Paid* that are linked are deleted automatically. Consequently, *Tax To Be Paid* is entity dependent on *Taxpayer* and, therefore, belongs to the same internal logical file as *Taxpayer* and *Address*. Whether any more entity types should be included in this internal logical file now also depends on the relationships of *Tax To Be Paid*.

The relationship between *Tax To Be Paid* and *Tax Assessment* is a 1:(N) relationship. The problem description above shows that an entity *Tax To Be Paid* may be deleted only if no *Tax Assessment* entities are linked to it anymore. Therefore, *Tax Assessment* has an autonomous meaning to this application and should consequently be considered entity independent in relation to *Tax To Be Paid*.

The relationship between *Tax Type* and *Tax To Be Paid* is also a 1:(N) relationship. A *Tax Type* may be deleted only if it does not have any *Tax To Be Paid* entities attached to it. *Tax To Be Paid* is therefore entity independent from *Tax Type*.

Now that all the relationships of *Taxpayer*, *Address*, and *Tax To Be Paid* have been analyzed, we can conclude that *Taxpayer*, *Address*, and *Tax To Be Paid*, together, make up one internal logical file with three record types.

As we have seen, *Received Payment* is entity independent with regard to *Taxpayer*. In order to determine whether this entity type is an internal logical file in and of itself, we must investigate its 1:(N) relationship with *Allocated Payment*. The problem description above shows that a *Received Payment* can be deleted only if there are no *Allocated Payments* attached to it anymore. This means that *Allocated Payment* is entity independent with regard to *Received Payment*. *Received Payment* is therefore an internal logical file with one record type.

Earlier we indicated that *Tax Assessment* is entity independent with regard to *Tax To Be Paid*. Additionally, *Tax Assessment* still has a 1:(N) relationship with *Allocated Payment*. According to the problem description above, any *Allocated Payments* linked to a *Tax Assessment* are deleted automatically when the *Tax Assessment* is archived and deleted. This means that an *Allocated Payment* is entity dependent on *Tax Assessment*. *Tax Assessment* and *Allocated Payment* together, therefore, make up one internal logical file with two record types.

As indicated above, *Tax To Be Paid* is entity independent in relation to *Tax Type*. Additionally, *Tax Type* does not have any relationships with other entity types and is not an FPA table. It is therefore an independent internal logical file with one record type.

The problem description above shows that a file with historical data does exist. This file is not included as an entity type in the data model. It is, however, required by the user. The composition of this file is different than the composition of the other internal logical files, so that a separate internal logical file with one record type must be counted for it.

Solution

Count internal logical files as indicated below.

Entity types:	Count as:	Number of record types:
Taxpayer + Address + Tax To Be Paid + Mailing List	1 ILF	4
Tax Type	1 ILF	1
Received Payment	1 ILF	1
Tax Assessment + Allocated Payment	1 ILF	2
Contact person	1 ILF	1
Standard letter	Count as part of the FPA tables ILF	1
Historical Tax Assessment	1 ILF	1

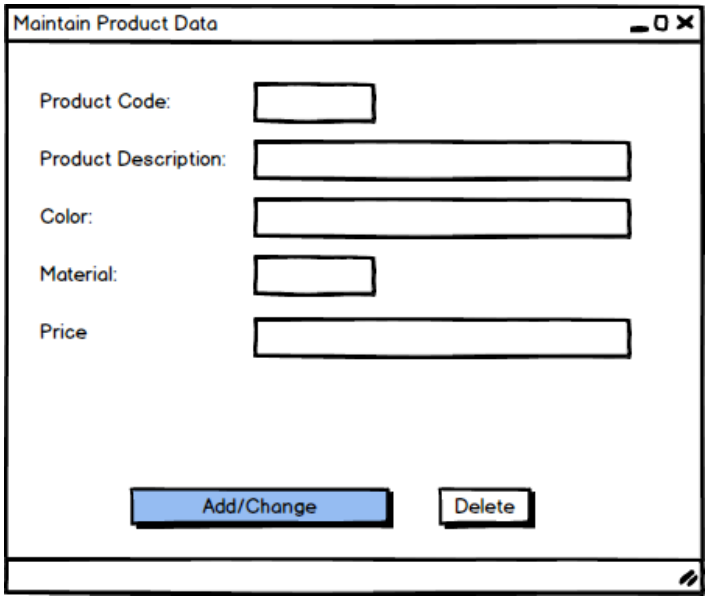
References to the standard

4.20, 4.21, 5.2.a, 5.2.b, 5.2.i and 5.2.k.

10 COMBINED EXTERNAL INPUTS

Problem description

An application provides the user with the option to maintain product data via the screen below.



After the user enters a product code, either an empty screen appears or a screen with product data entered earlier. When a new product code is typed in, other data can then also be entered into the remaining data fields on the screen. The data can be saved into the file by pressing the *Add/Change* button. When a product code already used for a product is entered onto the screen, the product data can be altered and saved with *Add/Change* button. A product can be deleted using the *Delete* button. When the user deletes data the application checks to see whether any stock of this product is present.

How many and what types of functions can be distinguished here?

Discussion

Entering the data of a new product is the first external input. Do not forget that the *Add/Change* button should be included in the count as a data element type.

Changing product data is the second external input. Note that the same set of data element types is used for another logical way of processing: to change product data. The same button is used and the button is counted for this external input too.

Deleting product data is the third external input. From a logical standpoint, this function also differs fundamentally from the other two above. If the user considers the stock data file as an individual file, this data must be included in the count when determining the complexity of this particular external input.

Displaying product data is not counted as a separate function because the user's objective is to add, change, or delete product data. Only when the user's objective is to

query the product data with this function should the displaying of data be counted as a separate external inquiry.

Solution

Count three external inputs.

References to the standard

4.7, 4.23, 7.2.g, 7.2.n, 7.2.o and 7.2.p

11 ANALYZING A TRANSACTION FILE

Problem description

A file with shop transactions is input to a Retail Management Application. Codes distinguish one transaction from another in the application. The codes are as follows:

01 = Cash sale counter

02 = Cash return counter

03 = Sale on account counter

04 = Return on account counter

05 = Cash sale, delivery other

06 = Cash return, delivery other

07 = Sale on account, delivery other

08 = Return on account, delivery other

09 = Goods dispatched

10 = Goods received

11 = Parts retrieval by service person

12 = Parts return by service person

13 = Old material dispatched

14 = Old material received

15 = Negative inventory difference

16 = Positive inventory difference

20 = Initial stock in store.

The following files are updated on the basis of the transaction code.

1 through 4 : Journal entry data, sales data, and stock data

5 through 8 : Journal entry data and sales data

9 through 14 : Journal entry data and stock data

15 and 16 : Inventory differences, journal entry data, and stock data

20 : Stock data

How many external inputs should be counted here?

Discussion

In this situation, particularly through the updating of different logical files, categories are made of different logical processing that can be identified. The transaction codes and what they stand for help in the categorization of the logical processing. The following external inputs are identified for processing the transactions:

1. Processing transactions that pertain to "counter activities" (transaction codes 1 through 4)
2. Processing transactions that pertain to "delivery other" (transaction codes 5 through 8)
3. Processing transactions that pertain to stock updates in the warehouse (transaction codes 9 through 14)
4. Processing transactions that pertain to inventory differences (transaction codes 15 and 16)
5. An external input for processing the initial stock (transaction code 20)

Solution

Count five external inputs.

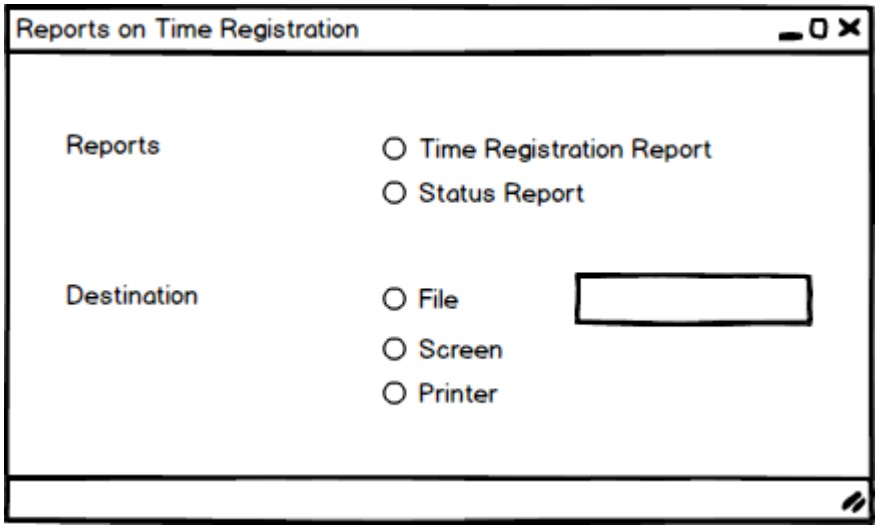
References to the standard

4.8, 7.2.a, 7.2.b, 7.2.d and 7.2.t

12 REPORTS ON DIFFERENT MEDIA

Problem description

Using the first selection screen on the following page, the user can make a selection from two kinds of reports (see the menu selection on the screen).



The *Time Registration Report* shows time registration data that has been entered, whereas the *Status Report* shows a list with the status of the time registration forms.

Also the destination should be entered on the screen.

As far as the layout and the attributes displayed are concerned, the *Time Registration Report* is exactly the same whether it is printed on paper, displayed on a screen, or exported to a file. The same is also true for the *Status Report*.

Is the report on paper a different external output than the one on screen or than the output to a file? How many external outputs are there? Is an external input counted for entering the destination?

How many external inputs should be counted here?

Discussion

The criterion used to determine the number of external outputs is that each external output must be unique. An external output is unique if no other external output exists with the same logical processing and with the same set of data element types for the application concerned.

The problem description above shows that the *Status Report* contains different information than the *Time Registration Report* so that two external outputs are present.

Additionally, the layout of the *Time Registration Report* in this example consists of the same set of data element types, regardless of the destination. The problem description does not indicate that there is a different logical processing for the different media to which the report can be sent. The *Time Registration Report* is therefore counted as one external output.

This also applies to the *Status Report*.

The data entered for the destination is control information. This data is used only for controlling where the output is sent. This means that no external input is involved.

Solution

Count two external outputs: one for the *Time Registration Report* and one for the *Status Report*.

The radio buttons for the destination and the field where a file name can be filled in must be counted as data element types when complexity is being determined.

References to the standard

4.15, 4.23, 8.2.e, 8.2.i, 8.3.b and 8.3.c

13 DAILY AND WEEKLY PROCESSING

Problem description

Each day a report of all the day's financial transactions is given. All the transactions that took place during the course of the week are placed on microfiche at the end of that week. Its layout is identical to that of the daily transaction report. After all appropriate transactions have been processed, the transaction file concerned is deleted. The content of the file is printed in the way described here only, and is ultimately placed on microfiche.

The user cannot obtain access to the file in any other way.

How many external outputs must be identified? Is an external input also to be counted for deleting the transaction file? Does the transaction file count as an internal logical file or as an external logical file?

Discussion

The transaction file is not accessible to the user and is therefore not a logical file. Because it is a temporary file, its deletion is considered a technical matter that does not play a role when the logic of the functions is assessed. From a logical perspective, therefore, there is no difference between the daily and weekly processing. The layout of the daily report is the same as the microfiche. Because the layout and the logical processing are the same, there is only one external output.

Solution

Count one external output.

References to the standard

5.2.f, 7.2.r and 8.2.a

14 CONVERSION

Problem description

The data of an existing system (ES) was initially converted for the installation of a financial system (FIS). The ES application is still being used and data is sent from FIS to ES each week.

How should the conversion software and the exchange of data be counted for the FIS application?

Discussion

When software has been developed for one-time conversion (as above), this function is not included when determining the application function point count. The conversion software, however, should be counted when determining the functional size of the project.

The weekly transmission of data from FIS is, however, a normal external output and should be included as such in the function point analysis.

Solution

Count the weekly conversion as one or more external outputs of the FIS application.

Do not count the initial one-time conversion when determining the application functional size, but count it when determining the functional size of the project.

References to the standard

3.6.2, 6.2.c, 8.2.b and 8.2.j

15 EXTERNAL OUTPUTS WITH SUMMARY INFORMATION

Two situations are covered with in this example: one in which the summary information is not considered a separate external output and one in which it is.

15.1 Summary information not counted as a separate External Output

Problem description

The following report can be produced:

Report 1A			Overview Audio				20/07/2017 Current month:07-17 Period: April-June
Land	Local Sales	Net $	QTY (x1000)	Turnover $ (x1000)	Net %	Margin $ (x1000)	
Austria	xxx.xx	xx.xx	xx.x	xxxxx	xx.x	xxxxx	
Portugal	xxx.xx	xx.xx	xx.x	xxxxx	xx.x	xxxxx	
Germany	xxx.xx	xx.xx	xx.x	xxxxx	xx.x	xxxxx	
Europe	xxx.xx	xx.xx	xx.x	xxxxx	xx.x	xxxxx	
Europe	xxx.xx	xx.xx	xx.x	xxxxx	xx.x	xxxxx	
Asia	xxx.xx	xx.xx	xx.x	xxxxx	xx.x	xxxxx	
Other	xxx.xx	xx.xx	xx.x	xxxxx	xx.x	xxxxx	

The report consists of an unknown number of pages. The totals for Europe, Asia, and Other are printed at the bottom.

How many external outputs appear here? Should the aggregated information on the report be counted as a separate function or is it the same external output?

Discussion

There is only one external output here (the report). True enough, it consists of two sections, but the sections have the same layout. Additionally, the summarizing information for Europe, Asia, and Other are inextricably bound to the rest of the report, meaning that there is one output product whose sections are not individually retrievable. No additional logical files are accessed in order to print the information desired. The information can be derived directly from the same logical processing. The guidelines indicate that only one external output should be identified in this situation.

Solution

Count one external output.

References to the standard

4.7, 8.2.g and 8.3.d

15.2 Summary information counted as a separate External Output

Problem description

An application produces the following report:

DAILY FINANCIAL TRANSACTIONS REPORT 99-99-9999 Page:99
Date: 99-99-99

Salesperson	Transaction type	Amount
x----------x	x----------x	9999,99
x----------x	x----------x	9999,99
x----------x	x----------x	9999,99
x----------x	x----------x	9999,99

FINANCIAL TRANSACTIONS TOTALS 99-99-9999 Page:99
Date: 99-99-99

Transaction type		QTY	Amount
x----------x	Day total	999	999,99
	Annual cumulative	99999	99999,99
	Daily average	999	999,99
x----------x	Day total	999	999,99
	Annual cumulative	99999	99999,99
	Daily average	999	999,99

Does one external output appear here, or are there more? Should the aggregated information be counted as a separate function on the report or is the same external output involved here?

Discussion

This report consists of one output product containing two sections. The first section is a list of all the transactions that have taken place on a given day. The second section provides daily totals, but also shows how many transactions of a certain kind have taken place in the previous year, in addition to how many per day on average. The two sections have a different layout. According to the guidelines, two external outputs should be counted if the sections can be retrieved individually or if they are realized via different logical processing. In this case, the sections cannot be retrieved individually. However, different logical processing is involved because data is used in the second section that is not contained in the first. The guidelines indicate that two external outputs should be identified here as a result.

Solution

Count two external outputs.

References to the standard

4.7 and 8.2.g

16 THE NUMBER OF DATA ELEMENT TYPES ON A REPORT

Problem description

The report below is made by product group each quarter and contains the sales for each country. (In this case, the product group is Audio.) The report is requested via a screen. The user must enter the quarter of the report. Only those countries are printed where at least one product of a given product group has actually been sold. The totals for Europe, Asia, and Other are the sums of the respective columns. The percentage is calculated from Qty and Turnover.

Report 1A		Overview Audio (1)			20/07/2017 Current month:07-17 (2) Period: April-June (3)
Country	Local Sales	Turnover $ (x1000)	Net %	Margin $ (x1000)	
Austria (4)	xxx.xx (5)	xxxxx (6)	xx.x (7)	xxxxx (8)	
Portugal	xxx.xx	xxxxx	xx.x	xxxxx	
Germany	xxx.xx	xxxxx	xx.x	xxxxx	
Europe (9)	xxx.xx (10)	xxxxx (11)	xx.x (12)	xxxxx (13)	
Europe	xxx.xx	xxxxx	xx.x	xxxxx	
Asia	xxx.xx	xxxxx	xx.x	xxxxx	
Other	xxx.xx	xxxxx	xx.x	xxxxx	

How many data element types should be counted when determining the complexity of the external output?

Discussion

The data element types to be counted are denoted by the figures in parentheses. The date in the heading is standard, just as "Report 1A" in the upper left hand corner, and, consequently, is not counted. The percentage is counted once in the detailed line and once again in the total line for each geographical unit because the logical processing differs. Each column total is counted. The variable fields "Audio", "Current Month", and "Quarter" are also counted. Additionally, the data entered on the screen are counted as data element types (one in this case) when the complexity of the external output is determined, and the initiation trigger is counted.

Solution

Fifteen data element types are distinguished in total.

References to the standard

4.23, 8.3.a, 8.3.b, 8.3.d, 8.3.f and 8.3.g

17 COMBINED EXTERNAL OUTPUTS

Problem description

At a user's command, a commercial application prints an action list on which appear, per department, the requests for quotation that require a response or that have already received a response. The action list shows all the requests for quotations grouped by department. Each request for quotation that the application prints contains the following status: "Request under consideration", "Current quotation", "Signed contract", and "Missed deal". The information displayed always covers the past week, calculated from the date of the request of the action list.

In practice, an action list can take many forms. This particular illustration presents four variants of the action list, categorized by degree of user-friendliness. Action list A, for example, provides the same information as action list D, but is not nearly as user-friendly. Action lists B and C should be considered somewhere between A and D as regards user-friendliness.

How many external outputs should be counted for each variant (A, B, C, and D)? This question should be answered within the context of two situations:

1. When the actions cannot be retrieved by status.
2. When the actions can be retrieved by status, and result in one report per status.

A discussion is carried out and a solution is given for each variant (A, B, C, and D) for both situations.

17.1 Variant A

Action list A contains all the data elements displayed in the report below.

XXX Dept. Action List								20/07/17	Page 1
Status:	Cust	Sales person	Req. Date	Req. for Quot.	Quot. Date	Contract Date	Expiry Date	Turnover	Reason missed
Req. u. cons.	xxxxx	xxxxxx	--/--/--	--/--/--	--/--/--	--/--/--	--/--/--	xxxxxx	xxxxxxxx
.....	xxxxx	xxxxxx	--/--/--	--/--/--	--/--/--	--/--/--	--/--/--	xxxxxx	xxxxxxxx
.....	xxxxx	xxxxxx	--/--/--	--/--/--	--/--/--	--/--/--	--/--/--	xxxxxx	xxxxxxxx
Current quote	xxxxx	xxxxxx	--/--/--	--/--/--	--/--/--	--/--/--	--/--/--	xxxxxx	xxxxxxxx
.....	xxxxx	xxxxxx	--/--/--	--/--/--	--/--/--	--/--/--	--/--/--	xxxxxx	xxxxxxxx
.....	xxxxx	xxxxxx	--/--/--	--/--/--	--/--/--	--/--/--	--/--/--	xxxxxx	xxxxxxxx
Signed Contract	xxxxx	xxxxxx	--/--/--	--/--/--	--/--/--	--/--/--	--/--/--	xxxxxx	xxxxxxxx
.....	xxxxx	xxxxxx	--/--/--	--/--/--	--/--/--	--/--/--	--/--/--	xxxxxx	xxxxxxxx
.....	xxxxx	xxxxxx	--/--/--	--/--/--	--/--/--	--/--/--	--/--/--	xxxxxx	xxxxxxxx
Missed deal	xxxxx	xxxxxx	--/--/--	--/--/--	--/--/--	--/--/--	--/--/--	xxxxxx	xxxxxxxx
.....	xxxxx	xxxxxx	--/--/--	--/--/--	--/--/--	--/--/--	--/--/--	xxxxxx	xxxxxxxx
.....	xxxxx	xxxxxx	--/--/--	--/--/--	--/--/--	--/--/--	--/--/--	xxxxxx	xxxxxxxx

Situation A.1: The actions cannot be retrieved by status

Discussion

There is one report. What must be investigated is whether there are several external outputs despite this. There can be several external outputs only when there are several sections with a different logical layout. This is not the case here and, consequently, only one external output should be counted.

Solution

Count one external output.

References to the standard

4.7, 8.1 and 8.2.g

Situation A.2: The actions can be retrieved by status and result in one report per status

Discussion

There are four reports (one for each status). The question now is whether identical functions are present. Functions are identical when:

- the logical layout of the reports is the same and
- the processing is the same, whereby the use of the same selection criteria with a different selection value is not seen as a different processing

In this particular situation, there are four identical logical layouts. For each report, the selection criterion "Status of the action" is used, but contains a different value in the four cases. Consequently, only one external output must be counted for the four reports.

Solution

Count one external output.

References to the standard

4.7, 8.1 and 8.2.g

17.2 Variant B

XXX Dept. Action List 20/07/17 Page 1

Status:	Cust.	Sales person	Req. Date	Req. for Quot.	Quot. Date	Contract Date	Expiry Date	Turnover	Reason missed
Req. u. cons.	xxxxx	xxxxxx	--/--/--	--/--/--	*)	*)	*)	xxxxxxx	*)
.....	xxxxx	xxxxxx	--/--/--	--/--/--	*)	*)	*)	xxxxxxx	*)
.....	xxxxx	xxxxxx	--/--/--	--/--/--	*)	*)	*)	xxxxxxx	*)
Current quote	xxxxx	xxxxxx	--/--/--	--/--/--	*)	*)	*)	xxxxxxx	*)
.....	xxxxx	xxxxxx	--/--/--	--/--/--	--/--/--	*)	*)	xxxxxxx	*)
.....	xxxxx	xxxxxx	--/--/--	--/--/--	--/--/--	*)	*)	xxxxxxx	*)
Signed contract	xxxxx	xxxxxx	--/--/--	--/--/--	--/--/--	*)	*)	xxxxxxx	*)
.....	xxxxx	xxxxxx	--/--/--	--/--/--	--/--/--	--/--/--	--/--/--	xxxxxxx	*)
.....	xxxxx	xxxxxx	--/--/--	--/--/--	--/--/--	--/--/--	--/--/--	xxxxxxx	*)
Missed deal	xxxxx	xxxxxx	--/--/--	--/--/--	--/--/--	--/--/--	--/--/--	xxxxxxx	*)
.....	xxxxx	xxxxxx	--/--/--	--/--/--	--/--/--	*)	*)	xxxxxxx	xxxxxxx
.....	xxxxx	xxxxxx	--/--/--	--/--/--	--/--/--	*)	*)	xxxxxxx	xxxxxxx
	xxxxx	xxxxxx	--/--/--	--/--/--	--/--/--	*)	*)	xxxxxxx	xxxxxxx

Action list B is almost the same as action list A. In action list B, however, data element types that do not have a value in the internal logical file are not printed. (They cannot have a value when actions have a certain status.) The fact that a data element type cannot have a value is denoted in the report layout by *). Only a difference in appearance exists in comparison to action list A.

Discussion

Visually it seems that there are several sections with a different layout. That is, it would seem that a different logical layout can be identified for each action status. However, all distinguishable sections contain the same data element types (i.e., the column headings are the same). The only difference is that a value is not printed for certain data element types because they do not yet have a value in the specified status of the action. According to the definition of logical layout, this means that the logical layouts are the same. Counting must therefore be carried out in the same way as for action list A. This applies to both situation B.1 and situation B.2.

Solution

Count one external output for both situation B.1 and situation B.2.

References to the standard

4.7, 8.1 and 8.2.g

17.3 Variant C

XXX Dept. Action List 20/07/17 Page 1

Status:	Cust.	Sales person	Req. Date	Req. for Quot.	Quot. Date	Contract Date	Expiry Date	Turnover	Reason missed
Req. u. cons.	xxxxx	xxxxxx	--/--/--	--/--/--	*)	*)	*)	xxxxxxx	*)
.....	xxxxx	xxxxxx	--/--/--	--/--/--	*)	*)	*)	xxxxxxx	*)
.....	xxxxx	xxxxxx	--/--/--	--/--/--	*)	*)	*)	xxxxxxx	*)
Current quote	xxxxx	xxxxxx	**)	**)	--/--/--	*)	*)	xxxxxxx	*)
.....	xxxxx	xxxxxx	**)	**)	--/--/--	*)	*)	xxxxxxx	*)
.....	xxxxx	xxxxxx	**)	**)	--/--/--	*)	*)	xxxxxxx	*)
Signed contract	xxxxx	xxxxxx	**)	**)	**)	--/--/--	--/--/--	xxxxxxx	*)
.....	xxxxx	xxxxxx	**)	**)	**)	--/--/--	--/--/--	xxxxxxx	*)
.....	xxxxx	xxxxxx	**)	**)	**)	--/--/--	--/--/--	xxxxxxx	*)
Missed deal	xxxxx	xxxxxx	**)	**)	**)	*)	*)	xxxxxxx	xxxxxxxx
.....	xxxxx	xxxxxx	**)	**)	**)	*)	*)	xxxxxxx	xxxxxxxx
.....	xxxxx	xxxxxx	**)	**)	**)	*)	*)	xxxxxxx	xxxxxxxx

Action list C is almost identical to action list A. Just as with action list B, all action list C's data element types that do not have a value are not printed. These data element types are denoted by *). Additionally, values no longer relevant as a result of the status of the action are not printed. These fields are denoted with **).

Discussion

According to the definition, the logical layouts prove to be the same once again because all data element types appear in each section. The only difference is that a value is not printed for certain data element types because the value is no longer relevant or because it does not appear in the internal logical files. Count as you did for action list A.

Solution

Count one external output for both situation C.1 and situation C.2.

References to the standard

4.7, 8.1 and 8.2.g

17.4 Variant D

XXX Dept. Action List

Requests under consideration

Customer	Salesperson	Request date	Quotation date	Turnover
Xxxxxx	xxxxxx	--/--/--	--/--/--	xxxxxx
xxxxxx	xxxxxx	--/--/--	--/--/--	xxxxxx
xxxxxx	xxxxxx	--/--/--	--/--/--	xxxxxx

Current quotes:

Customer	Salesperson	Quotation date	Turnover
xxxxxx	xxxxxx	--/--/--	xxxxxx
xxxxxx	xxxxxx	--/--/--	xxxxxx
xxxxxx	xxxxxx	--/--/--	xxxxxx

Signed contracts:

Customer	Salesperson	Contract date	Expiry date	Turnover
xxxxxx	xxxxxx	--/--/--	--/--/--	xxxxxx
xxxxxx	xxxxxx	--/--/--	--/--/--	xxxxxx
xxxxxx	xxxxxx	--/--/--	--/--/--	xxxxxx

Missed deals:

Customer	Salesperson	Reason missed	Turnover
xxxxxx	xxxxxx	xxxxxxxxxxxxx	xxxxxx
xxxxxx	xxxxxx	xxxxxxxxxxxxx	xxxxxx
xxxxxx	xxxxxx	xxxxxxxxxxxxx	xxxxxx

This action list contains exactly the same information as action list C, but is a bit different in layout.

Situation D.1: The actions cannot be retrieved by status

Discussion

There is one report here. What must be investigated is whether there are several external outputs despite this. There can be several external outputs only when several sections exist and each section has a different layout. In this case, four individual logical layouts would exist because each of the sections consists of a different set of data

element types. (Data element types that do not have a value or that are not relevant do not appear in action list D.) This in turn would seem to indicate that four external outputs exist. According to the guidelines, however, for this to be true a user would have to be able to retrieve each section individually or there should be different logical processing for bringing about each section. The sections in this example cannot be retrieved individually. According to the counting guidelines, furthermore, different logical processing does not exist here because all the sections report about "actions" and are accomplished on the basis of the same internal logical files. One external output should therefore be counted.

Solution

Count one external output.

References to the standard

4.7, 8.1 and 8.2.g

Situation D.2: The actions can be retrieved by status and result in one report per status

Discussion

There are four reports (one for each status). The question now is whether identical functions are involved. Identical functions are said to be present when the logical layout and the logical processing are the same in which the use of the same selection criterion with a different selection value is not seen as a different processing. There are four different logical layouts in this situation for the same reasons as in situation D.1. Four external outputs must then be counted as a result.

Solution

Count four external outputs.

References to the standard

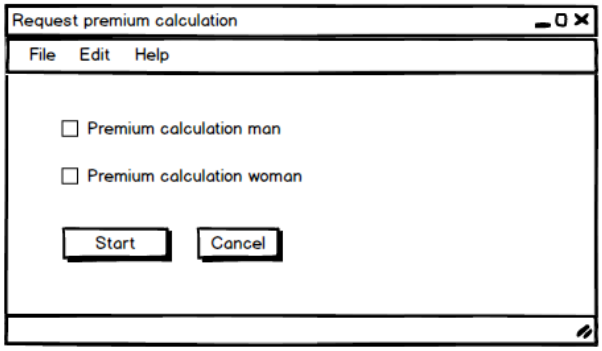
4.7, 8.1 and 8.2.g

18 COMBINATION EFFECTS WITH FUNCTIONS

18.1 One combination option

Problem description

The user has a screen with which insurance premiums can be calculated and printed.



The following options appear on the screen:

1. Premium calculation for a Man
2. Premium calculation for a Woman

The user can check off one of these options, or both simultaneously.

Suppose that the report for a Man has a different logical layout and/or logical processing than the report for a Woman. Additionally, when both options have been checked, the reports Man and Woman are printed successively and conclude with a sub-total line, a group discount, and a final amount line.

How many external outputs should be counted in this case?

Discussion

The functions *Calculate Premium for Man* and *Calculate Premium for Woman* are unique functions and, so, one external output is counted for each. They are not external inquiries because calculations are made. An additional external output is counted because the combined report is more than the sum of its parts.

Solution

Count the following external outputs:

- Two external outputs for the unique choices Man and Woman
- One external output for the combined report

Reference to the standard

8.2.s

18.2 Multiple combination options

Problem description

The user has a screen with which insurance premiums can be calculated and printed. The following options exist:

1. Premium calculation for a Man
2. Premium calculation for a Woman
3. Premium calculation for a Child

The user can check one, two, or three of these options.

Suppose that the reports for a Man, a Woman, and a Child each have a different logical layout and/or a different logical processing. Here, again, the reports for the checked options are printed successively and conclude with a sub-total line, a group discount, and a final-amount line.

How many external outputs should be counted in this case?

Discussion

The user can now execute the following (individual or combined) functions:

Man, Woman, Child, Man + Woman, Man + Child, Woman + Child, Man + Woman + Child.

The reports for Man, Woman, and Child are each separate and unique external outputs. When combinations of the above are made, furthermore, more information becomes available than the sum of the individual parts. When combinations are made and used, however, the additional information is generated as a result of similar logical processing. (There are no different kinds of combination effects.) Consequently, one external output is counted in total for the combinations.

Note: If the processing for the combination Man + Woman + Child, for example, would have been different in comparison to other combinations, then two additional external outputs would have been counted for the combined reports. (See guideline 8.2.s.)

Solution

The number of external outputs is:

- Three external outputs for the unique choices Man, Woman, and Child
- One external output for all of the combinations together

Reference to the standard

8.2.s

19 QUERYING WITH DIFFERENT SEARCH KEYS

In this example, two situations are dealt with in which the option exists to retrieve data with different criteria.

19.1 Combination of unique and non-unique search criteria

Problem description

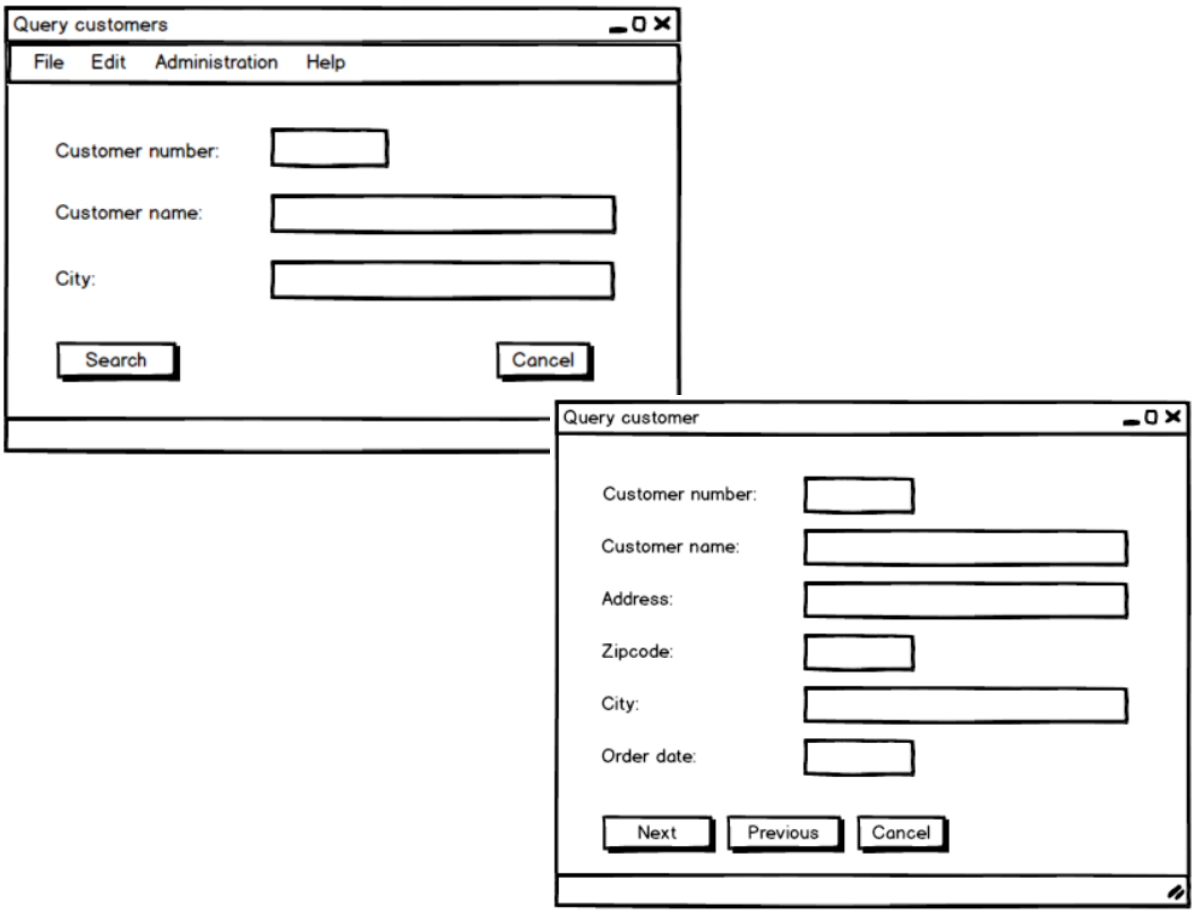
When a unique Customer Number is entered into the function *Query Customers*, data about the customer attached to that number will appear on the screen. The buttons Next and Previous are not active when this is done.

If a Customer Name (or part of a Customer Name) is entered, all customers with that particular name are retrieved. However, only the first customer with this name is displayed on the screen. When the name of a City is also entered, only those customers that reside there are selected.

If only the city is entered, then all the customers from that city are selected.

The buttons *Next* and *Previous* allow the user to browse forward or backwards through the customers selected.

How many and what type of functions should be counted?



Discussion

In this case, the user has the option to enter either the customer number or the customer name, and may even combine the customer name with the city. Two exclusive or separate selections are possible, each of which is considered an individual function.

Querying by customer number is an external inquiry. The size of the output is fully determined: namely, all data about a particular customer.

The external inquiry consists of seven data element types: customer number (twice), customer name, address, zip code, city, and order date. Besides the error message and the *Search* button are counted. Total number is nine data-element-types.

Querying by (a part of a) customer name and/or by city is an external output. The output varies in size because the number of customers that will be selected is not known beforehand. In this case, there is only one external output because the user has more options in which the selections he makes do not exclude each other (i.e., an and/or situation).

Ten data element types determine the function's complexity: customer number, customer name (twice), address, zip code, city (twice), and order date plus the error message and the *Search* button.

The buttons *Next* and *Previous* are used to navigate through the output and are therefore not counted as additional functions or data element types.

Solution

Count one external inquiry with nine data element types for querying by customer number.

Count one external output with ten data element types for querying by customer name and/or by city.

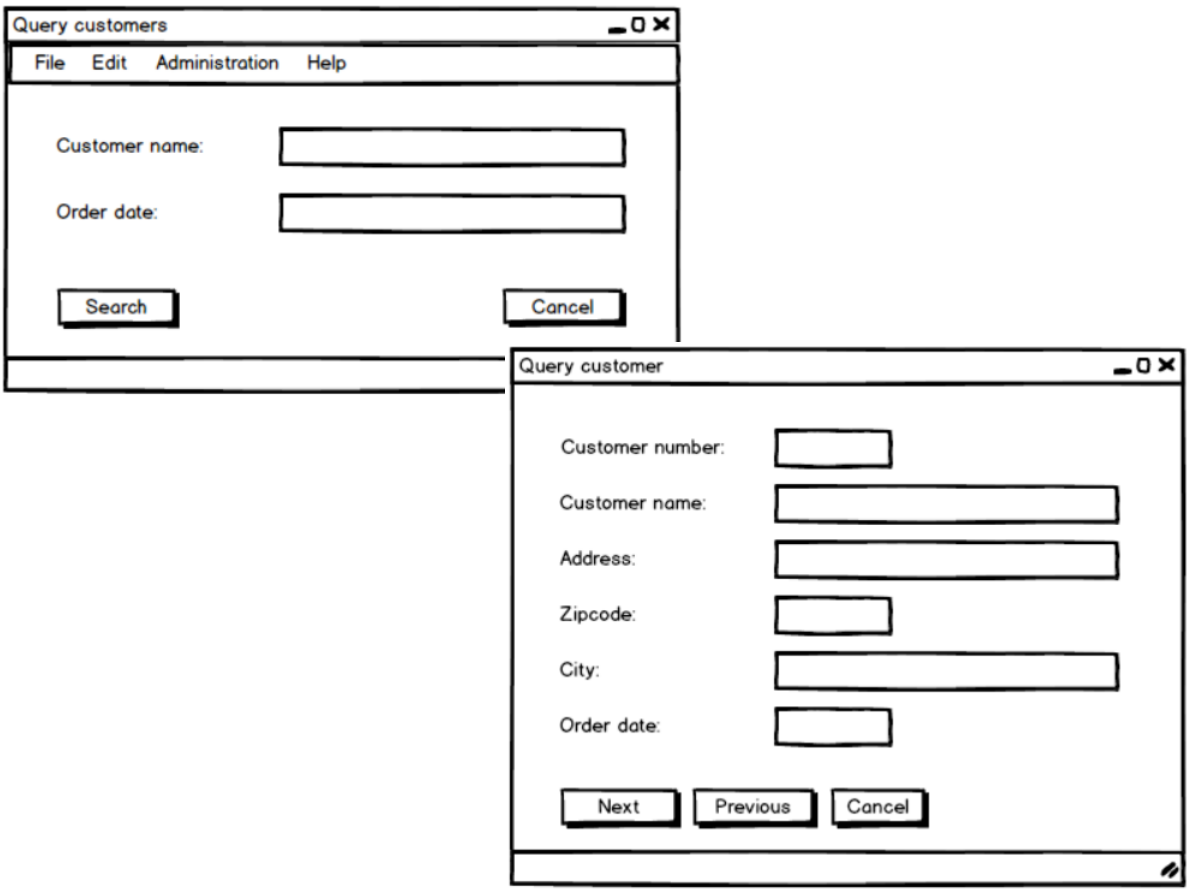
References to the standard

4.23, 8.2.a, 8.2.c, 8.2.q, 8.3.a, 8.3.b, 8.3.g, 9.2.h and 9.3

19.2 Combination of non-unique search keys

Problem description

An application has the two screens below at its disposal.



The user can query the data of a customer either via *Customer Name* or via *Order Date*.

The buttons *Next* and *Previous* allow the user to move to a following or a previous customer that meets the selection criterion.

How many external outputs and/or external inquiries are present here?

Discussion

The output for querying by name varies in size because it is not known beforehand how many customers are selected.

The size of the output for querying by order date is also variable and cannot be predicted. The logical processing is different for both queries.

Two individual external outputs should be counted because the user must choose between querying by name and querying by order date. A combination is not an option.

The Next and Previous button are used to navigate through the output and are therefore not counted as additional functions or as data element types.

Solution

Count one external output with eight data element types for querying by name (the seven data element types are customer number, customer name (twice), address, zip code, city, and order date plus the activation button)

Likewise, count one external output with eight data element types for querying by order date.

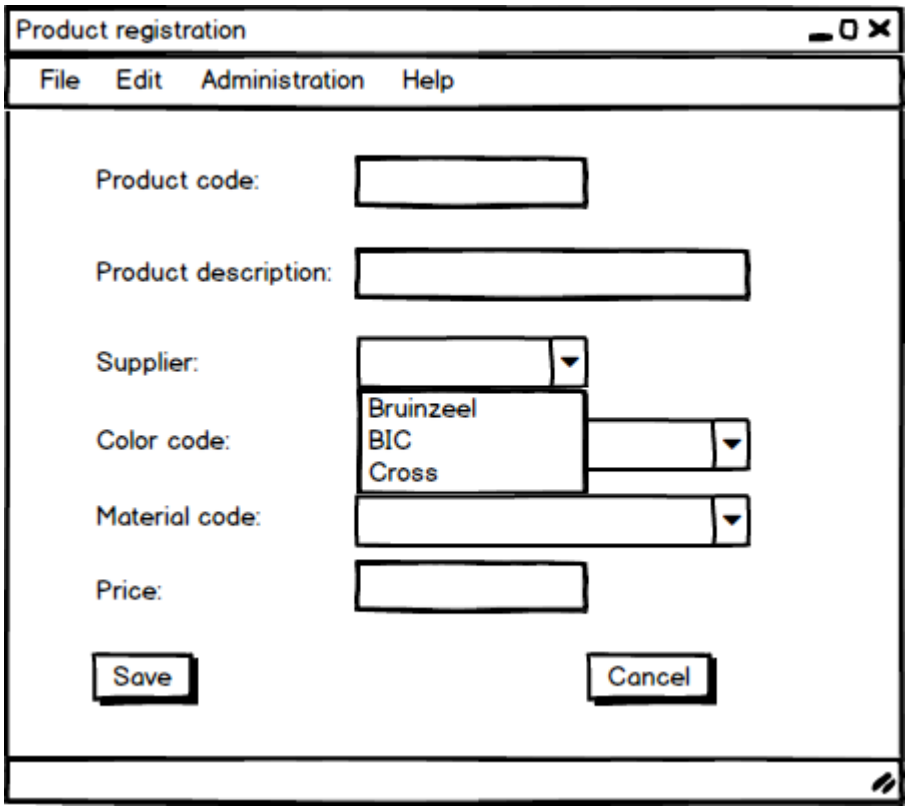
References to the standard

4.23, 8.2.a, 8.2.c, 8.2.q, 8.3.a, 8.3.b and 8.3.g

20 SCREENS WITH LIST FUNCTIONS

Problem description

When entering product data, the user can use drop-down-list-boxes in the fields *Supplier*, *Color Code*, or *Material Code* in order to display all the valid codes or numbers of the field chosen, as well as the description that goes along with each of these codes or numbers. A code or a number can then be chosen and copied from this list to the input field. The color description is retrieved from a file containing the attributes color code and color description. Material description and supplier name come from two files containing various data about the material or the supplier, respectively.



Should these kinds of supporting functions be counted and, if so, how?

Discussion

Count different list functions of the application basically as separate external outputs. The list functions are external outputs because their outputs vary in size. The functions are different because (in this example) the logical layout of the list function that displays material codes is different from the list function that provides potential suppliers with their codes (i.e., the list function consists of other data element types and provides information from a different logical file).

Although this also applies to the list function with color codes, this function is not counted as a separate external output because it pertains to an FPA table. One external output is counted for the FPA tables ILF in the entire application.

Copying to the input field is not counted as a separate function.

Solution

In this example, count the list function for supplier and material each as one external output, provided that the function has not been counted elsewhere already.

Additionally, count one external input for saving product information.

Do *not* count a separate external output for the list function with color codes.

References to the standard

4.16, 4.20, 5.2.k, 8.2.a and 8.2.t

21 BROWSE AND SCROLL FUNCTIONS

Browse and scroll functions may appear in many shapes and sizes. FPA strives to count these different shapes and sizes in the same fashion when they provide the same functionality even though they have been realized in a different way. As a result, this illustration will go into a large number of different situations and will indicate how each situation should be counted.

21.1 Selection via uniquely identifying data

Problem description

A unique customer number is entered for the function "Show Customer Data". Once this has been done, the following situations can present themselves:

1. The data of the customer concerned is displayed. No option exists to use functions keys in order to retrieve the data of a different customer.
2. The data of the customer concerned is displayed, after which the data of the following or previous customer can be retrieved by using function keys.
3. The core data of all customers is displayed on an overview screen (one line per customer), starting from the customer number entered. The user can scroll through this data when the screen cannot display all of it because of a lack of room.
4. The core data of all customers is displayed on an overview screen (one line per customer), starting from the customer number entered. The user can scroll through this data when the screen cannot display all of it because of a lack of room. After one of the customers on this screen has been selected, the application displays its detailed data.
5. The detailed data of the customer concerned is displayed. Via a function key, a user can then request a screen-display overview of the core data of all customers (one line per customer), starting from the customer that was shown on the detailed screen. The user can then scroll through this data if the screen cannot display all of it because of a lack of room. A particular customer can then again be selected on the overview screen, after which the application displays its data on a detailed screen.
6. The core data of all customers is displayed on an overview screen (one line per customer), starting from the customer number entered. The user can scroll through this data if the screen cannot display all of it because of a lack of room. After one of the customers on this screen has been selected, the application displays its detailed data, after which the data of the following or previous customer can be retrieved by using function keys.

Which external outputs and/or external inquiries should be identified in each of the situations above?

Discussion

Situation one is clearly an external inquiry; nothing more, nothing less. The customer is determined in a unique fashion by its customer number. Only one customer has that number. No opportunity to browse is given.

Situation two also seems to be a case of an external inquiry. In reality, however, the function allows the user to browse through all the customers from a defined starting point. The entire collection of customers is provided and the quantity of customers that can appear varies. This means that one external output is present.

Situation three also has an external output. Here, too, a starting point has been defined. Several customers are displayed and the number of customers that will follow from that starting point is not known. As a result, one external output must be identified. It does not matter whether the user can scroll further with the function key because there are more customers than the screen can display. Scrolling within the same collection is not a separate function, but rather a part of the external output. The only difference between situation three and two is that all the data of a customer can be displayed in two and only core data in three.

Two functions are in fact provided in situation four. Just as in situation three, the overview screen is an external output.

Displaying data of a specific customer on the detailed screen is considered a different functionality because a different set of data element types is involved. (Only the core data of a customer appears on the overview screen, whereas all the data of a customer is displayed on the detailed screen.) Moreover, calling the function is optional. Additionally, the function itself could exist independently. Therefore, this function is also an elementary process. This, in turn, means that the displaying of detailed data is counted as a separate function. It is an external inquiry because the user cannot scroll through information once he is on the detailed screen. There is one external output and one external inquiry.

From a functional standpoint, situation five is the same as situation four, only the screens appear in a different sequence. The sequence of screens is not important to FPA. The same functions identified in situation four are identified in five.

Just as in situation four, two functions are provided in situation six. The overview screen is once again an external output. Displaying data of a specific customer on the detailed screen is considered a different functionality because a different set of data element types is involved. (Only the core data of a customer appears on the overview screen, whereas all the data of a customer is displayed on the detailed screen.) Moreover, calling the function is optional. The function itself could exist independently. Therefore, this function is also an elementary process. This, in turn, means that the displaying of detailed data is counted as a separate function. Unlike situation four, however, situation six does allow the user to browse through the detailed screens and, so, the same functionality is provided as in situation two. The displaying of detailed data is therefore counted as one external output. As a result, situation six has two external outputs in total.

Solution

Identify the following functions:

Situation 1: One external inquiry

Situation 2: One external output

Situation 3: One external output

Situation 4: One external output and one external inquiry

Situation 5: One external output and one external inquiry

Situation 6: Two external outputs

References to the standard

4.17, 8.2.a, 8.2.c, 8.2.u, 9.2.c, 9.2.f, 9.2.g, 9.2.h and 9.2.j

21.2 Selection via non-uniquely identifying data, followed by browsing

Problem description

When a user enters a unique representative number for the function *Show Customer Data*, the first customer of the representative concerned is displayed. Using the functions keys, the user can then browse to a previous customer of the representative or to a following one.

Are there one or more external inquiries present here and/or one or more external outputs?

Discussion

When a user enters a unique representative number, he does not know how many customers this representative has. This means that the output varies in size and that it is counted as one external output. The browse function is a part of the external output, and the function keys used to browse with are not counted as an additional function or as data element types.

Solution

Count one external output.

References to the standard

4.17, 8.2.a, 8.2.c, 8.2.u, 8.3.g and 9.2.j

21.3 Selection via uniquely identifying data, followed by browsing after another selection

Problem description

When querying customer data via a unique customer number, a user can retrieve a previous or following customer of the same representative by using function keys.

Is there one or more external inquiries present here and/or one or more external outputs?

Discussion

When a user queries the data by customer number, the output is determined uniquely by that customer number and does not vary in size. This is an external inquiry.

When function keys are used to retrieve the previous or the following customer of the same representative, the customer number of the customer displayed and the representative number are used as search keys. This means that a different logical processing is necessary. Even though the customer specifically shown has been determined uniquely, the user now browses through the collection of customers belonging to a single representative. The size of this collection varies and, therefore, an external output is present.

Solution

Count one external inquiry and one external output.

References to the standard

4.17, 8.2.a, 8.2.c, 8.2.u, 9.2.c, 9.2.f, 9.2.g, 9.2.h and 9.2.j

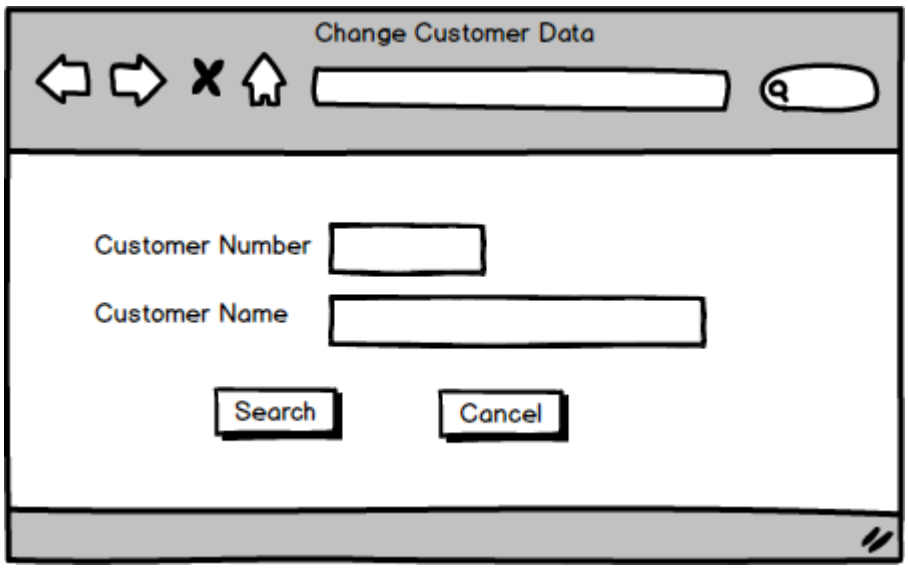
22 SELECTION SCREENS AND CHANGING DATA WITH A SEARCH KEY

This example treats a change function whose objective is twofold: A user should be able to change customer data but, before he does this, he should first be able to select the customer in a user-friendly way. From a functional standpoint, this can be realized in different ways. This section will discuss two different implementations of this functionality and will indicate how counting should take place in both situations.

22.1 Selection via a separate selection screen

Problem description

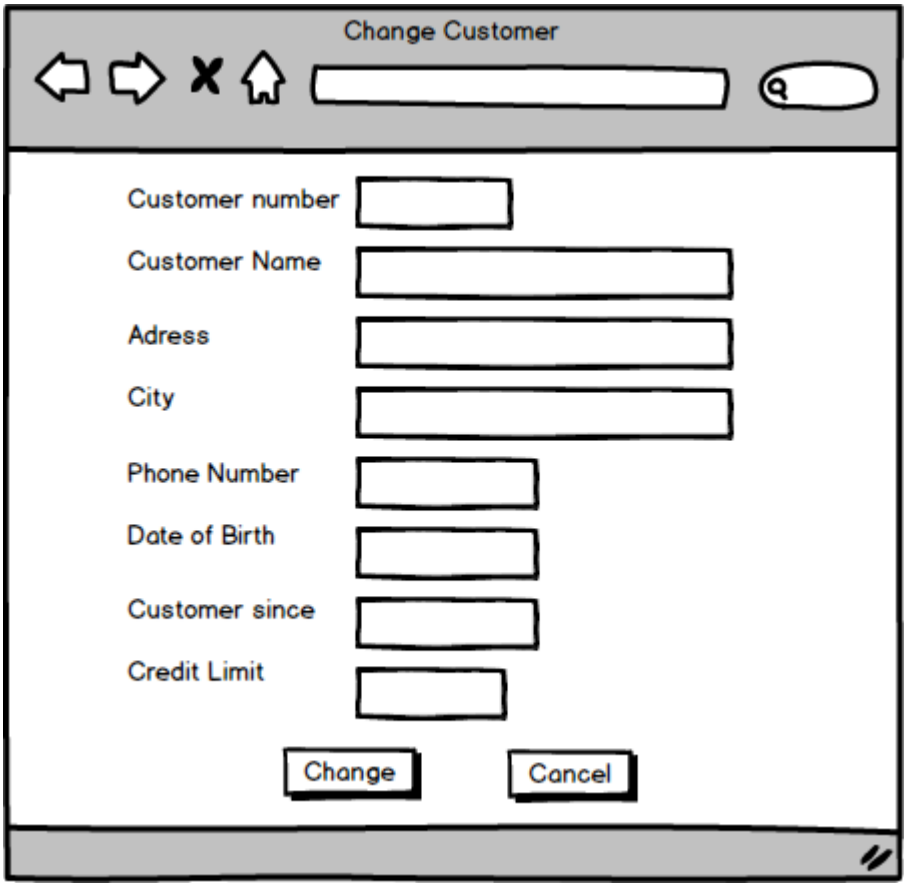
Using a menu, the user indicates that he wants to change customer data. The application subsequently presents screen 1 on which the user must enter a unique customer number or a (part of a) customer name. The user should not enter both.



Screen 1

When a unique customer number is entered, the data for the customer concerned appears on the change screen. (See screen 2.)

When a customer name (or a part of a customer name) is entered, the application retrieves all customers with that name. If only one customer is found, the data of that customer appears immediately on the change screen (screen 2).

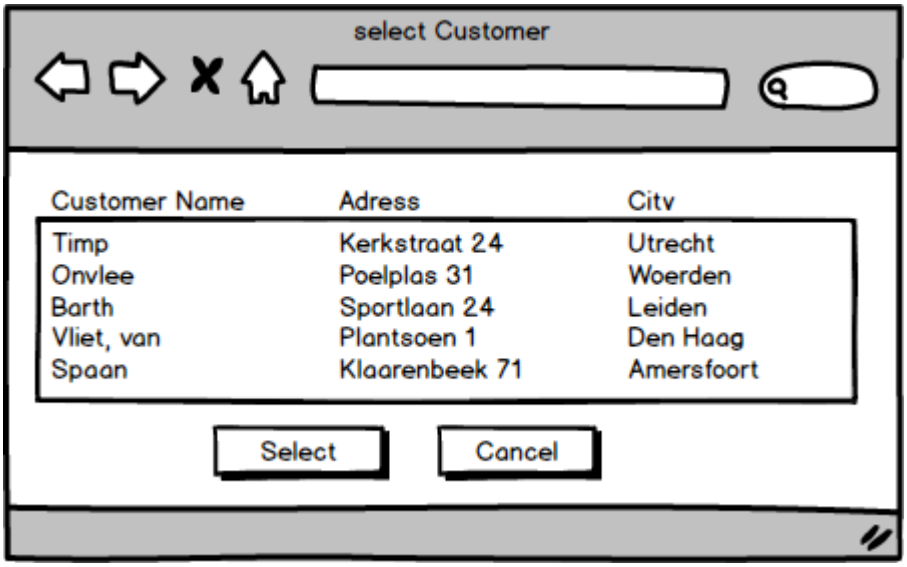


Screen 2

If several customers are found, they appear on the selection screen. (See screen 3.)

After the user enters the customer desired via the Choice field (screen 3), more extensive data of that customer appears on the change screen (screen 2).

The customer data can then be changed via screen 2.



Screen 3

Does the option to select data make up a part of the option to change data, or is it a separate function? Is a separate external input counted for entering the customer desired? Is the display of the customer data a separate external inquiry? How many functions should be counted in total?

Discussion

If the correct customer data is displayed on the change screen (screen 2) as a result of a customer number having been entered on screen 1, then the display of the data to be changed is part of the external input "Change Customer". This display is not counted as an individual external inquiry.

Counting is carried out as follows, however, when the user enters a non-unique customer name and customers meeting this criterion appear on screen 3, after which the customer to be changed can be chosen. Searching for the customer via customer name results in a displayed selection of customers on the selection screen (screen 3). This selection is not determined fully in size beforehand; i.e., the size of the selection varies depending on how many customers meet the selection criterion (name). The customer desired can then be selected. This display of selected customers on a separate screen is seen as additional functionality.

Since the logical layout of this overview is different, an additional function is counted; because the output varies in size, one external output should be counted.

Sometimes an application retrieves only one customer when a user selects via customer name. When this happens, the application does not display the selection screen (screen 3). Instead, it immediately retrieves the detailed data that screen 2 displays. This is merely an optimization and is therefore considered to be part of the external output counted for the selection screen (screen 3). No additional external output or external inquiry is counted for this.

After the user has made a selection on screen 3, all the data for the customer selected is displayed on screen 2, after which changes can be made. The display of data for the customer selected on screen 2 (just as when selecting via customer number) is seen as a part of the change function and is not an individual external inquiry.

Indicating the selected customer via the Choice field does not result in a separate external input.

The change function is the same in both cases, and is therefore counted as one external input.

Solution

Count one external input for the change function.

Count one external output for displaying the customers that meet the selection criterion.

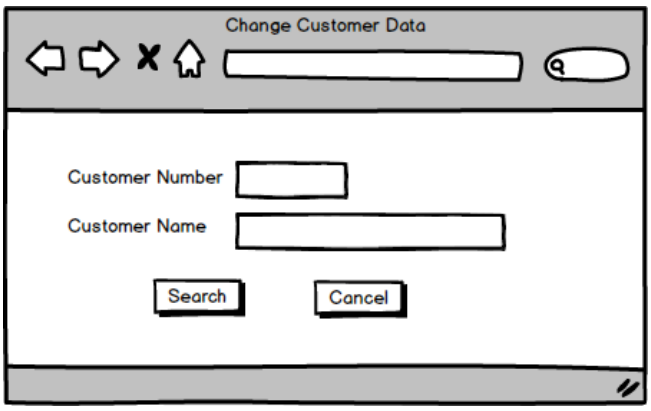
References to the standard

4.16, 4.17, 7.2.a, 7.2.n, 7.2.x, 7.3.d, 8.2.a, 8.2.c, 8.2.q, 8.2.u, 8.3.a, 8.3.b and 9.2.j

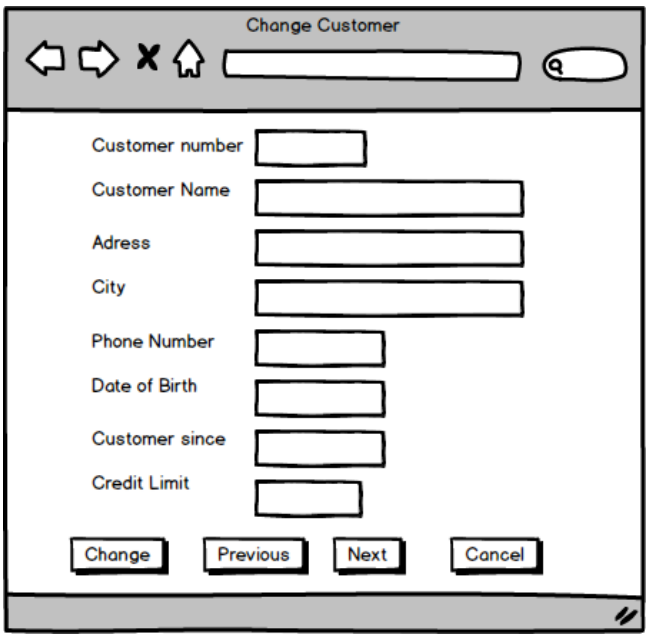
22.2 Selection via the change screen

Problem description

Using a menu, the user indicates that he wants to change customer data. The application subsequently presents screen 1 on which the user must enter a unique customer number or a customer name (but not both). The change function then has the screen sequence illustrated below.



After entering one of the two selection criteria, the data of the (first) customer that meets the criterion appears on screen 2. If Customer Number is used as the selection criterion, then only one customer can satisfy the criterion. Function keys are not active in such a case. If Customer Name is used as the selection criterion, however, a number of customers may satisfy the criterion. The user will not receive the kind of overview screen he did in the previous example in order to select a customer, but instead will be able to use the function keys PF2 and PF3 to browse through the customers selected until he has found the one he wants.



Are external outputs or external inquiries counted for this external input (i.e., for the ability to change customer data)? How many functions should be counted in total?

Discussion

If the correct customer data is displayed on the change screen (screen 2) as a result of a customer number having been entered on screen 1, then the display of the data to be changed is part of the external input *Change Customer*. The display is not counted as an individual external inquiry.

Analyzing must be carried out as follows when a user can enter a non-unique customer name on screen 1, browse through the customers on screen 2 until the correct one has been found, and then change the data of the customer. When the user searches for a customer via customer name, this search may result in the selection of a number of customers that can be displayed on the change screen via the function keys. This selection is not fully determined in size beforehand; i.e., the size of the selection varies depending on how many customers meet the selection criteria. The customer desired can then be selected. The display of the selected customers is seen as additional functionality. An additional function is identified because the functionality provided is in fact the same as in the previous illustration in which the selected customers were represented on an overview screen, and because the display and ability to browse through the selected customers entails a different logical processing than when data is merely presented and changed. More concretely, one external output should be identified because the output varies in size.

Sometimes it occurs that an application retrieves only one customer when a user selects data via customer name, in which case the function keys for browsing are not active. Such a situation is considered to be a part of the external output that is counted for selecting. The situation does not result in an additional external output or external inquiry.

The change function is the same in both cases and is therefore counted as one external input.

Solution

Count one external input for the change function.

Count one external output for the ability to browse through the customers that meet the selection criterion.

References to the standard

4.17, 7.2.a, 7.2.n, 7.2.x, 7.3.d, 8.2.a, 8.2.c, 8.2.q, 8.2.u, 8.3.a, 8.3.b and 9.2.j

23 DIRECT AND DELAYED PROCESSING

Problem description

An application provides its users the opportunity to update an insurance group for those insured on the basis of region; e.g., regional theft insurance premiums. The application enables its users to assign a zip code series to a different insurance group by means of a screen. In this illustration, three functionally different situations are handled. Each situation shows how function point analysis should be carried out.

The update in the first situation takes place immediately after a zip code series has been entered. Then a new series can be entered. When the user is finished with the function and one or more insurance groups have been adjusted, the application prints a report of the transactions for verification purposes.

In the second situation, the user can enter one or more zip code series. The processing of the data is done at night. When the insurance groups have been adjusted, the application prints a report of the transactions for verification purposes. Once the zip code series have been entered, they can no longer be maintained.

In the third situation, the user can enter one or more zip code series. The application processes the data at night, after which it prints a report of the transactions for verification purposes. Now, however, the zip code series entered can still be changed or deleted after they have been entered, but before their nightly processing.

For each of the situations above, determine which functions should be identified. Does the transaction file with the zip code series in situation 2 and/or 3 count as an internal logical file?

23.1 Direct processing

Discussion

The main objective of this function is the adjustment of the insurance groups. The data that is saved is functionally permanent data. This means that an external input is present. The creation of the transaction report is inextricably bound to the function, and the report itself fulfills a functional requirement; i.e., it is necessary for verification purposes. The transaction report also crosses the application boundary. For these reasons, an external output is identified for the transaction report, even though the function is inextricably bound to the external input.

Solution

Count the following functions:

- One external input for entering and adjusting the insurance groups
- One external output for the transaction report

References to the standard

7.2.r and 8.2.p

23.2 Delayed processing

Discussion

The main objective of this function is the adjustment of the insurance groups. The data saved is functionally permanent data. This means that at least one external input is present. FPA considers the entering of the zip code series and the nightly processing of the data as delayed processing. It sees the nightly processing and the entering of the zip code series as a whole.

The zip code series temporarily saved cannot be maintained and are not permanent because the data no longer exists after being processed during the nightly processing. In other words, the data is "consumed". The zip code series therefore form a temporary dataset that cannot be considered an internal logical file.

The creation of the transaction report is inextricably bound to the nightly processing, and the report itself fulfills a functional requirement; i.e., it is necessary for verification purposes. The transaction report also crosses the application boundary. For these reasons, the transaction report is identified as an external output, even though the function is inextricably bound to the external input.

Solution

Count the following functions:

- One external input for entering the insurance groups and for the nightly adjustment of the insurance groups
- One external output for the transaction report

References to the standard

5.2.f, 7.2.r and 8.2.p

23.3 Delayed processing and maintenance

Discussion

This main objective of this function is the adjustment of the insurance groups. The data that is saved is functionally permanent data. This means that at least one external input is present. FPA considers the entering of the zip code series and the nightly processing of the data as delayed processing. It sees the nightly processing and the entering of the zip code series as a whole.

The zip code series stored can be maintained and therefore make up an internal logical file. Furthermore, two maintenance functions are identified: one for changing zip code series and one for deleting them.

The creation of the transaction report is inextricably bound to the nightly processing, and the report itself fulfills a functional requirement; i.e., it is necessary for verification purposes. The transaction report also crosses the application boundary. For these reasons, the transaction report is identified as an external output, even though the function is inextricably bound to the external input.

Solution

Count the following functions:

- One external input for the initial input of zip code series and the nightly processing together.
- One internal logical file for the transaction file containing zip code series.
- Two external inputs for the changing and deleting of the zip code series from the transaction file.
- One external output for the transaction report

References to the standard

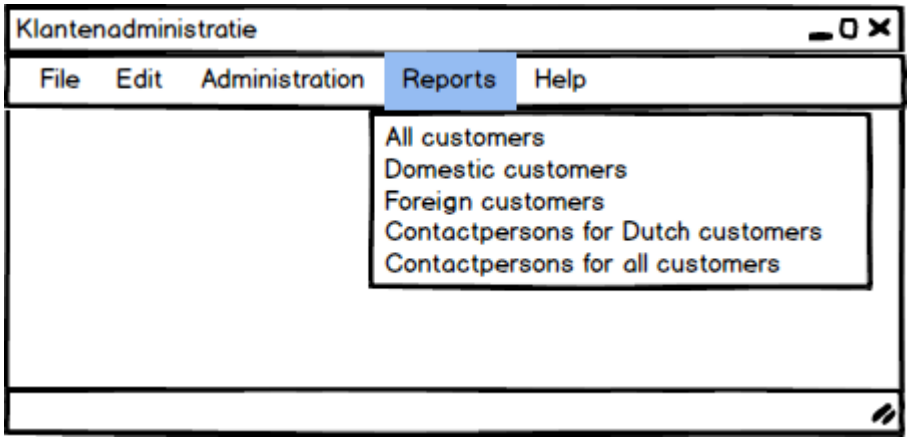
5.2.f, 7.2.r and 8.2.p

24 CASE STUDY CUSTOMER APPLICATION

Problem description

The functional specifications below have been made for a small customer application at an early stage of application development.

The following is maintained for each customer: Name, Address, City, Country Code, Telephone, and Contact Person. The registration numbers of Dutch customers registered at the chamber of commerce (CoC) are also maintained. Users would like to be able to add, change, and delete data. When a user wants to change and delete data, the customer data present must be shown for verification. Users also want to be able to print the following reports via the menu below:



A sketch of each of these reports is given on the following page. The name of a country is retrieved from a file called Countries that contains the name of a country for each country code. This file is maintained by a different application.

1. Report "All customers"

This report contains all customers and is ordered by company. The country for Dutch customers is not printed.

All customers

Business name	Country	Telephone	CoC-nr	Contact person
AeroDat	België	00-32-2-3456789		du Spiré
BankBetaal		030-3141592	12345	Westerhof
ImportRossia	Rusland	00-7-812-4567890		Ivanets
LuchtBelga	België	België		VandenBerghe
SehFern AG	Duitsland	00-49-30-1234567		Strohmann
TevreeConsult		020-7777777	45678	Doeven

2. Report "Domestic customers"

This report contains all Dutch customers ordered by company.

Domestic customers				
Business name	**Country**	**Telephone**	**CoC-nr**	**Contact person**
BankBetaal		030-3141592	12345	Westerhof
TevreeConsult		020-7777777	45678	Doeven

3. Report "Foreign customers"

This report contains all foreign customers. The user wants the country to appear at the beginning of each line on the list.

Foreign customers				
Business name	**Country**	**Telephone**	**CoC-nr**	**Contact person**
AeroDat	België	00-32-2-3456789		du Spiré
LuchtBelga	België	België		VandenBerghe
SehFern AG	Duitsland	00-49-30-1234567		Strohmann
ImportRossia	Rusland	00-7-812-4567890		Ivanets

4. Report "Contact persons for Dutch customers"

The report contains the telephone number and the contact person of all Dutch customers.

Contact persons for Dutch customers		
Business name	**Telephone**	**Contact person**
BankBetaal	030-3141592	Westerhof
TevreeConsult	020-7777777	Doeven

5. Report "Contact persons for all customers" (by country)

This report contains the telephone number, the chamber of commerce number, and the contact person of all customers. Customers are grouped by country.

Contact persons for all customers (by country)

Business name	Telephone	CoC nr.	Contact person
België			
AeroDat	00-32-2-3456789		du Spiré
LuchtBelga	00-32-81-7654		VandenBerghe
Duitsland			
SehFern AG	00-49-30-1234567		Strohmann
Nederland			
BankBetaal	030-3141592	12345	Westerhof
TevreeConsult	020-7777777	45678	Doeven
Rusland			
ImportRussia	00-7-812-4567890		Ivanets

For this application a high level function point analysis has to be carried out.

Discussion

The entity type *Customer* can be maintained in the application and is an internal logical file. *Country* is an FPA table that the application can only read. This is counted as a record type in the FPA tables ELF. Other FPA tables do not exist; therefore, the FPA tables ELF in this case consists of only one record type.

The specifications indicate that customer data can be added, changed, and deleted. This means that three external inputs are identified. The fact that a chamber of commerce number may not be entered for foreign customers does not play a role.

The user has not requested a separate external inquiry. The showing of current customer data for the purpose of verification when a user changes and deletes data is not counted as a separate external inquiry.

Reports 1, 2, and 3 together count as one external output because the following applies in all cases:

- The same object is being reported on (customer)
- The selection criterion is the same (country)
- The processing in order to produce the output products is the same (Except for the selection mechanism, no additional processing is needed.)
- The logical layout of the output products (set of data element types and their structure) is the same; i.e., business name + (country) + telephone + (CoC-nr) + contact person. The parentheses denote optionality. The sequence is not important.

It is irrelevant that a heading is not printed in all cases, as when data is not present or desired; e.g., a CoC-nr or the name of a country, respectively. The headings, after all, have been defined for the output product. Although the sequence of the columns is different in report 3, this is no reason to identify a separate external output. In these three cases, a direct selection takes place via the heading Country.

The same result could also be realized with a fill-in screen in which the user is provided with country code as a selection criterion. The fill-in screen would not be counted as a separate external input. Within FPA, the data to be filled in would be considered control information for the external output, and each piece of data would be included in the analysis as a data element type.

While it is true that report 4 selects the same customers as report 2, the logical layout is different because the set of data element types in report 4 is different: business name + telephone + contact person. Report 4 therefore counts as a separate external output.

Report 5 selects the same customers as report 3. The set of data element types is the same in both reports. However, the structure of the output product is different (the data element types are grouped differently) because the country is presented once each time. Therefore the logical layout is different. As a result, report 5 is identified as a separate external output.

According to the guidelines, no transactional functions are identified at all for the FPA tables ELF, even if external inquiries or external outputs would be present.

Solution

A logical file is counted as low in a high level function point analysis and a transaction as average. This results in the following functional size:

Function	Type	Complexity	Function points	Comments
Customer	ILF	Low	7	
FPA-tables-ELF	ELF	Low	5	
Add customer	EI	Average	4	
Modify customer	EI	Average	4	
Delete customer	EI	Average	4	
Report 1	EO	Average	5	
Report 2	-	-	-	Is the same as report 1 in FPA
Report 3	-	-	-	Is the same as report 1 in FPA
Report 4	EO	Average	5	
Report 5	EO	Average	5	
Menu	-	-	-	Is not counted
TOTAL			39	

The functional size of the application is 39 function points.

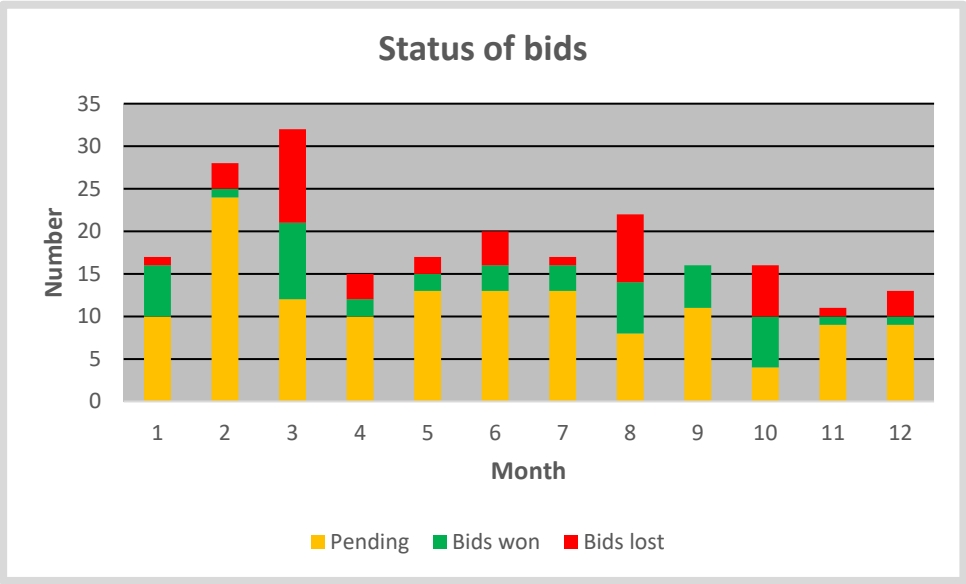
References to the standard

3.2.2, 4.20, 6.2.g, 7.2.m, 7.2.n, 8.1, 8.2.w and 8.3.b

25 GRAPHS

Problem description

A commercial management information system displays in a graph the results of outstanding bids per month according to the following example.



Which user transaction(s) are identified and of which type? Which data element types are counted? How is the complexity determined?

Discussion

As the guideline states, graphs (just as reports) can be considered output. In addition in this example there is a recurring group of data (bids).

Solution

Count an external output because there is a recurring group of data.

For the complexity:

- count three data-element-types: status bid, number, month.
- count one logical data file: bid.

So the complexity is Low.

Reference to the standard

4.12

26 IDENTIFYING DATA ELEMENT TYPES

In this example, two situations are presented indicating which data element types are identified.

26.1 Identifying process data

Problem description

An external output shows the sales price (including VAT) and the VAT amount on a product screen. The data file ARTICLE contains, per item, the sales price excluding VAT and the VAT code. The current VAT amount is determined by the VAT code, VAT entry date, VAT rate.

Which of these data count(s) for the determination of the complexity of the external output?

Discussion

In fact the question is: should the data elements underlying the calculated value and / or the calculated value itself (i.e.: VAT code, VAT entry date, VAT rate and / or VAT amount) be counted as DET?

The calculation of the DETs (VAT code, VAT entry date, VAT rate) should not be counted as DET of the transaction. Only data that cross the boundary of the application i.e. sales price and VAT amount are counted as DET of the transaction.

Solution

Just count the DETs that cross the boundary of the information system. In other words: sales price, VAT amount and initiation trigger.

References to the standard

4.23, 8.3.d and 8.3.e.

26.2 Data element types within a data file

Problem description

Given a data file containing, among other things, the DETs Initials, Prefixes and Last Name. How should a field on a screen be displayed for a user transaction, showing <Surname> + ', ' + <initials> + ' ' + <prefixes> (e.g., 'Graaf, R. de')

Which DETs should be counted in the user transaction?

Discussion

The question is: should the entire field be counted as 1 DET, or should the three fields (3 DETs) be counted from the data file?

Solution

If for the users of the application only the full name has a meaning, then it should be counted as one DET. This also applies if the fields "initials", "prefix" and "last name" were technically stored as three fields.

Reference to the standard

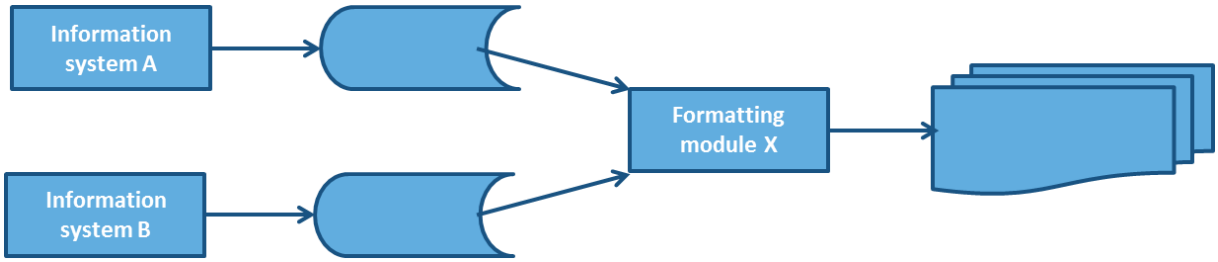
4.23

27 GENERIC FORMATTING SYSTEM

27.1 Technically generic

Problem description

A company uses multiple back-office information systems (A, B, etc.) for performing different statutory duties. Out of each information system, regularly large quantities of letters are created. These letters are described in the designs of the back-office information systems. The IT department has decided to resolve the formatting function and the sending of letters generically and to put it in a separate formatting module, see the figure below.



For each letter, the fixed letter texts are stored in the formatting module X. The format is defined in a number of design patterns. Also parts like the salutation and the signature are included therein. The back-office information systems (A and B) supply all variable data in a transfer file. This includes the address information.

The question is, how are the letters counted? Do we identify multiple user functions per letter or do we identify only one EO per letter, and to which information system does this function belong?

Discussion

Each letter that is created from a back-office information system counts as an EO. Upon a change of a letter in information system A, a number of situations may occur:

1. The content of one or more variables used in the letter is calculated differently. This means a change in the EO in information system A. The formatting module X does not need to be modified, because the variables already exist, only the content is determined differently.
2. New data (variable) must be added to the letter. In this case the EO must be modified in information system A, but also in the formatting module X a modification has to be made in order to read the new data and place it on the letter.
3. There is a modification to the fixed text of the letter. For this change, only formatting module X needs to be modified. The variables, or the content are not referenced, so the EO in information system A does not change.

In addition, there can also be additional modifications introduced to the design patterns (layout, corporate identity, logo, font, etc.), which apply to all letters. In these cases, only formatting module X is affected.

Every type of letter that exists in the back-office information systems thus has a counterpart in the formatting module X. Does this mean that every EO in the back office information system is followed by an EI and EO in formatting module X? The texts and design patterns in module X are maintained by specialized programmers. Therefore, extra effort is to be delivered.

From the end-user perspective, creating a letter is one output process. The end user does not know that there is a separate formatting module where all letters are collected to be formatted and sent. From this perspective, there is one EO per letter. Any modification of the letter means a modification of this EO, regardless whether the modifications must be applied to information system A, formatting module X, or both. In the project planning both types of effort must be estimated.

Solution

The end-user has not explicitly asked for formatting module X, therefore this is regarded as a technical solution. Count 1 EO per letter. When a generic modification is made, count as many EO as the number of changed letters.

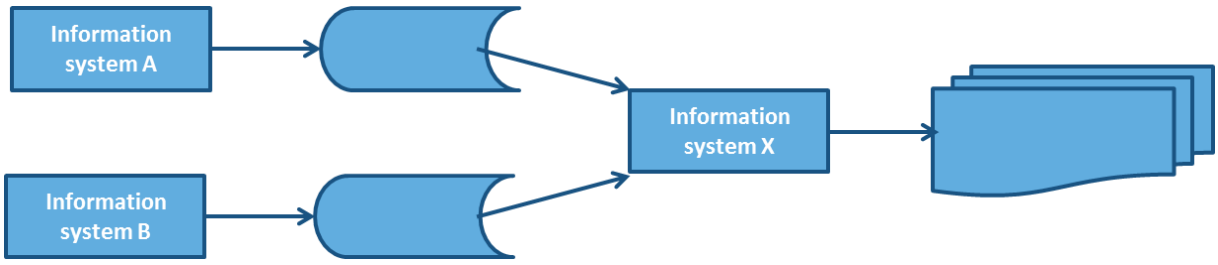
References to the standard

3.5.1, 4.1 and 4.22

27.2 Functionally generic

Problem description

The same situation as above, but now the end-user explicitly asks for a solution where the formatting of letters can be maintained in a simple and generic way. A functional design is drafted for the recognized information system X with the purpose of easy maintenance of letter formatting.



Discussion

Count 1 EO per letter for the back-office information systems.

Perform a function point analysis on the functional design of information system X. Assuming that the history of letters is recorded, count at least 1 EI and 1 EO, plus all maintenance functions to be realized.

In case of modifications, count all changed functions.

Solution

Count one external output per letter for the back-office information systems.

Perform a function point analysis on the functional design of information system X. Assuming that the history of letters is recorded, count at least one external input and one external output, plus all maintenance functions to be realized.

In case of modifications, count all changed functions.

References to the standard

3.5.1, 4.1 and 4.22

27.3 Comments on the two variants

The choice to use a formatting module to maintain letters in variant 1 is made by the IT department in order to carry out the maintenance more profitable. Initially a separate formatting system must be purchased, but changes that affect all letters concern can be applied simpler and that is how the investment is recouped.

In variant 2, the choice to use a formatting module to maintain letters, is made by the end-user. Here the reason was also here that the investment would be recouped in the maintenance of the letters.

In terms of function points: in variant 1 less function points are counted for the construction project than in variant 2. When modifications to the letters occur, in variant 1 more function points are counted than in variant 2.

In terms of productivity: in variant 1, relatively more hours are required per function point for the construction, and less in maintenance. In variant 2, the number of hours per function point is relatively normal, and also during maintenance.

The above shows the effect of the choice that can be made on the boundaries of an information system. The same (technical) solutions produce a different result for function point analyses.

28 IDENTIFYING ILF

Problem description

A municipal tax administration system has a collection module in which payments are recorded. These payments relate to real-estate tax, waste tax, sewage charges and tourist taxes. The payments received are stored in four tables whose structure is documented below.

PAYMENT_REAL-ESTATE-TAX	PAYMENT_WASTE-TAX	PAYMENT_SEWAGE-CHARGES	PAYMENT_TOURIST-TAX
PaymentNumber	PaymentNumber	PaymentNumber	PaymentNumber
Amount	Amount	Amount	Amount
FiscalYear	FiscalYear	FiscalYear	FiscalYear
SocialSecurityNumber	SocialSecurityNumber	SocialSecurityNumber	
ChamberOfCommerceNr			ChamberOfCommerceNr
PaymentDate	PaymentDate	PaymentDate	PaymentDate
AssessmentNumber			
	TaxationNumber	TaxationNumber	
			ClaimNumber
AccountNumber	AccountNumber	AccountNumber	AccountNumber
PeriodSerialNumber	PeriodSerialNumber	PeriodSerialNumber	PeriodSerialNumber
PaymentReference	PaymentReference	PaymentReference	PaymentReference
IndicationCollectionPayment	IndicationCollectionPayment	IndicationCollectionPayment	

All cases are about bank transfer payments. There is no essential difference between the assessment number, taxation number and claim number; these are sheer fiscally-legal designations. The system determines, based on the payment reference of a payment received (in which an indication of the tax category is included) in which of the four entities the concerned payment is booked. The procedure for all payments received is the same: the entries are collected daily in an output file for the general ledger system. The collection module provides the same functionality to monitor the status of payments for all types of payments.

How many ILF should be identified in this situation?

Discussion

The definition of an ILF is:

An internal logical file is *a logical group of permanent data seen from the perspective of the user* that meets each of the following criteria:

- It is *used* by the application to be counted
- It is *maintained* by the application to be counted

The aforementioned entities meet the listed criteria: they are both used and maintained by the information system to be counted. However, the phrase, "from the perspective of the user" is crucial. The guideline states the following on this subject:

"... a group of data that an experienced user considers as a significant and useful unit or object. An equivalent to this kind of logical group of data is an object type in data modeling."

For users, the four different types of taxes are also four different things. Taxes and the accompanying regulations are indeed mutually substantively totally different. The designer of the above model has wanted to connect to the different taxes and opted to accommodate four types of payments in four separate tables; these are largely similar in structure. Logically, however, they all concern one object type *PAYMENT*, of which one of the data element types could be: TaxType. For the valuation of the logical file the additional data element type will be counted and no additional record types are distinguished.

In a situation like this it is recommended to adjust the functional design in order to avoid discrepancy between the count and the design.

Solution

Count one ILF with 1 RET and 14 DET's.

Reference to the standard

5.1

29 STUBS AND DRIVERS

Problem description

For the purpose of testing systems or parts of systems, often so-called stubs and drivers are built. A stub is a simulation program that replaces a program, including the associated input and output streams, and is called upon by the test object. A driver is a simulation program that replaces a program that provides the control or call to the test object.

Should the stubs and drivers, developed by a project team be considered in a function point analysis?

Discussion

Development effort is involved to build stubs and drivers. They are not a part of the product at delivery and thus in any case do not belong to the product size, similar to conversion software for example. The question remains whether they, like for example conversion software, can be considered as a part of the project size.

Conversion software is, even if used only once, and adding no functionality to the system developed, delivered to the client as a project result.

Stubs and drivers are testing tools that are used during the project and potentially are transferred to application maintenance. Stubs and drivers are tools that must be localized entirely within the domain of system development as well as all other provisions that should be taken during the project to bring the project to a successful conclusion. They are no features that are recognized by the user and they are meaningless from the user's perspective.

Solution

Stubs and drivers should neither be included in the product size, nor in the project size.[1]

References to the standard

3.6.1, 3.6.2, 7.1, 8.1 and 9.1

[1] This solution only applies under the assumed conditions in this example, where there is no explicit requirement by the end-user to deliver stubs and drivers to the production environment. If this had been the case, the stubs and drivers would become part of the product functional size.

30 SAME FORMAT, DIFFERENT PROCESSING

Problem description

An insurance company has 3 divisions: Life, Property and Health. At group level of this insurance company a report is drawn up for each division with the risk profile of the division. See the example below:

Report risc profiles dd-mm-jjj

	Risc profile care	**Damage and income**	**Life and pensions**
Risc year:	yyyy	yyyy	yyyy
Premiums received:	€ 999.999.999	€ 999.999.999	€ 999.999.999
Payments:	€ 999.999.999	€ 999.999.999	€ 999.999.999
SCR:	999,999%	999,999%	999,999%
MCR:	999,999%	999,999%	999,999%

- The report is intended both for the National Bank that requires an understanding of the risks of the insurance company on the basis of its audit function, and also for use in the internal management of each division.
- The 3 parts of the report are identical in layout and data they provide, they contain for each division data that together display the risk profile.
- The data on which the different risk profiles are compiled come from different policy and claim administrations of the 3 divisions. The logical processing to get the data on the report vary by division.

The question is how much external outputs are counted.

Discussion

Apparently this is 1 report. However:

- The logical processing varies by division, there are three elementary functions.
- The data of the three columns can be used independently per division.

Solution

Count three external outputs.

Reference to the standard

8.1

31 STARTING AND STOPPING OF BATCH FUNCTIONS

Problem description

An information system comprises of a number of different batch processes, each of which may be considered as a basic function. Key data of a process is identification, process name, start date / time, end date / time and process status.

A process can be started and stopped by an administrator. The administrator can also retrieve a list of running processes, interrupt running processes and restart interrupted processes again.

How do we count this functionality offered to the administrator?

Discussion

The administrator is a user within the meaning of FPA. If the administrator has requested this functionality, it should be counted. Based on the above, an internal logical file "process" can be recognized.

The requested functions are starting, stopping, pausing and restarting a process. We assume that the retrieval process serves not only to pause and restart processes, but also meets an information need of the administrator to see which processes are running. So an external output must also be counted.

Solution

Count the following:

1 ILF Process

1 EO Overview processes

4 EI Start process, Stop process, Pause process and Restart process

Reference to the standard

2.6

32 CODE AND DESCRIPTION

Problem description

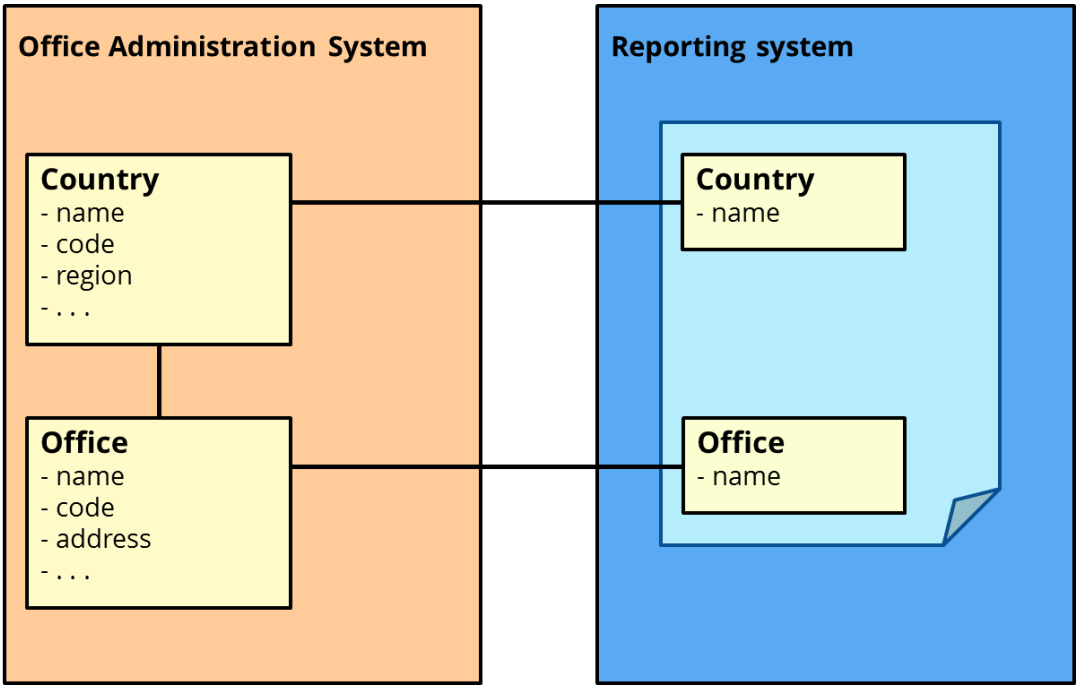
In an Office Administration System, data on offices of an organization is saved and maintained.

- Regarding the countries, the following data is recorded: name of the country, country code, area code, date of entry, date of termination
- For the office, the following data is recorded: name of the office, office code, address, location, phone number, date of entry, date of termination

In a Reporting system reports are put together made from various information systems.

The Office Administration System generates a summary with offices per country.

See figure below.



How many ILFs are identified for the summery generated by the Reporting system?

Is one ELF *Office* identified, are two ELFs *Country* and *Office* identified, or is an FPA tables ELF identified?

Discussion

The main question is how you deal with data being an ILF in one information system (the Office Administration System in this example), that is used in another system.

Country and *Office* are both an ILF in the Office Administration System.

For the Reporting system however only the names are of interest, all other fields have no meaning for this Reporting system. Seen from the Reporting system one could conclude that one ELF *Office* exists, where country name is an attribute of *Office*.

Solution

If during the function point analysis of the Reporting system it is known how the information is used in the Office Administration System, we know that both *Country* and *Office* are an ILF in the Office Administration System. In this case they must be counted twice as an ELF for the Reporting system.

If during the function point analysis of the Reporting system it is not known which type of functions *Country* and *Office* are in the Office Administration System, the function point analyst should make an assumption. In this example an ELF *Office* is the most obvious choice, but it is also possible to identify an FPA table ELF for *Country*.

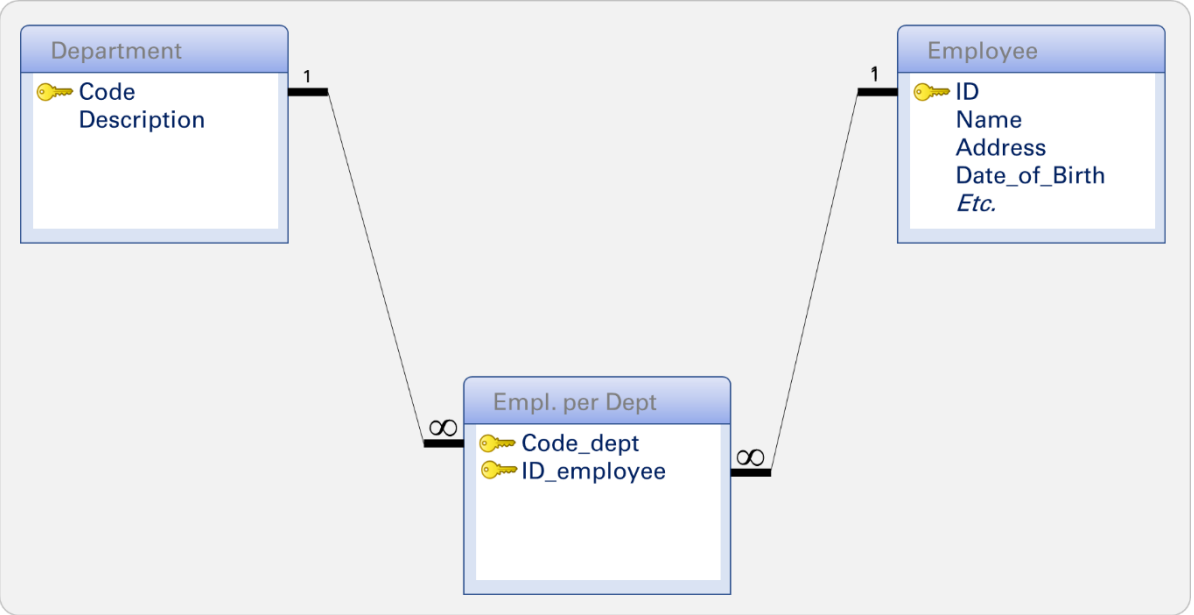
References to the standard

4.20, 6.1 and 6.2.g

33 FPA-TABLE WITH N:M-RELATION

Problem description

Within an application there is an entity *Department* with the characteristics of a FPA-table with an N:M-relation with another entity *Employee*. This relation is in the third normal-form in a 'key-key-entity' *Employee per Department.* If an *Employee* is deleted, then all the linked departments are deleted from *Employee per Department.*



How many ILFs should be identified in this situation?

Discussion

The following denormalization rules are described (see section 4.21.2):

1. Determine which entity types in the data model are FPA tables. FPA tables are valued in a specific way. See section 4.20 and guidelines 5.2.k and 6.2.g.
2. Determine which entity types are a "key-key entity" without other attributes. These represent an n:m relationship in the normalized data model and are not valued at all. The referring attribute (foreign key) is identified as a data element type for both logical files connected by this key-key entity.
3. Determine which entity types are a "key-key entity" *with* other attributes. Note that two situations can arise here as a result:
    a. The additional attributes are technical by nature (not requested by the user; e.g., a date/time stamp) are not identified as data element types. If they are the only data element types, then the entity type should be dealt with as indicated in step 2 above.
    b. The additional attributes are functional in nature (required by the user), in which case, they should be treated as indicated in step 4.

4. Examine the remaining entity types as to whether they are a logical file on their own or whether together, with one or more related entity types, they make up a logical file. Determining factors are:
   - The nature of the relationship(s) with another entity type (cardinality and optionality)
   - The dependence or independence of the entity type's existence

   Both of these ideas are examined further below. See sections 4.21.3 and 4.21.4.

After the nature of the relationship(s) has been determined, you can assess how the entity types involved should be considered using the table in section 4.21.5. In a data model as described here two denormalization rules come together with another outcome if carried out in a different order.

Step 1 of the denormalization rules describes that the FPA-tables must be identified. In this example the entity *Department* is a FPA-table, according to the rules of section 4.20.

Step 2 of the denormalization rules describes the determination of the 'key-key-entities' without other attributes. These are not valued at all. The referring attribute (foreign key) is counted as a data-element-type for both logical files connected by this key-key entity.

In this example the entity *Employee per Department* is a key-key entity without additional attributes. The attribute *Department-Code* is to be added as a reference attribute to the entity *Employee* and the attribute *ID-Employee* is to be added as a reference attribute to the entity *Department*. By addition of this attribute the entity *Department* does not meet the criteria of a FPA table.

However if the logical files are determined based on a third normal form data model, the denormalization rules (section 4.21.2) must be applied in the exact order as described.

Notwithstanding all the other rules in the manual section 4.21.2 is only applied to an entity in third-normal form. In this example it means that in the first step the entity *Department* is earmarked as a FPA-table. And therefore only the entities Employee per department and Employee go through the next steps.

For both the entities (*Employee per department* and *Employee*) the steps 2 and 3 are not relevant. Step 4 is important. Examine if both the entities are separate logical files. The nature of the relationship (section 4.21.3) and the dependence (section 4.21.4) are key for this determination. All possibilities are mentioned in section 4.21.5.

Solution

Count the entity *Department* as a FPA-table. It will become a RET within the FPA-tables-ILF.

Count the entities *Employee* and *Employee per department* together as one Internal Logical file (ILF). There is entity dependence because *Employee per department* and *Employee* are deleted in the same action.

References to the standard

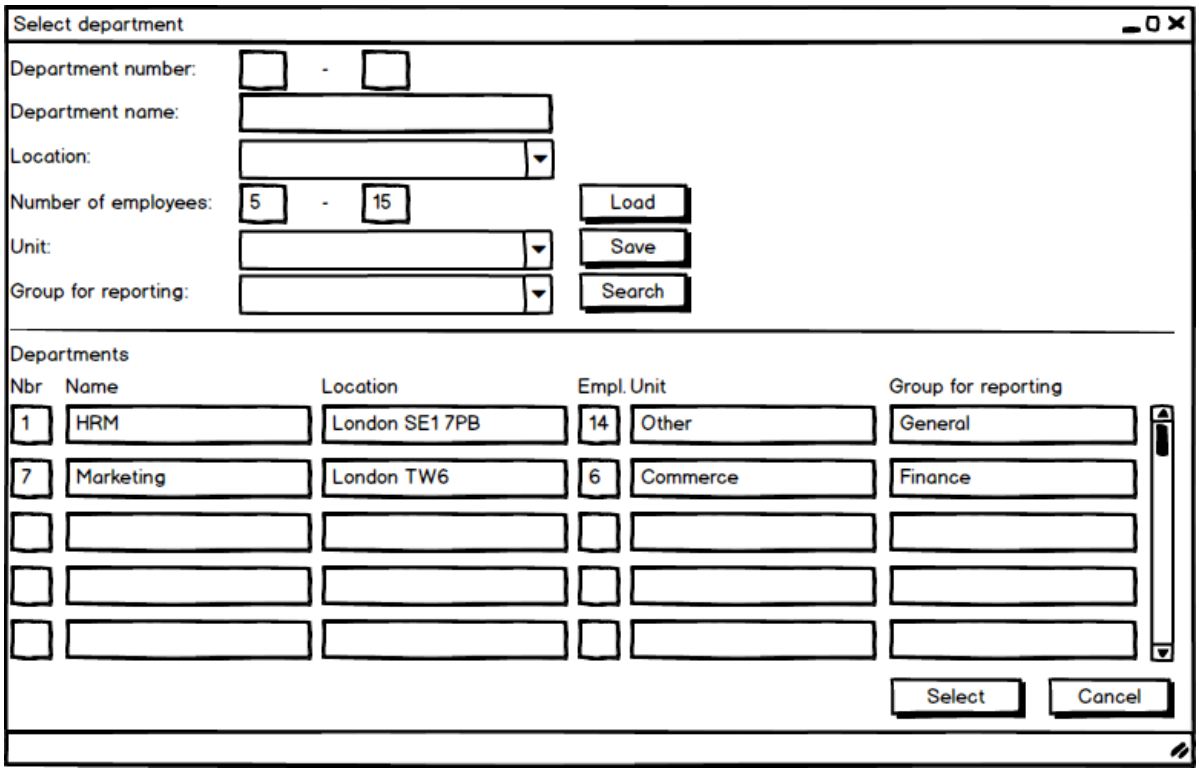
4.20, 4.21, 5.2.k and 6.2.g

34 SAVING OF SELECTION CRITERIA

Problem description

The employee registration system has a search function for selecting departments (see screenshot below). This search function enables the user to locate departments through one or more search criteria.

On the user's request a search screen is added with the ability for the user to save entered selection criteria and to load them. This is because certain selection criteria are commonly used. In this way a user can save one set of personal selection criteria. In the data model an entity exists which defines these selection criteria per user.



How should this function for selection criteria be counted (leave aside the list boxes)?

Discussion

The first question is whether the storage of selection criteria is to be counted separately, or should be seen as a part of the output function and therefore should not be counted.

Because the user has asked for this functionality, it is to be counted separately.

Each user can define his or her own selection criteria, so the data may contain more than one occurrence. So count an ILF (another possibility is a RET in an ILF User).

Saving of the selection criteria is counted as an external input.

The question is whether the retrieval of stored selection criteria should be seen as part of the output function, or as separate external inquiry (there's certainly no question of an external output because there is no list from which the user can choose).

The loading of previously saved selection criteria makes it easier for the user to enter selection criteria in order to generate an overview of departments. The user has specifically requested for this functionality. On the screen there is a choice to use the button *Load* or manually enter the selection criteria. That's why we count an external inquiry for showing the previous selection criteria.

Solution

Count one ILF for the selection criteria, if existing independently, an external input for storing the selection criteria and an external inquiry for displaying the selection criteria.

References to the standard

8.2.e, 8.2.t and 9.2.d

35 MASTER-DETAIL SCREENS

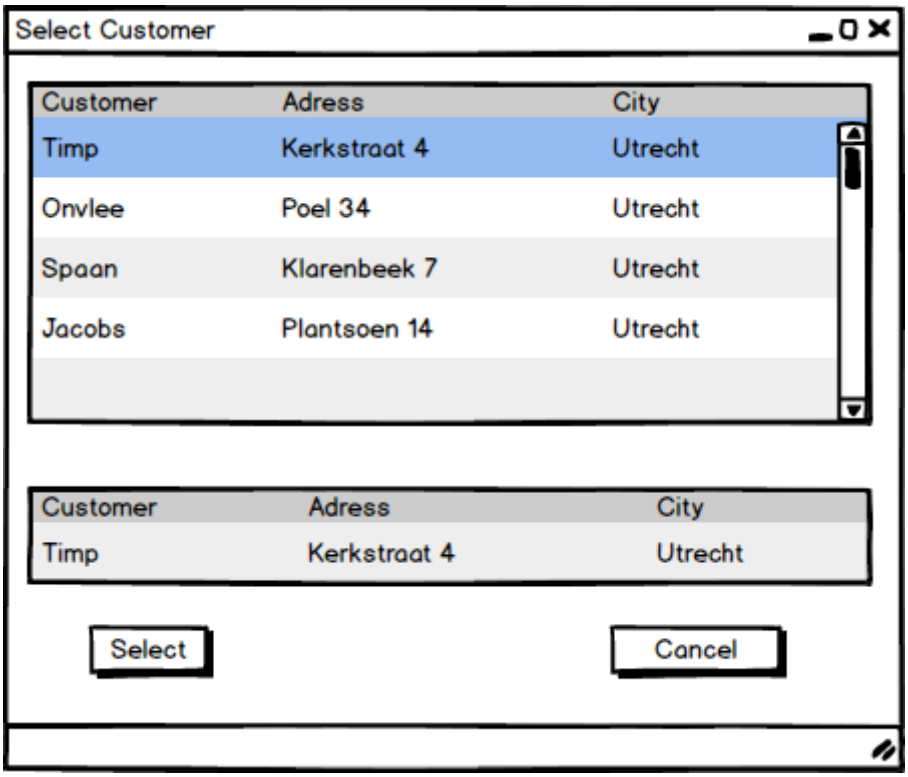
Problem description

Many applications contain so-called master-detail screens. Screens where in the top (the master section), a window is displayed, in which the main identifying data of occurrences of particular data are displayed and where one can select a single occurrence. At the bottom of these screens (the detail section) the attributes of the selected occurrence are displayed.

Function Point Analysis hereby recognizes two transactional function types (maintenance options are ignored in this example):

- a function for displaying the identifying data of all occurrences for the purpose of making a selection (an external output)
- a function for displaying the details of the selected occurrence (an external inquiry)

Within a given application there are master-detail screens where in the detail section exactly the same attributes are displayed for an occurrence as in the master section.



Is one EO and one EQ counted, or only one EO?

Discussion

The discussion is whether you must recognize for this particular case, in addition to an external output for the master section also an external inquiry for the detail section or not.

In terms of information to the user the detail section adds nothing, but the screen is described in detail in the design and the user has requested this detail section with the same attributes.

The master section is an EO, because multiple occurrences are displayed in order to make a further selection from there.

The detail screen is an EQ, in spite of the fact that the master section and the detail screen display exactly the same DET's. This is because the design explicitly asked for this and the format (selection data) differs from the 'master section EO.

Solution

Count one external output and one external inquiry.

References to the standard

8.1 and 9.1

36 LISTS WITHIN LISTS

Problem description

A process exists that shows an overview of customer data for audit purposes. The overview has the following lay-out:

Company J. Paulson
Mainstreet 18
12345 Village
Netherlands

Tel.number: 046-1234567
E-mail: jpaulson@paulson.com

Oders

Order date:	01-05-2015			
1111054	Ballpens Basic	10	€ 10,79	€ 107,90
3125003	White copy paper 500 sheets	20	€ 19,49	€ 389,80
.....				
Order date:	06-05-2015			
3581487	Folder A4 4 rings 5p.	12	€ 17,99	€ 215,88
3142122	Window envelopes 500p.	15	€ 13,93	€ 208,95
Order date:	08-05-2015			
3563005	Dossier folders 10 p	50	€ 05,95	€ 297,50
3125003	Suspension files 20 p.	20	€ 19,49	€ 389,80

Outstanding invoices

2015-1004	31-03-2015	€ 1020,56
2015-1053	30-04-2015	€ 1533,50
....		

Company J. Petersen
....

Tel.number: 057-7654321
E-mail: jpetersen@petersen.com

{Etcetera. Identical lists per customer. For all customers.}

How many external outputs should be counted?

Discussion

Rule 8.2.g states that an output product can comprise several external outputs. That is the case when:

- the output product contains different logical layouts and these logical layouts can be retrieved individually, or
- the output product contains different logical layouts that have been established by different logical ways of processing and are combined for ease-of-use.

The rule also states the following on individual logical processes: “when the different parts report about a different object or when they come about as a result of other logical files.” In this case there is no question of individual retrieval. So the first condition does not apply.

It may be different for the second condition. The part of the list about orders comes from a different logical file than the part of the list about invoices. This would lead to the counting of two external outputs, one report on orders and one on invoices.

In the decision to count one or two external outputs, one should consider that the second clause contains the condition: “and are combined for ease-of-use”. Rule 8.2.g is not a license to split up output products that are meant as one single product purely because the different parts of the output product are retrieved from different logical files. The second condition of rule 8.2.g concerns a situation in which output products that should be considered as elementary functions are presented onto one output product (paper or screen) for practical purposes.

In this example the overview is meant to get an overview of the (debt)position of a customer for audit purposes: this amount of orders, this amount of outstanding invoices. That is one unambiguous purpose. The point of view is the customer for all parts of the overview. The fact that multiple logical files or objects must be addressed to describe the customer position does not lead to the conclusion that there are multiple external outputs for that reason.

Solution

Count one external output.

Reference to the standard

8.2.g

37 COUNTING TRANSACTION FILES

Problem description

System A produces a transaction file for system B. This file contains several record types (with different logical layouts), which cannot be retrieved individually.

Should this be counted as one or more external outputs?

Discussion

The key question is whether rule 8.2.g. or rule 8.2.j. must be applied.

Rule 8.2.g states that one output product can comprise several external outputs. This is the case when:

1. there are several different logical layouts which can be retrieved individually, or
2. there are several different logical layouts) that have been established by different logical ways of processing and are combined for ease-of-use.

This rule is not applicable here. By definition transaction files cannot be retrieved individually and ease-of-use is not in question here.

There is one transaction file here. Rule 8.2.j. relates to this situation. According to this rule there are different external outputs if the logical layout or the processing is different. This corresponds to the situation described in the problem description.

Solution

In this situation apply rule 8.2.j. So there are several external outputs.

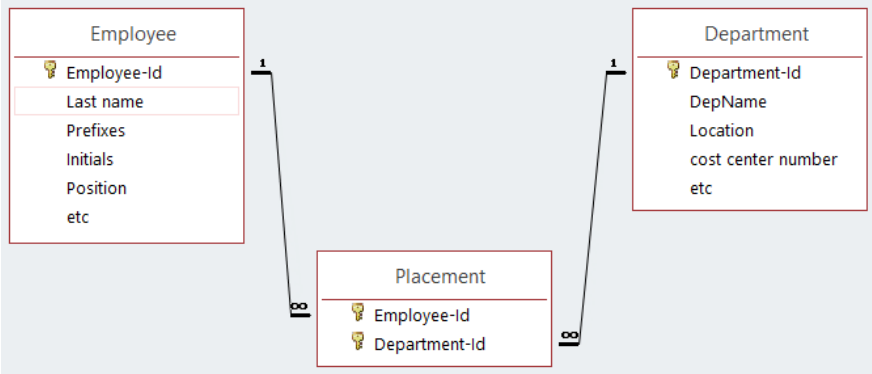
Reference to the standard

8.2.g and 8.2.j

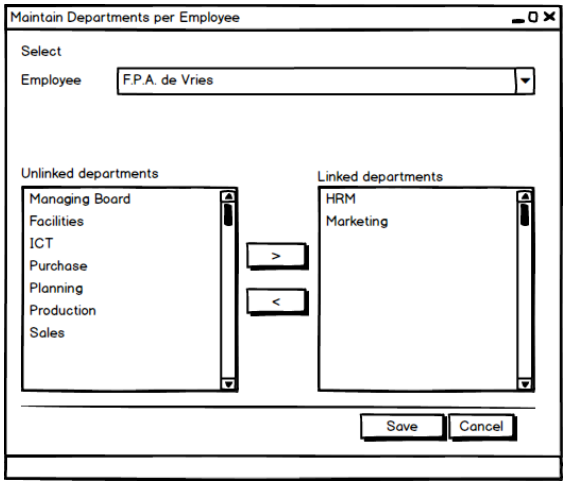
38 MAINTAINING AN N:M-RELATION

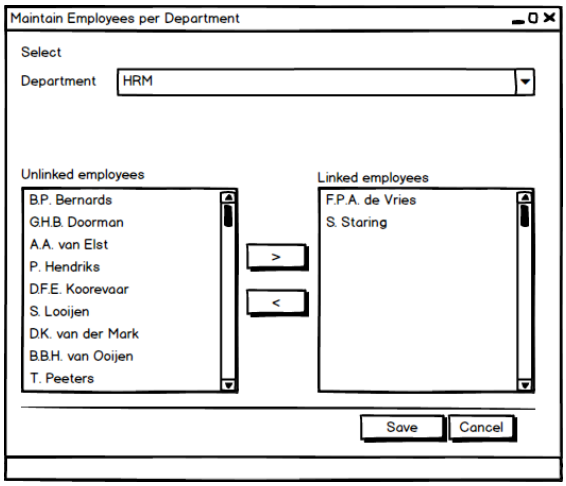
Problem description

In a business administration system employee and department data can be maintained. It is also possible to link departments to an employee and link employees to a department. For this purpose, a relationship entity (Placement) has been added, which contains only the keys of the two entities, resulting in an N:M relationship. In the data model of this business administration system the following entities are present:



The possibility to link one or more departments to an employee or break this link is shown in the image below, left. The possibility to link one or more employees to a department or break this link is shown in the image below, right.





The function works as follows: The left list shows the names of departments not linked to the selected employee, the right list shows those that are linked to this employee. The user can move department names from one list to the other and vice versa, using the buttons (“>”, “<”) between the lists. These movements can be persisted in the relationship entity using the Save button.

By the way, this is only one example of an implementation of this function.

The question at hand here is, whether external inputs should be identified? And, if so, how many.

The functionality of displaying the lists of departments and employees and the logical files concerned, are not elaborated here. The discussion of this example is limited to the functionality of the linking.

Discussion

A user story for Maintaining Departments per Employee could be:

"As an HR worker I want to be able to link a department to an employee and to break such a link, so I can show for every employee the department(s) they are working for."

A similar user story will exist for Maintaining Employees per Department.

In fact this function adds occurrences to and removes occurrences from the relationship entity Placement.

In the Nesma counting guideline is a section about denormalization (4.21.2 point 2), stating that a relationship entity is not valued at all.

Following this section the data model shown on the previous page results in two ILFs (Department and Employee). Every entity gets an extra data element type (DET) to implement the N:M relation.

Section 7.1 defines an External Input Function. Summarized this is: unique, user recognized, data entered from outside the application, elementary and add, change or delete data in or from one or more ILFs.

Combining these two sections (4.21.2 point 2 and 7.1) excludes identifying one or more EIFs to maintain the relationship entity Placement as this is not a valid ILF.

The entities Department and Employee are valid ILFs. According to section 4.21.2 point 2 these are the ILFs maintained by the function Maintain Departments per Employee.

After denormalization we have the ILFs:

Department	(Department-ID, *(Employee-ID_1, Employee-ID_n)*, Name, Location, cost center number, etc.)
Employee	(Employee-ID, *(Department-ID_1, Department-ID_n)*, Last name, Prefixes, Initials, etc.)

In *italics* the both ways added attributes. These are repeating groups. They don't have a meaning without the ILF they are part of. For FPA a repeating group is a record element type (RET). Such a RET can be empty (holds no elements).
When linking a department to an employee both an occurrence in the RET in Employee (reference to the department) and the RET in Department (reference to the employee) have to be added.

When breaking such a link the opposite goes and two occurrences (one from the RET of every entity) have to be removed.

So these are two different EIs that always change occurrences in both ILFs.

For linking/delinking of an employee to/from a department, the same reasoning applies.

Solution

Count for every screen 2 EIs (linking and delinking), so in total 4 EIs.

Reference to the standard

4.21.2 *point 2* and 7.1

39 TABLE WITH CODE, NAME, START- AND END-DATE

Problem description

For the registration of used colors for articles, a system has a table containing the following information:

- Color number
- Color name
- Start date
- End date

The table is maintained through a screen in which the user can add, delete, or modify new colors.

How is this table classified?

Discussion

There are two possibilities:

1. The entity is recognized as an FPA-table
2. The entity is recognized as internal or external Logical File (possibly with other entity types)

The table with color number, color name, start and end date does not appear to fall under the FPA-tables because the attributes are not similar. This would give rise to the table as Logical File (possibly along with other entity types) in this case.

However, the question that needs to be answered is: are the start- and end-dates technical in nature (for example, a time-stamp) or are these functional in nature (for example, colors can be changed in the future).

Solution

If the addition of the start and end dates are purely technical in nature, the table will then be valued as an FPA-table (as part of the FPA-tables-ILF).

If the addition of the start and end dates is functional, the table should be valued as internal or external Logical File (or as part of). In this case, the maintenance functions must also be valued separately.

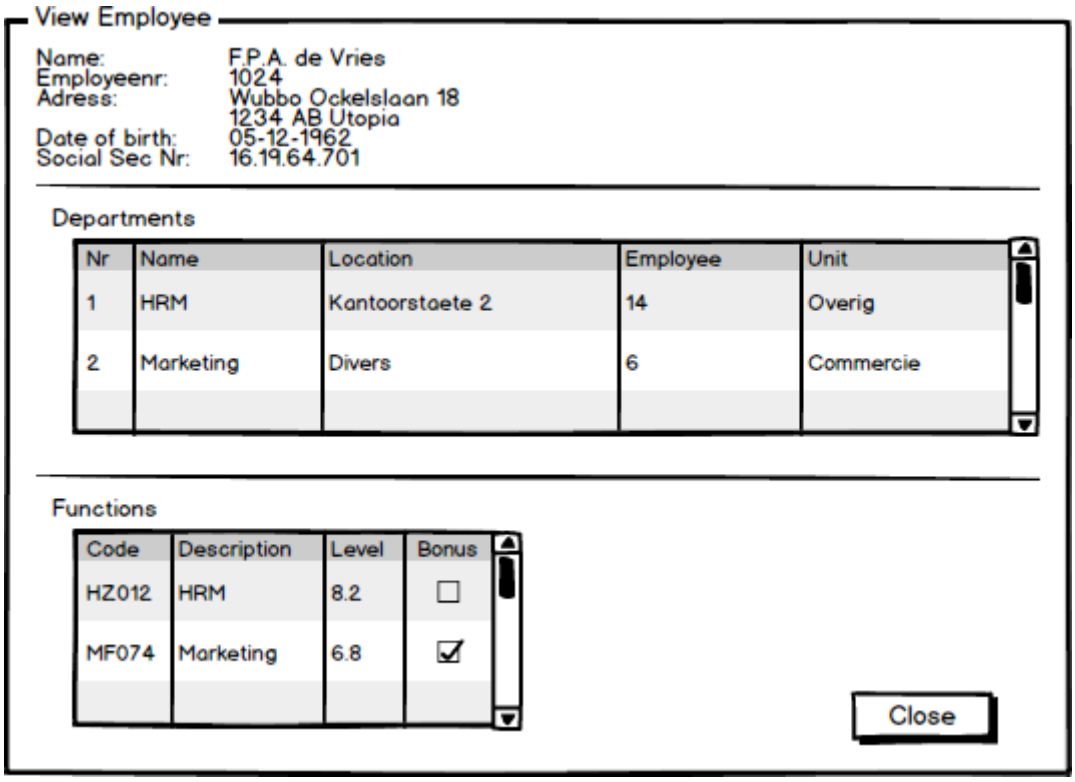
Reference to the standard

4.20

40 COMBINED EXTERNAL INQUIRY

Problem description

Using a viewing function the data of one employee can be viewed. In addition to the data stored in the Employee entity, all data from the related departments and functions of the selected employee are displayed, according to the screenshot.



The question is how this function should be counted.

Discussion

Basically there are three possible solutions to count this feature:

1) as 1 external inquiry (employee data) and 2 external outputs (department and function data) because in this case there are three different logical ways of processing.

2) as 2 external outputs (department and function data) because the choice to display employee data once at the top of the screen and not to repeat lines or data groups is an implementation choice.

3) as 1 external output because the view function can only be shown in its entirety

The first option is not correct, because the difference between the upper block and the lower block is nothing but a difference in frequency/number of occurrences. This is no reason to recognize multiple functions. A report function with a head and repeating details is counted by anyone directly as a single external output normally.

Nor can be chosen on the basis of rule 8.2.g.2 for counting two external outputs. First, the different parts are not separately retrievable and secondly the fact that this report is derived from various logical files is not a sufficient reason to conclude that in this case there are different logical ways of processing within the meaning of Rule 8.2.g.2. The sections shown are indeed related to the employee.

Based on the name of this function (View employee) and its description it has to be concluded that the user has asked "show me the employee data." In other words, the function should be seen as a whole and, therefore as one elementary process.

Solution

Count one external output.

Reference to the standard

2.7, 8.2.g

41 MODIFICATION OF A GENERIC COMPONENT IN ELEMENTARY PROCESSES

Problem description

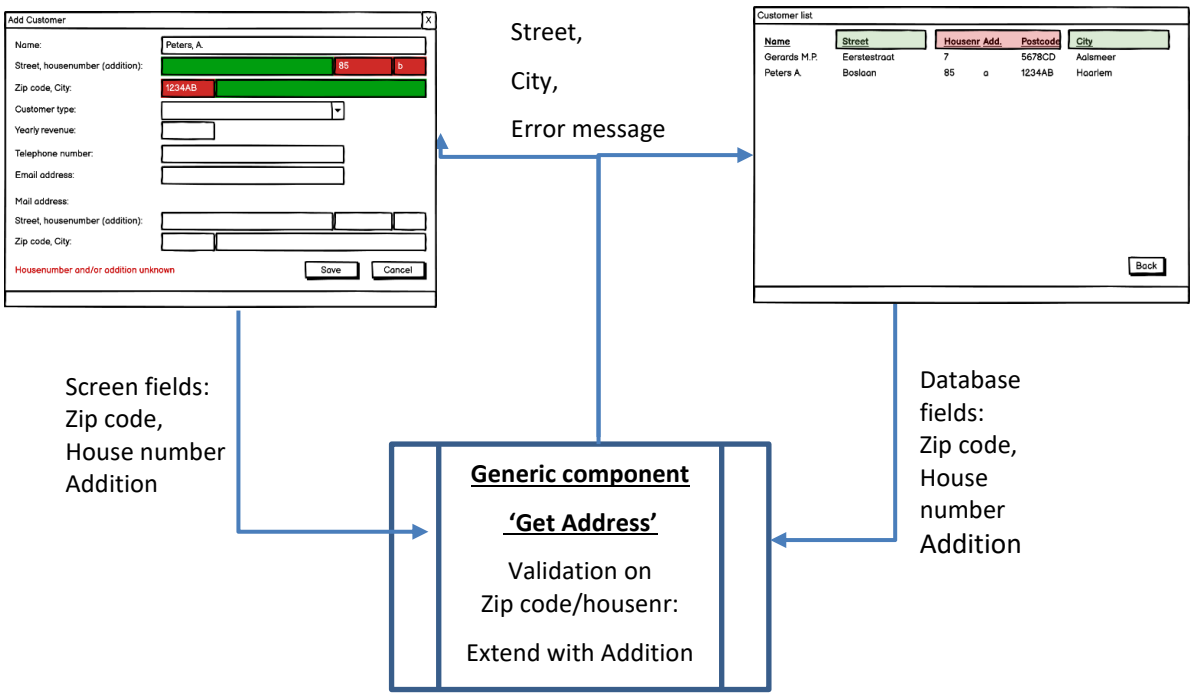
Within a Customer Registration System there is a generic component 'Get address'. This generic component is called in the following transactional functions:

- Add Customer
- Change Customer
- Overview Customers

For the generic component 'Get address' a functional change has been described: *the validation by House number / Zip code is extended with a validation on house number addition*. This means that any error message must also be adjusted.

The external input (Add and Change Customer) calls the generic component with filled in fields Zip code, House number and House number addition and 'Get address' returns Street name, City or an error message that is shown on the screen.

The external output (Customer Overview) calls the generic component with the same fields filled with the contents of the corresponding "Customer" database fields and returns the same three fields. In principle, the error message (validation result) will never be returned, because it concerns already validated data.



How should this functional change be taken into account in the enhancement project?

Discussion

In order to determine the number of function points for the enhancement project, it is necessary to determine which transactional functions functionally change and the number of function points they represent after the change.

The generic component in this example is not an elementary process in itself, but is called by the three transactional functions mentioned. Does this mean that a change in this generic component automatically changes the three transactional function functionally?

One of the criteria for considering a transactional function type as functionally modified is: the logical processing of the transactional function type is changed in the enhancement release (for example, as a result of added, modified and/or deleted validations or calculations). A transactional function type must be functionally modified from the user's point of view after enhancement. The purpose of the transactional function type must not have changed.

For the external inputs, the change in the generic component is noticeable (new error message) and desired.

For the external output, the change is neither noticeable nor desired.

Solution

Count only the external inputs (Add and Change 'Customer') as changed transactional function for the maintenance project.

Reference to the standard

3.6.3, 3.7.2, 7.1, 8.1

COMMENTS AND IMPROVEMENT SUGGESTIONS

We are constantly seeking to improve these examples. When you have comments or suggestions with respect to these examples, send them to:

E-mail: cpc@nesma.org

Website: nesma.org

Product: Examples

Edition: 2022.1

Description:

Enclosed information:

Name: _____

Organization: _____

Address: _____

Phone nr: _____

Date: _____

To establish in an easy manner which rule is elaborated in which example this reference matrix has been compiled.

Counting guideline \ Example	1. Standard authorization functions	2. Specific authorization functions	3. Report generator and Query facility	4. Help functions	5. Error messages	6. Menu structures	7. FPA-tables	8. Denormalization	9. Counting logical files (data functions)	10. Combined External Inputs	11. Analyzing a transaction file	12. Reports on different media	13. Daily and weekly processing	14. Conversion	15. External Outputs with summary information	16. The number of data element types on a report	17. Combined External Outputs	18. Combination effects with functions	19. Querying with different search keys	20. Screens with list functions	21. Browse and scroll functions	22. Selection screens and changing data with a search	23. Direct and delayed processing	24. Case study customer applications	25. Graphs	26. Identifying data element types	27. Generic formatting system	28. Identifying ILF	29. Stubs and drivers	30. Same format, different processing	31. Starting and stopping of batch functions	32. Code and description	33. FPA-table with N:M-relation	34. Saving of selection criteria	35. Master-detail screens	36. Lists within lists	37. Counting transaction files	38. Maintaining an N:M-relation	39. Table with code, name, start- and end-date	40. Combined external inquiry	41. Modification of a generic component
2.1.2 Rationale behind FPA																																									
2.6 Users																														■											
2.7 Functions and function types																																							■	
3.2.2 High level function point analysis																							■																		
3.5.1 Determining the application boundary																										■														
3.6.1 Determining the scope of a project function point analysis																												■												
3.6.2 Functional size of development projects														■														■												
3.6.3 Functional size of enhancement projects																																								■
3.7.2 Modification of a transactional function																																								■
4.1 Analyzing from a logical perspective																										■														
4.7 Screens, windows and reports									■						■		■																							
4.8 Input and output records										■																														
4.9 Security and authorization	■	■																																						
4.11 Report generators and query facilities			■																						■															
4.12 Graphs																									■															
4.13 Help facilities				■																																				
4.14 Messages					■																																			
4.15 Menu structures						■						■																												
4.16 List functions																				■		■																		
4.17 Browse and scroll functions																					■	■																		
4.20 FPA tables	■	■					■		■											■				■								■	■						■		
4.21 Deriving logical files from a normalized data model							■	■	■																								■					■		
4.22 Shared use of data																											■													
4.23 Generic rule for counting data element types					■	■				■		■				■			■							■														
5.1 Definition of an internal logical file																													■											
5.2.a Start from the conceptual data model							■	■	■																															
5.2.b Maintenance of data is of decisive significance									■																															
5.2.f Files introduced for technological reasons													■																											
5.2.i Historical files									■														■																	
5.2.k Entity types with constants, text and decoding	■						■		■											■														■						
6.1 Definition of an external logical file																																	■							
6.2.c Exchange of data via a transaction file														■																										
6.2.g Entity types with constants, text and decoding																								■								■	■							
7.1 Definition of an external input																													■								■			■
7.2.a Counting input of data											■											■																		
7.2.b Maintaining several internal logical files											■																													
7.2.d Activating external inputs											■																													
7.2.g Input screen with different functions										■																														
7.2.m Menu structure						■																		■																

Counting guideline \ Example	1. Standard authorization functions	2. Specific authorization functions	3. Report generator and Query facility	4. Help functions	5. Error messages	6. Menu structures	7. FPA-tables	8. Denormalization	9. Counting logical files (data functions)	10. Combined External Inputs	11. Analyzing a transaction file	12. Reports on different media	13. Daily and weekly processing	14. Conversion	15. External Outputs with summary information	16. The number of data element types on a report	17. Combined External Outputs	18. Combination effects with functions	19. Querying with different search keys	20. Screens with list functions	21. Browse and scroll functions	22. Selection screens and changing data with a search	23. Direct and delayed processing	24. Case study customer applications	25. Graphs	26. Identifying data element types	27. Generic formatting system	28. Identifying ILF	29. Stubs and drivers	30. Same format, different processing	31. Starting and stopping of batch functions	32. Code and description	33. FPA-table with N:M-relation	34. Saving of selection criteria	35. Master-detail screens	36. Lists within lists	37. Counting transaction files	38. Maintaining an N:M-relation	39. Table with code, name, start- and end-date	40. Combined external inquiry	41. Modification of a generic component
7.2.n Presenting data as part of an external input										■												■		■																	
7.2.o Presenting data on a change screen										■																															
7.2.p Separately specified external inquiry										■																															
7.2.r Time-delay													■										■																		
7.2.t Processing of a transaction file											■																														
7.2.x Selection via a non-unique selection criterion																						■																		
7.3.b The way to get to a function						■																																		
7.3.d Several screens																						■																		
7.3.h The number of referenced logical files		■																																						
8.1 Definition of an external output																	■							■					■	■					■						■
8.2.a Counting output of data													■						■	■	■	■																		
8.2.b Counting output products														■																										
8.2.c Distinguish between an external output and external inquiry																			■		■	■																		
8.2.e Entering control data												■																							■						
8.2.g An output product can comprise several external outputs															■		■																			■	■			■	
8.2.i Output to different media												■																												
8.2.j Transaction file as several external outputs														■																							■			
8.2.n Messages about the excution of one function					■																																		
8.2.p Output product as logical result of maintenance of an ILF																							■																	
8.2.q Output on the basis of several selection criteria																			■			■																		
8.2.s Combination of functions																		■																						
8.2.t List from which a user can make a selections																				■															■					
8.2.u Browsing or scrolling																					■	■																		
8.2.w Several output products one external output																								■																
8.3.a Counting all data element types															■				■			■																		
8.3.b Counting control information												■			■				■			■		■																
8.3.c Counting address data												■																												
8.3.d Counting process data															■	■										■														
8.3.e Only data element types that cross the boundary are included																■										■														
8.3.f Counting standard data																■																								
8.3.g Counting fixed data																■			■		■																		
8.3.j Counting check data																																								
9.1 Definition of an external inquiry																													■						■					
9.2.c Distinction between an external inquiry and an external input																					■																		
9.2.d. Query facility and external inquiry			■																															■					
9.2.f Input part of an external inquiry																					■																		
9.2.g Output part of an external inquiry																					■																		
9.2.h Entering data to control data processing																			■		■																		
9.2.j Browsing or scrolling																					■	■																	
9.3 Determining the complexity of external inquiries																			■																				