IBM MQSeries Workflow

ARIS to MQSeries Workflow Bridge

Version 1.0.0

First Edition (March 2001)

This book is edited by International Business Machines Corporation, USA

Published by:

IBM Deutschland Entwicklung GmbH
IBM MQSeries Workflow Development (Department 1892)
Schönaicher Straße 220
71032 Böblingen

IBM MQSeries Workflow related publications

All publications listed in this paragraph are contained in the zu IBM MQSeries Workflow library. For orders please contact your authorized IBM partner or IBM office.

- *IBM MQSeries Workflow: List of Workstation Server Processor Groups,* IBM Form GH12-6357, describes the processor groups for MQ Workflow.
- *IBM MQSeries Workflow: Concepts and Architecture*, IBM Form GH12-6285, describes the fundamental concepts of MQ Workflow. In addition the architecture of MQ Workflow and the collaboration of components is described.
- *IBM MQSeries Workflow: Getting Started with Buildtime,* IBM Form SH12-6286, describes the usage of MQ Workflow Buildtime.
- *IBM MQSeries Workflow: Getting staretd with Runtime*, IBM Form SH12-2962, contains an introduction into the usage of MQ Workflow Runtime Client.
- *IBM MQSeries Workflow: Programming Guide,* IBM Form SH12-6291, describes the application programming interfaces (APIs).
- *IBM MQSeries Workflow: Installation Guide,* IBM Form SH12-2963, contains informations and procedures related to installation and customizing of MQ Workflow.
- *IBM MQSeries Workflow: Administration Guide,* IBM Form SH12-6289, describes how to manage an MQ Workflow System.

ARIS related publications

Please contact IDS Scheer AG, Altenkesseler Straße 17, D-66115 Saarbrücken (Tel. +49(0)681/210-0) or find internet informations at www.ids-scheer.de.

- *ARIS Methods Version 5, manual of the ARIS 5.0 e-Business Suite*, describes the fundamental concepts of „Architecture of Integrated Informationsystems (ARIS)“ and all modelling methods supported by the ARIS Toolset

Chapter 1 Introduction

"Workflow suitability" of business process models

The *ARIS to MQSeries Workflow Bridge*[1] is a software tool that supports the translation of business process models created with "IDS ARIS Toolset" into MQ Workflow models. You can subsequently execute the workflow models with the IBM MQSeries Workflow Runtime component.

The program described in this guideline fulfills this task, provided that the design of the selected ARIS models is "workflow suitable". You can also translate workflow unsuited models, but you will have to rework them using the MQSeries Workflow Buildtime component.

Often, ARIS models are "workflow suitable" to some extent only, because they were designed before their selection for a workflow project with MQ Workflow. Chapter 4 contains related hints how a workflow suited ARIS model should look like.

There are various reasons why you have to conform your ARIS models to the technical requirements of workflow automation. Here is a list of some of them:

Capturing the actual appearance of already existing business processes is a typical intention of ARIS modelling. Usually, you can achieve optimal cost by means of a workflow system, provided that you reorganize the business process according to the science of business administration. This implies the integration of IT applications into the workflow process, a modified staff assignment, and considering reduced activity cycle times.

- Many modelling methods supported by ARIS help you to describe business processes from a "requirements definition" point of view (cp. the related ARIS publications). On the other hand, planning a workflow application requires informations related to the "design specification" level and the "implementation description" level.
- The ARIS Toolset supports almost a hundred modelling methods that serve to represent a variety of subjects related to business processes. The methods offer a great deal of flexibility and let the modeller choose from a rich set of equivalent object and connection types. This makes models possible that conform to the concepts and professional jargon that has been established in a company. The ARIS Bridge was designed independently from a special business nomenclature. That is why sometimes semantic ambiguities must be resolved before you can translate an ARIS model into an MQ Workflow model.
- Some subjects found in ARIS models are not relevant to a workflow model. For instance, the ERM model of a database usually conforms to a database design, but it does not conform to the data view of a workflow design.
- Other informations that are important for a workflow model are often missing completely. Examples:
  - specifications of IT applications that have to be integrated
  - "logical conditions" describing the logic of process control
  - description of the data flows between the workflow activities
  - staff assignment rules
- The ARIS process model violates topological design rules of workflow modelling. Examples:
  - Process chains encompass control flow loops or back branches that must not occur in "cycle-free" workflow graphs (They have to be represented in a different manner, for instance, using "blocks" with a looping logic).
  - The contents of an ARIS diagram (i.e. "eEPC") does not represent a self-contained workflow process:
    - Several processes are comprised in a single eEPC diagram.
    - A single process is shared among several diagrams that are linked by means of "process interface" symbols.

[1] IBM MQSeries Workflow ARIS Bridge will subsequently be abbreviated as "ARIS Bridge".

- Subprocesses are not represented as "assignments" of ARIS functions.
- The ARIS model contains relationships of n:m cardinality, but the corresponding MQ Workflow relationships are restricted to 1:n cardinality. Examples:
  - ARIS models let you assign more than one manager to an organizational unit, but MQ Workflow allows one manager only.
  - You can assign more than one "eEPC" diagrams to an ARIS function, but an MQ Workflow process activity implements a single subprocess only.
- The names used in an ARIS model do not conform to the naming conventions of the workflow model.[2]

"Version control" of business process and workflow models

You should consider the rework of workflow-unsuited ARIS models in every workflow project. This book will not prescribe rigid rules for this. But, in general, the following procedures are feasible:

- The development of ARIS models has been subject to workflow design guidelines that were established before the beginning of the project.
- Already existing ARIS models are redesigned according to workflow design guidelines before translating them into MQ Workflow models..
- The ARIS models are imported to MQ Workflow Buildtime after their translation into a workflow model. MQ Workflow Buildtime is used to rework the model into a "workflow suitable" version.

Obviously, the first option is the perfect case. But in practice it will not always be available, because ARIS models are often selected for workflow projects at a later point of time. The other two options raise the difficulty of maintaining the "versions" of ARIS models on the one hand and MQ Workflow models on the other hand. In particular, the "synchronization" of the model versions may be difficult. The ARIS Bridge does not support an automatic version control or the "reverse translation" of MQ Workflow models into ARIS models. That is why version controlling should also be subject to every workflow project. Table 1 might help you to plan a version control by considering the essential differences of task and character between ARIS models on the one hand and MQ Workflow models on the other hand:

	Task	Character
ARIS	flexible modeling with free choice of the appropriate level of detail	"abstraction" (decision-related subjects are made precise, insignificant things are omitted)
MQSeries Workflow	automatic execution and control of business processes	"template" that serves the control of a real process[3]

Table 1: Differences between ARIS Toolset and MQSeries Workflow requirements

Mapping scope between ARIS models and MQ Workflow models

The ARIS Bridge is a very flexible tool, because it admits of much design freedom to the ARIS modeller. Basically, neither the choice of ARIS model types (eEPC, PCD, Office Process, etc; cp. page 23) nor the choice of ARIS object and connection types is restricted.[4]

---

[2] This ARIS Bridge partially solves this problem by automatically converting ARIS names into MQ Workflow compatible names. Note that very long names that differ only in the last characters (names with equal prefix) may be mapped to identical names after they were shortened to the maximum allowed character string length. As a consequence, the translated workflow model will be wrong.

[3] MQ Workflow models belong to the category of computer programs. That is why they are specified by means of a programming language called „Flow Definition Language" (or briefly „FDL"). In deed, in comparison to ARIS models, the term of an MQ Workflow „model" is misleading, because model abstraction does not belong to the characteristics of a „template". On the contrary, mapping precision directly determines the actual process behavior. „Inaccuracies" due to abstraction must be considered as „process faults".

The model translation is controlled by a set of standard mapping rules which allow for alternate ARIS modelling constructs with equivalent meaning. You may find details thereto in Chapter 5 “Mapping rules” from page 53 on. Example:

> In order to model an IT application, you may select from ARIS object definition types “Application system type”, “IT function type”, or “Module type”.

The design of the mapping rules resulted from a compromise between graphical and semantic mapping accuracy. Here are some examples:

I. Graphical mapping accuracy

A. You find the activity icons in an MQ Workflow process diagram at positions that are similar in position of the corresponding “function” icons of the ARIS model (cp. section „Function view” on page 24).

B. The contents of every ARIS model that represents “process chains” (i.e. an eEPC diagram) is interpreted as a self-contained workflow process.

C. MQ Workflow models translated from ARIS never show “dummy” objects (i.e. inserted activities or connections) that have no corresponding object in the ARIS model.

D. There is no automatic conversion of incorrect control flow loops into an MQ Workflow “block”, because, in general, this may require the insertion of dummy objects (i.e. in the case of crossing back branches) and the restructuring of process chains which not always may have a one-to-one solution.

E. There is no automatic conversion of incorrect control flow loops into MQ Workflow “blocks” or “process activities”, because, in general, this requires a restructuring of the process network, such that the models lose their topographical similarity.

II. Semantic mapping accuracy

A. Every MQ workflow model construct has an unambiguous semantics that is expressed by the process behavior at runtime. That is why only such mappings are subject of the mapping rules where you obviously can expect similar meanings from the given names (i.e. an ARIS “function” is translated into an MQ Workflow “activity”).

B. Mapping ambiguities (i.e. multiple managers assigned to an organizational unit) are solved by translating the first found assignement. The other possible assignments will occur with a warning in the translation protocol (log file). Similar warnings will handle incorrect control flow or data flow loops.

C. Section „Process view“ (from page 39 on) describes the specific mapping rules for the representation of control flow (For instance, the total of all control flow paths between two ARIS functions is mapped onto exactly one control connector of the workflow process). Note that apart from one exception the logical operator symbols of the ARIS “rule”-objects are not considered, because, in general, their is no simple method how to assign them to the tree types of control flow conditions that you may find in a workflow process (cp. section „Control flow conditions“ from page 41 on).

D. The mapping rules always refer to the “definitions” of ARIS object or relationships. They do not refer to the respective symbol “occurences” in an ARIS diagram. That is why the copying of ARIS objects may lead to unexpected mapping results if you do not carefully observe the difference between a “definition copy” and an “occurence copy”.

E. ARIS functions that carry another model containing process chains (i.e. another eEPC model) as an “assignment” will be mapped to workflow “process activities”. But, note that no ARIS functions are mapped to workflow “block activities”, because you cannot find clear discriminating characteristics in any ARIS model (cp. section „Subprocesses“ from page 51 on).

---

[4] Improved clearness may require to model organizations in an organigram, data structures in an ERM diagram, or process chains in an eEPC diagram. References to organizational units or data within an eEPC diagram will be assigned appropriately, even when the details are found in a separate diagram type.

Chapter 2 Installation

Prerequisites

In order to install the ARIS Bridge you need as an operating system Microsoft Windows NT 4.0 or Microsoft Windows 2000. You also need ARIS e-Business Suite Version 5.0[5] with the ARIS „Tool Integration" component[6].

It is not necessary to install IBM MQSeries Workflow[7]. But it is recommended to have the MQSeries Workflow Buildtime component installed on the same system, such that you can import and review the workflow models that you created in the FDL format with the ARIS Bridge.

Procedure

In order to install the ARIS Bridge, insert the ARIS Bridge CD-ROM disc into your CD-ROM drive. The installation program will come up automatically and guide you step by step through the installation procedure. With a mouse click on button „Next" you get to the dialog window of the respective next installation step. Clicking on button „Back brings you back to the previous installation step where you may change your data input if needed. With button „Cancel" you may leave the installation program at any time.

Picture 1: Installation program („Welcome")

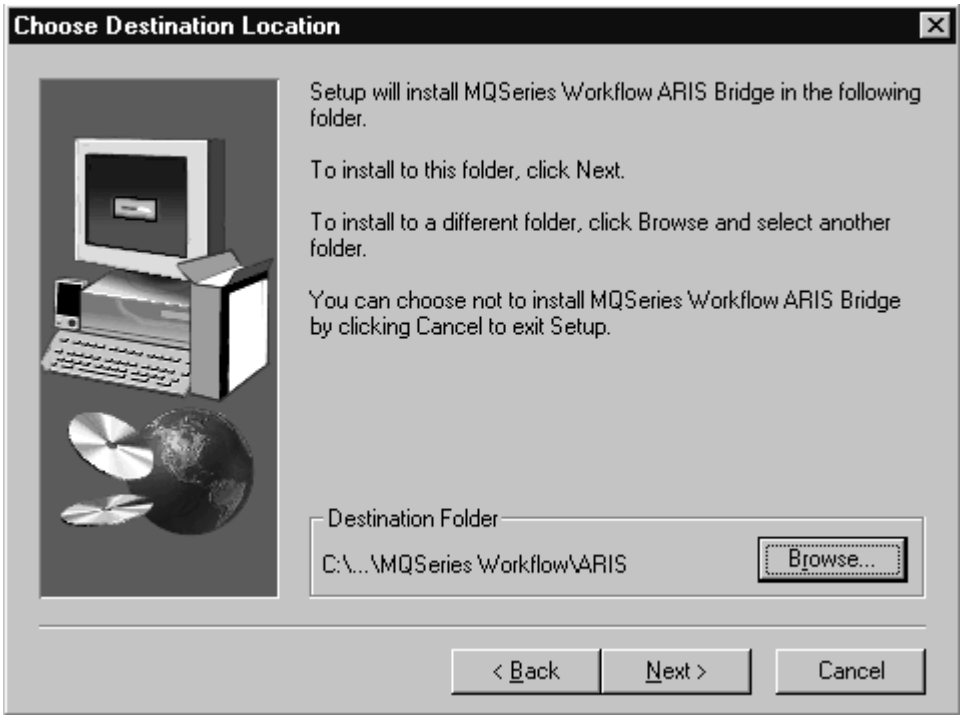
Clicking on button „Next" of the „Welcome" window (Picture 1) brings you to the dialog window „Choose Destination Location" (Picture 2). There you will be prompted for the folder where you desire to install the ARIS Bridge. Click on button „Browse", if you want to replace the proposed folder by another one.

---

[5] In the following the ARIS e-Business Suite will be abbreviated as „ARIS Toolset".

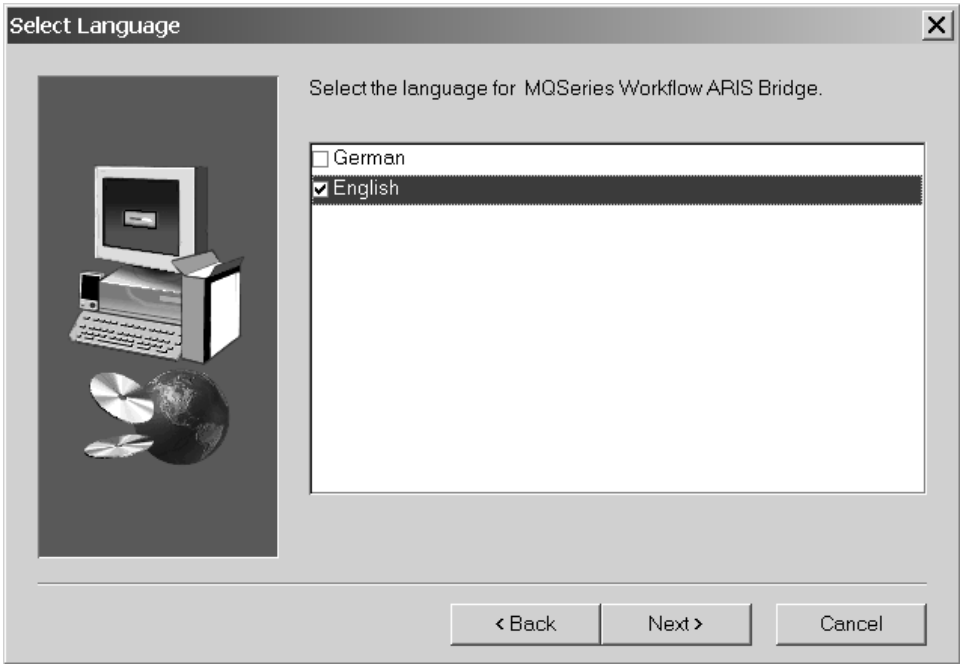
[6] You have to install ARIS Tool Integration in a separate step. Please, lookup the ARIS installation documentation for further details.

[7] MQSeries Workflow Version 3.2.2 or later.



Picture 2: Installation program (selecting the installation folder)

The next dialog lets you select from the languages English and German (Picture 3).



Picture 3: Installation program (language selection)

Another click on button „Next“ extracts the compressed installation files into the target folder that you entered before. A window titled „Dialog Complete“ reports the completion of this step (Picture 4). Clicking on button „Finish“ will terminate the installation of the ARIS Bridge.

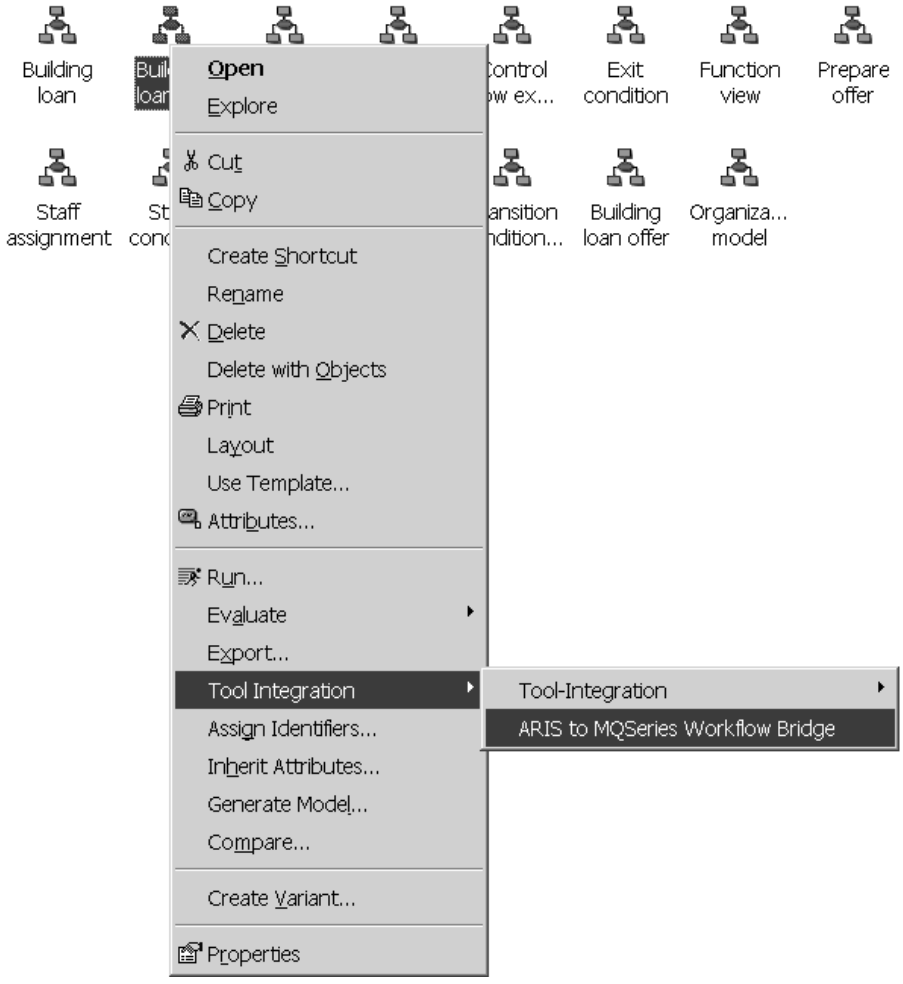
Picture 4: Installation program (completion message)

Chapter 3 How to translate ARIS models into “MQSeries Workflow” models

Selecting ARIS models

The IBM MQSeries Workflow ARIS Bridge translates ARIS models into workflow models using the FDL format[8]. You may use the IBM MQSeries Workflow Buildtime component to import the FDL file and view it with a graphical editor. There you may also change your model if needed. You can select more than one model for translation. Note that this might result in longer response times with large models. Independently from the number of selected models, each translation creates exactly one FDL file. There is no restriction on the model types that you can select from. But note that only certain ARIS model types are reserved for the generation of workflow processes. The processes will inherit the names of the respective models (cp. section „Which types of ARIS models can you translate into MQSeries Workflow models?“ on page 23). The translation will automatically consider any relationships of the objects contained in the model. For instance, the ARIS Bridge will also translate models that were assigned as “subprocesses”, even if you did not select them (cp. section „Subprocesses” on page 51.

Starting the ARIS Bridge



Picture 5: Starting the ARIS Bridge

[8] FDL = Flow Definition Language

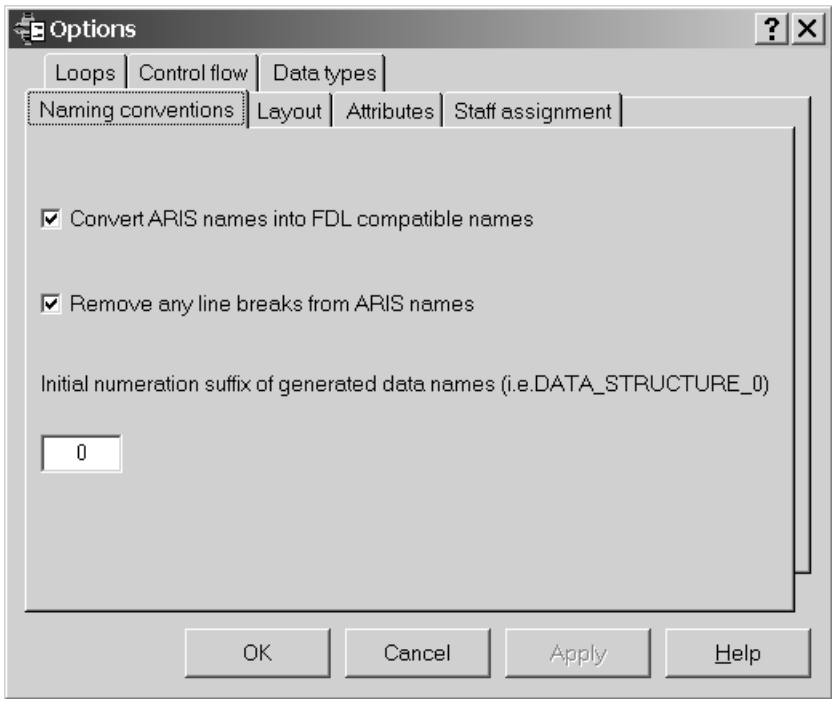
After your selection of one or more ARIS models activate the context-menu with the right mouse-button and start the ARIS Bridge with a left mouse-button-click on menu-item „Tool-Integration“ → „ARIS to MQSeries Workflow Bridge“ (Picture 5).

The ARIS Bridge begins with offering you an options dialog. The following sections explain the settings you can specify there.

Options dialog

If you change any of the below described translating options, your settings will be stored as default values for the next start of the ARIS Bridge. After finishing with the options you get with a click on button „OK” to the FDL file selection window (page 19) or you decide to quit the ARIS Bridge with „Cancel“.

Naming conventions



Picture 6: options dialog panel „Naming conventions”

Convert ARIS names into FDL compatible names

With this option you may decide that the names used in ARIS models will be automatically converted into FDL compatible names, if they violate the naming conventions of MQSeries Workflow. Each name conversion will be reported with a respective message in the log file (cp. section „The log file” on page 21).

If you activate this option, it may happen that similar names that differ only in the last characters will be short-cut to the allowed maximum character length, such that they turn out to be identical names after this operation. The workflow model will be incorrect in this case. Therefore, be careful with long names and first try to import to import the FDL file without name conversion into MQSeries Buildtime. You also will find respective warnings there.

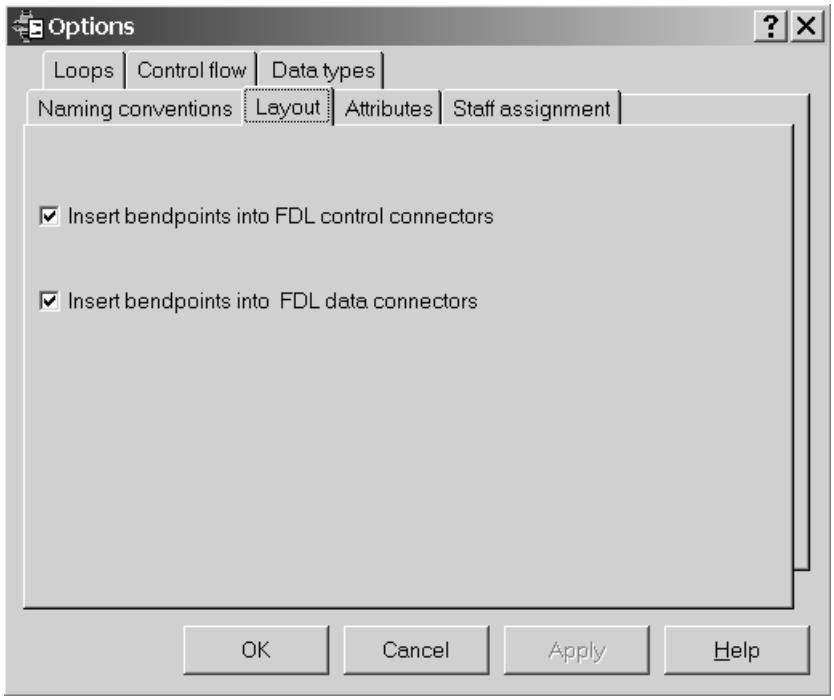
Remove any line breaks from ARIS names

While the ARIS Bridge accesses ARIS text attributes, this option effects that any existing line breaks are removed automatically.

Initial numeration suffix of generated data names

The ARIS bridge automatically creates names for such data structures that it aggregates from ARIS data objects (cp. section „Data containers” from page 33 on). The name is „DATA_STRUCTURE_nn“ with a count index „nn“ that starts from zero by default. It is recommended to have “numbering ranges” for the data structures that you organize with the count index. Numbering ranges help you to avoid overwriting already existing data structures with the same index during FDL import into MQSeries Workflow-Buildtime. Overwriting existing data structures might happen if you start the ARIS Bridge repeatedly while you are translating several models sequentially.

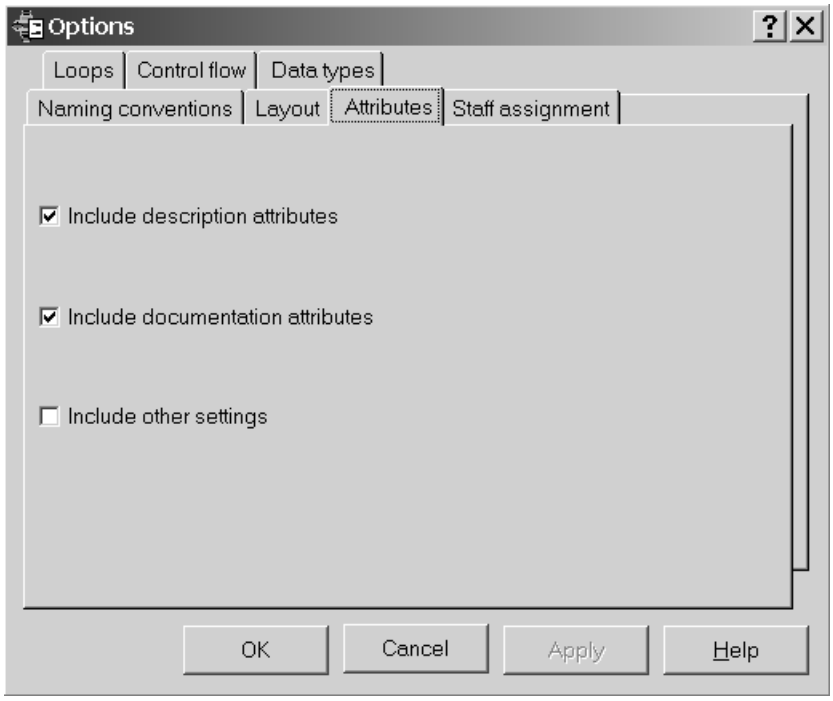
Layout



Picture 7: options dialog panel „Layout”

By default, activities in an MQSeries Workflow process diagram are linked by straight-line data or control connectors. Therefore, it may happen that data and control connectors belonging to identical activity pairs get overlaid. You can avoid this effect with inserting bend points. The options of settings register „Layout“ lets you decide for both connector type the inserting of exactly one bend point for each connector instance. The ARIS Bridge derives the position of the bend points from the geometrical center of the ARIS objects that contribute to data flow or control flow, respectively.

Attributes



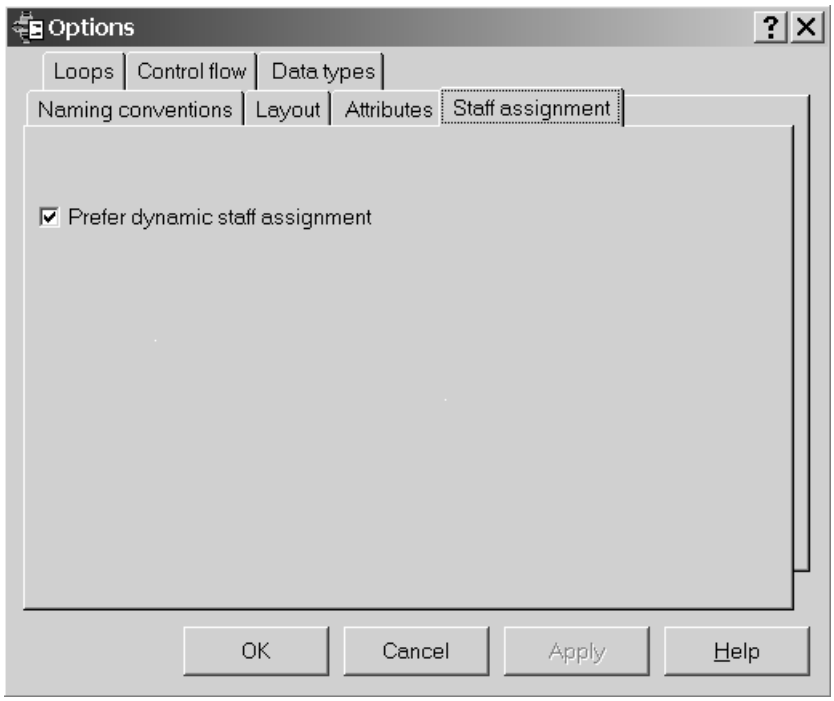
Picture 8: options dialog panel "Attributes"

On options dialog panel „Attributes" you may decide whether you want to have ARIS text attributes with descriptions, documentations, or "other settings" copied from your ARIS models to the workflow model. "Other settings" are FDL expressions that you may enter in order to specify settings of MQ Workflow constructs without an equivalent inside the ARIS model. You may lookup the following mapping rules for the ARIS text attributes that are considered during translation:

- ACTIVITY_DESCRIPTION (page 55)
- ACTIVITY_DOCUMENTATION (page 55)
- ACTIVITY_START_MODE (page 56)
- ACTIVITY_EXIT_MODE (page 56)
- ACTIVITY_OTHER_SETTINGS (page 57)
- CONTROL_CONNECTOR_DESCRIPTION (page 57)
- DATA_CONNECTOR_DESCRIPTION (page 57)
- DATA_MEMBER_DESCRIPTION (page 57)
- DATA_MEMBER_DOCUMENTATION (page 57)
- DATA_MEMBER_SIZE_OF_ARRAY (page 57)
- DATA_STRUCTURE_DESCRIPTION (page 58)
- DATA_STRUCTURE_DOCUMENTATION (page 58)
- ORGANIZATION_DESCRIPTION (page 59)
- PERSON_DESCRIPTION (page 59)
- PERSON_OTHER_SETTINGS (page 60)
- PROCESS_DESCRIPTION (page 60)
- PROCESS_DOCUMENTATION (page 61)
- PROCESS_OTHER_SETTINGS (page 61)
- PROCESS_CATEGORY_DESCRIPTION (page 61)
- PROGRAM_DESCRIPTION (page 61)

- PROGRAM_DOCUMENTATION (page 61)
- PROGRAM_OTHER_SETTINGS (page 61)
- ROLE_DESCRIPTION (page 61)

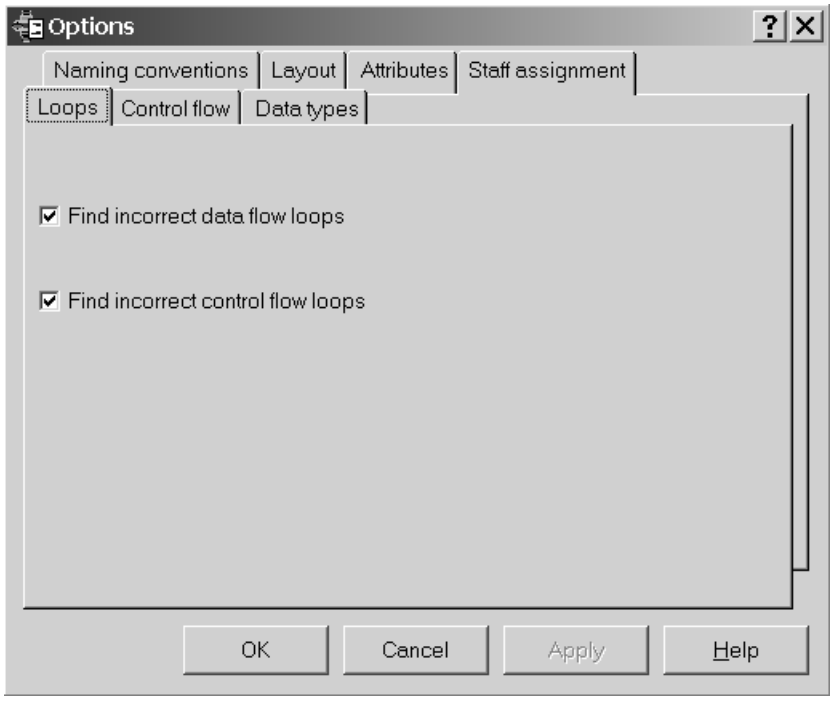
Staff assignment



Picture 9: options dialog panel „Staff assignment”

MQSeries Workflow models let you select from two staff assignment policies. One of them lets you assign individuals to workflow activities, the other one is called „dynamic staff assignment” and lets you specify the responsibility of organizations or roles. If the staff assignment policies occurring in your ARIS model are not specified without ambiguity, you may decide with this option which policy your prefer to be considered during translation for conflict cases (cp. section „Staff assigment strategies” from page 27).

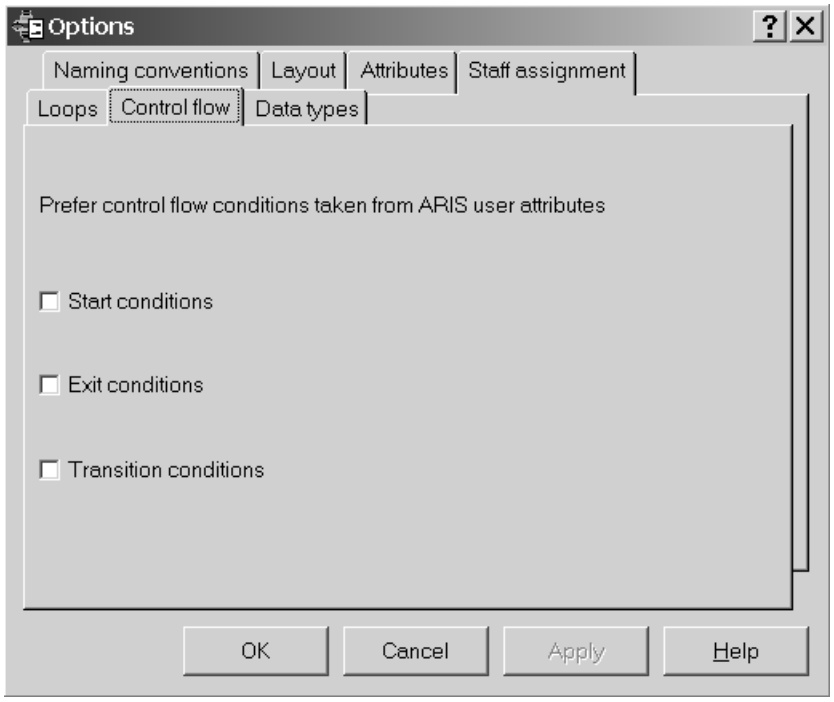
Loops



Picture 10: options dialog panel „Loops"

In general, MQ Workflow models must not contain loops that are setup from data or control connectors. With this option you may let the ARIS Bridge search for incorrect loops and write respective warnings into the log file (cp. section „Data connectors" on page 35 and section „Control connectors" on page 39).

Control flow



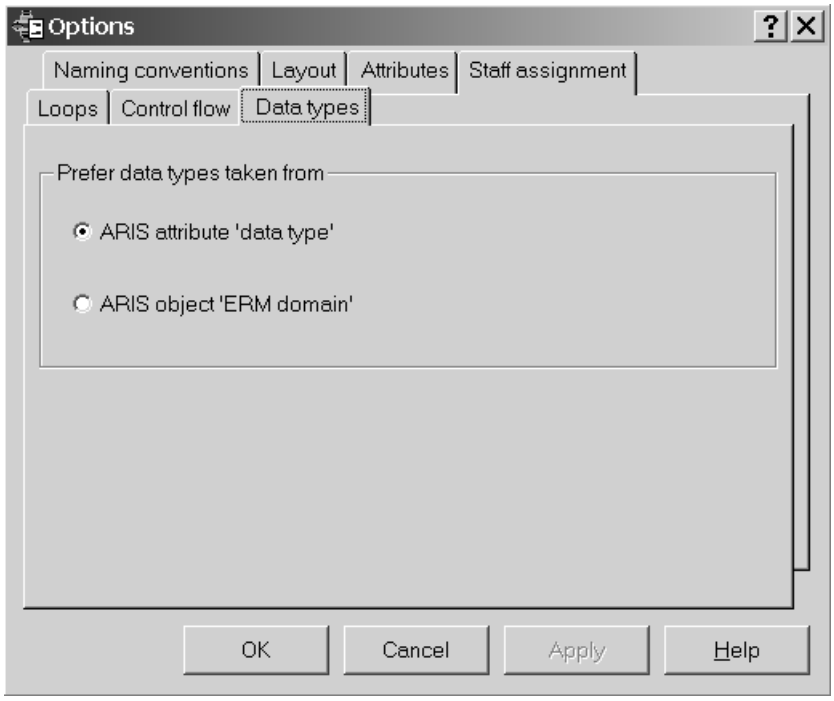
Picture 11: options dialog panel „Control flow"

The ARIS Bridge supports two alternate modes how you can assign logical conditions to an ARIS control flow such that they can be translated to an MQSeries Workflow model:

1. graphical modelling with automatic generation of the logical expressions
2. explicit specification of logical expressions by means of ARIS user attributes

This option lets you specify which description mode you prefer in conflict cases (cp. section section „Control flow conditions” on page 41).

Data types

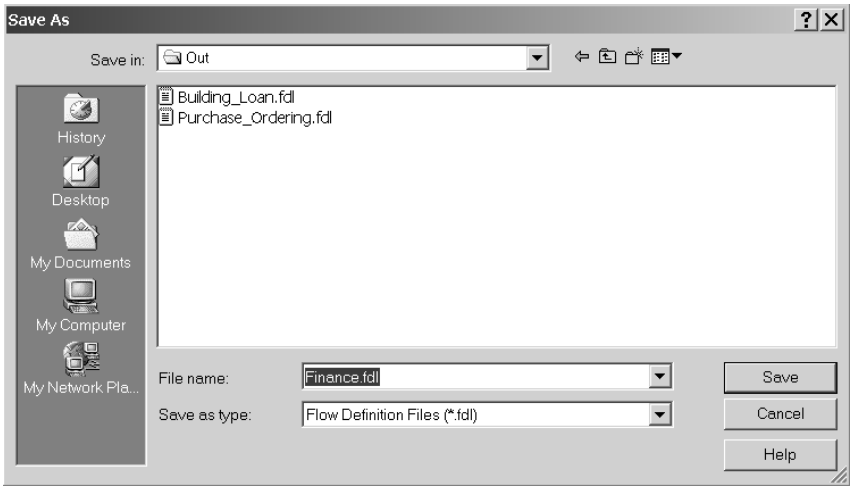


Picture 12: options dialog panel „Data types”

The ARIS Bridge supports two alternate modes how you can specify a data type in your ARIS model. This option lets you choose which description mode you prefer in conflict cases (cp. section section „Data type“ on page 31).

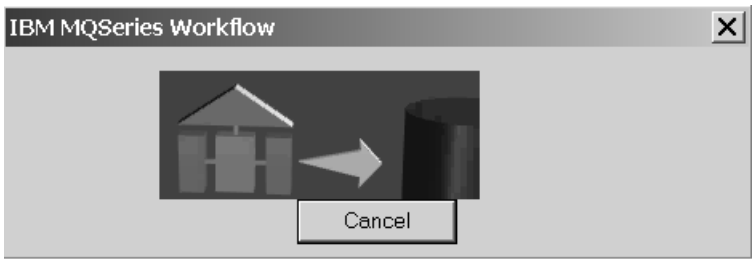
The FDL file and starting the translation

After leaving the options dialog with a click on the „OK“-button a file selection window is displayed. It lets you specify the path and name of the FDL file that you want the ARIS Bridge to write the translated ARIS model into, using the FDL format (Picture 13).



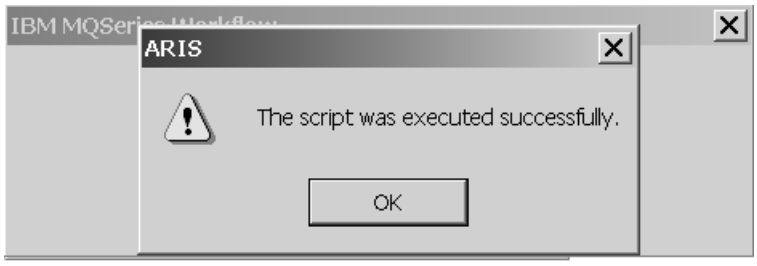
Picture 13: FDL file selection dialog

After that, you will import the FDL file into the database of either IBM MQSeries Workflow Runtime or MQSeries Workflow Buildtime. The latter one is recommended for the review of the generated models. You may quit the ARIS Bridge with a mouse-click on button „Cancel“. You start the translation with a click on button „Save”. The translation process will run as an background operation, while the window of Picture 14 is displayed.[9]



Picture 14: progress indicator of the ARIS bridge

The message shown in Picture 15 indicates the termination of the translation process.



Picture 15: termination message of the ARIS Bridge

[9] Attention: Note that quitting the translation process with button „Cancel“ is no longer possible at this point of time.

The log file

With the file selection dialog for the FDL file (cp. page 19) you also specify the folder which the ARIS Bridge stores the translation protocol (log file) into. The log file gets the same name as the FDL file, except for the file extension, which is called „log“.[10]

The log file provides you with the following informations:

mapping messages	Information about which data from the ARIS model database were mapped to which FDL constructs.
progress messages	For example, the message that the properties of a workflow process are setup.
conversion messages	Information about the conversion of ARIS names into MQ Workflow compatible names.
warnings	For example, the warning that an incorrect loop was detected.
termination message	Reports the total number of FDL objects created.

[10] By the way, you can observe the translation progress, if you open the log file with a text editor and refresh the displayed contents from time to time. But, note that the FDL file itself will not be written before the translation process is finished.

Chapter 4 How can you make workflow-suitable ARIS models?

MQSeries Workflow considered from an ARIS point of view

You can easily understand the modelling samples contained in this chapter, if you already have basic knowledge about the following subjects:

1. „Architektur integrierter Informationssysteme (ARIS)“
2. Concepts and structure of MQSeries Workflow models

The modelling samples are ordered according to the ARIS descriptive views:

Data view	Data model of process related conditions and events.
Organization view	Organizational units and performers of functions, as well as their relationships and structures.
Function view	Functions to be performed (operations) within a business process, as well as their relationships.
Resources view[11]	IT resources and their relationships.
Process view[12]	Relationships between the objects of data view, organization view, function view, and resources view.

Which types of ARIS models can you translate into MQSeries Workflow models?

In principle, you can translate any ARIS model with the ARIS Bridge. The program automatically searches for all ARIS model components that can be converted with the mapping rules into corresponding FDL constructs (cp. „Mapping rules“ in Chapter 5 from page 53 on). But, you should note that only certain model types are suitable for workflow processes (cp. Table 2 and mapping rule „PROCESS“ on page 55).

[11] The ARIS method considers the IT resources as the conditional framework that constrains the remaining views (data, functions, organization). It assigns the resources view to the descriptive levels „design specification“ and „implementation description“.

[12] Also called „control view“ in the ARIS terminology.

ARIS-Modelltyp	Typnummer
eEPC	13
eEPC (material flow)	50
eEPC (column display)	134
eEPC (table display)	154
eEPC (row display)	140
Functions allocation diagram	14
Industrial process	103
Office process	100
PLOVC	138
Process chain diagram (PCD)	18
PCD (material flow)	51
UML Activity Diagram	124
Value added chain diagram	12

Table 2: ARIS model types suitable for workflow processes

In practice, you will use several ARIS modelling methods simultaneously, such that you apply the most suitable methods for each descriptive view. Example:

Data view	eERM
Organization view	Organizational chart
Function view	Function tree
Resource view	Application system type diagram
Process view[13]	eEPC

In this example, it is sufficient to select the eEPC model in the ARIS Toolset only, in order to let the ARIS Bridge export the respective business process with all its views. The ARIS Bridge will automatically also translate related informations in the other diagrams, provided that the eEPC model refers to them by respective object or relationship occurences.[14]

Example: Imagine that the eEPC contains a function chain and some occurrences of „organization units". Then the ARIS Bridge will also find and translate the corresponding details in the organizational chart (i.e. the „persons" that belong to the „organization unit").

How are the ARIS views translated into MQSeries Workflow models?

Function view

The ARIS Bridge translates ARIS objects of type „Function (22)"[15] into corresponding objects of type "activity" of the MQ Workflow model. After a transformation of coordinates, the activity positions

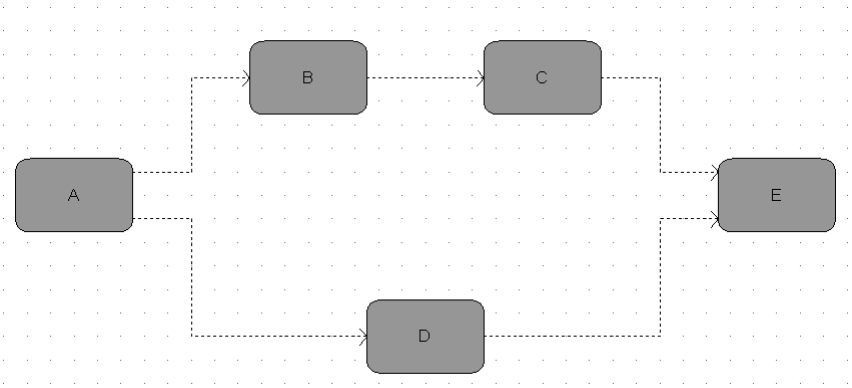
[13] Also called „control view" in the ARIS terminology

[14] Please, note that, during model translation, the ARIS Bridge always considers any relationships to objects with occurrences in other models, even when the relationships are not visible in the current diagram.

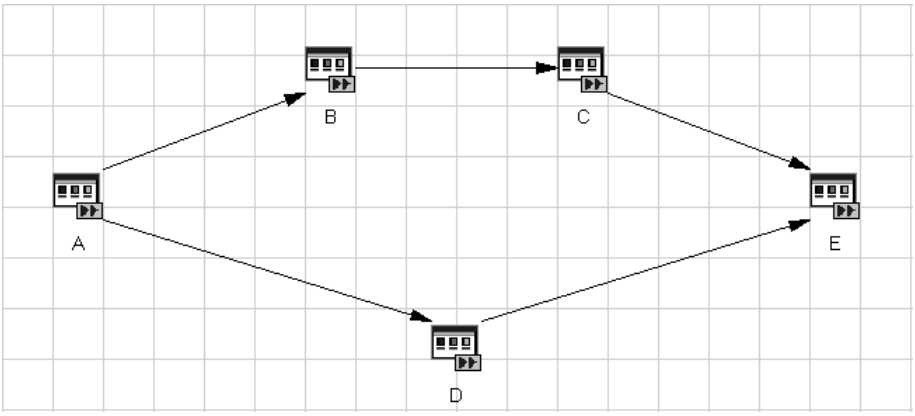
[15] Subsequently, ARIS model elements are always marked with respective ARIS type numbers in parentheses.

inside the workflow process diagram correspond to the positions of the function occurrences inside the ARIS diagram.

The process diagram of the workflow model will show one activity symbol only for each function definition with multiple occurrences in the ARIS model.[16] In this case, the position of the function occurrence first found in the ARIS diagram will be used to calculate the position of the activity.



Picture 16: Functions inside an ARIS model



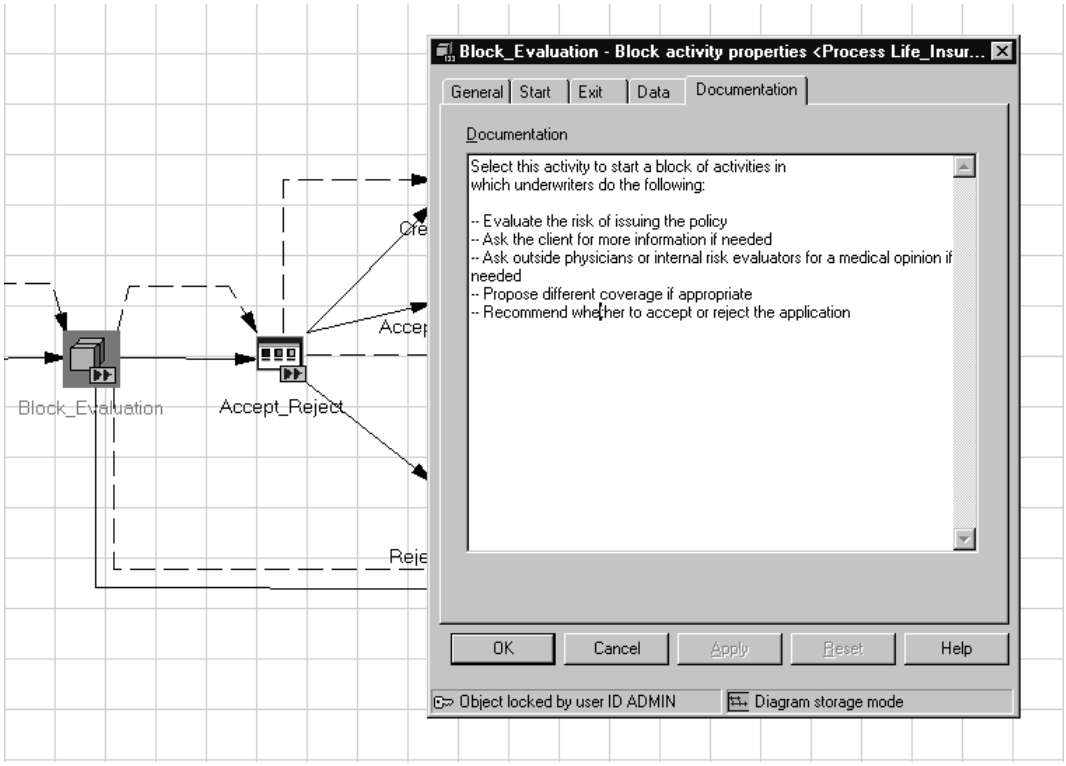
Picture 17: Activities inside an MQ Workflow model (after translation)

Picture 16 shows a simple ARIS model with functions that are chained to each other with connectors of type „is predecessor of (118)“. Picture 17 shows the corresponding MQ Workflow model after the translation.

The arrows between the workflow activities are so-called „control connectors“, which describe the logical order of sequence. You find the details there of in section „Control connectors“ from page 39 on.

Any textual entries into function attribute „Description/definition (9)“ are automatically copied into the „Description“-field of the workflow activity. Accordingly, attribute „Remark/Example (8)“ is copied into the „Documentation“-field. Picture 18 shows the example of a documentation.

[16] MQ Workflow models contain at most one occurrence of each activity, only.



Picture 18: Workflow activity with text entered into the documentation field

Organization view

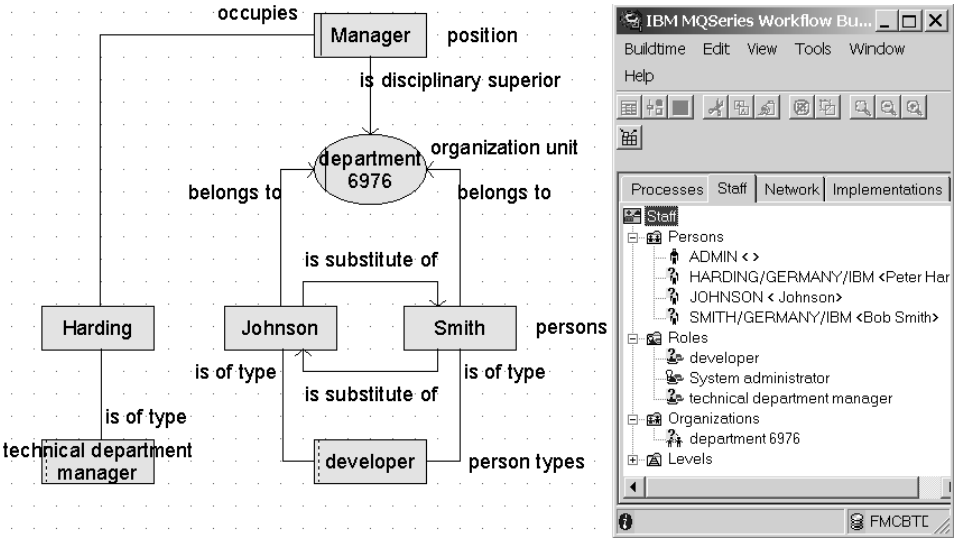
Objects and relationships

ARIS	MQ Workflow
Organizational unit (37)	ORGANIZATION
Person (46)	PERSON
Person type (78)	ROLE

Table 3: Objects of the organization view

Table 3 summarizes the transformation rules for the organization view.

Picture 19 presents to you the example of a simple ARIS organizational chart with an organization unit, the persons and tasks. Immediately to the right, you see a tree representation of the translation into the workflow model.



Picture 19: ARIS organizational chart (translation into the workflow model to the right)

The diagram illustrates some important relationships that can occur in an organizational chart.

Relationship	Mapping Rule
Person belongs to an organization	PERSON_RELATED_ORGANIZATION (p.60) ORGANIZATION_RELATED_PERSON (p.59)
Person performs a role	PERSON_RELATED_ROLE (p.60) ROLE_RELATED_PERSON (p.62)
substitute relationship	PERSON_SUBSTITUTE (p.60)
organization hierarchy	ORGANIZATION_PARENT_ORGANIZATION (p.59)
manager of an organization	ORGANIZATION_MANAGER (p.59)
coordinator of a role	ROLE_COORDINATOR (p.62)

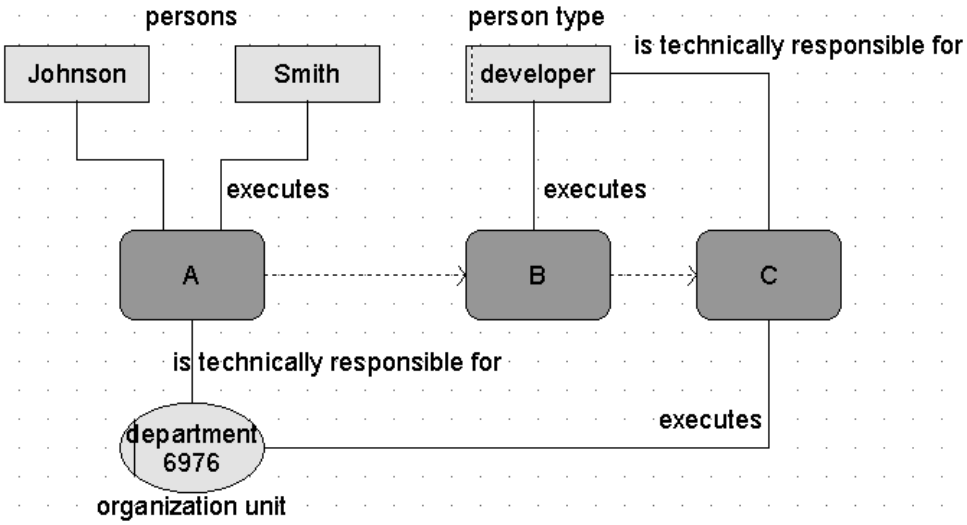
Table 4: Relationships inside an organization and related mapping rules

You may lookup the mapping rules that determine the translation in Table 4 and in Chapter 5 „Mapping rules“.

Please, note that, other than in ARIS models, relationship cardinalities in an MQ Workflow model are confined, as follows:

- Every person belongs to one organization, only.
- Every person can have one other person as a substitute, only.
- Every organization can have one superior organization, only.
- Every organization can have one manager, only.
- Every role can have one coordinator, only.

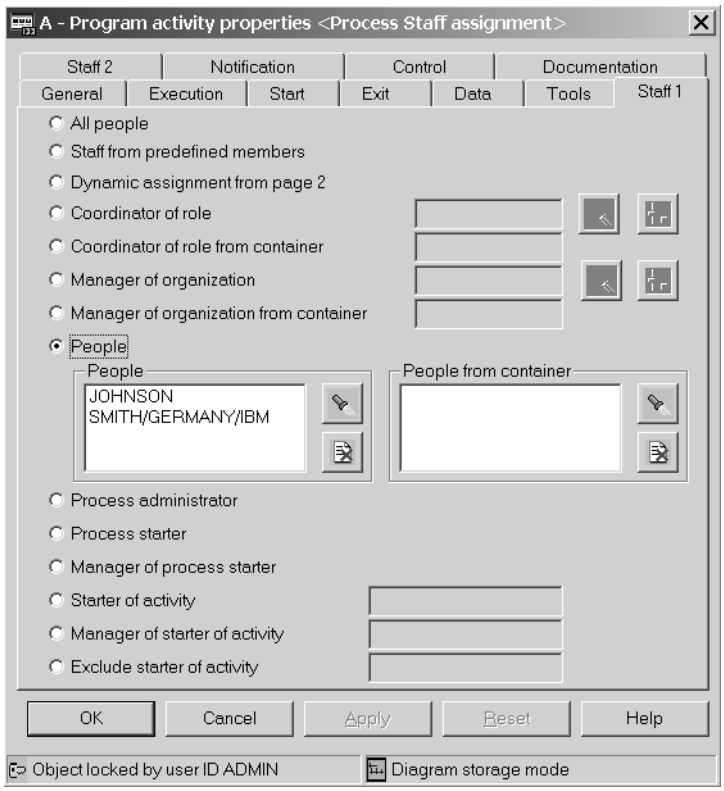
Staff assigment strategies



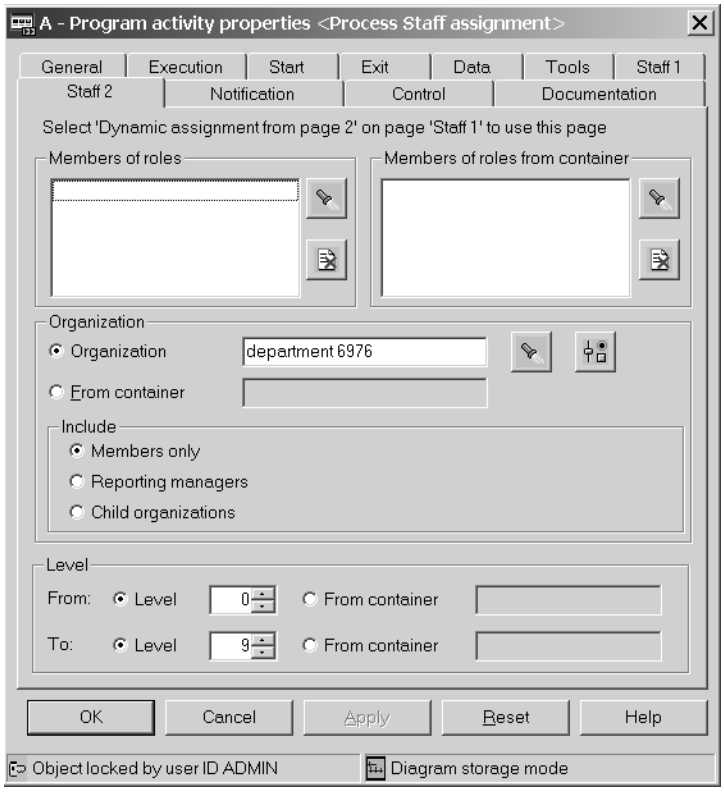
Picture 20: Alternate possibilities of staff assignment

Picture 20 demonstrates the possibilities how you can represent in a ARIS model the assignment of staff to the functions of a process. Note that the activities of an MQ Workflow model can be assigned either individual persons or organizations and roles. The latter is called “dynamic staff assignment”. Dynamic staff assignment and the assigning of individual persons exclude each other. That is why there is a panel in the options dialog that lets you specify which staff assignment policy you prefer for the translation.

For example, assume that you enabled option „Prefer dynamic staff assignment”. The properties notebook of activity „A“ in the translated workflow model will show persons Johnson and Smith as members of the “People”-list on panel „Staff 1“ (Picture 21). The translation ignores in this case that you also assigned „department 6976” to function „A“ in the ARIS model: that is to say, there are no entries on panel „Staff 2“. On the other hand, if you select the dynamic staff assignment, the organizational unit „department 6976” will occur on panel „Staff 2“ after translation, while the “People”-list on panel „Staff 1“ remains empty (Picture 22).



Picture 21: Individual persons assigned to activity „A“ (Option „dynamic staff assignment“ disabled)



Picture 22: Organization assigned to activity „A“ (Option „dynamic staff assignment“ enabled)

Data view

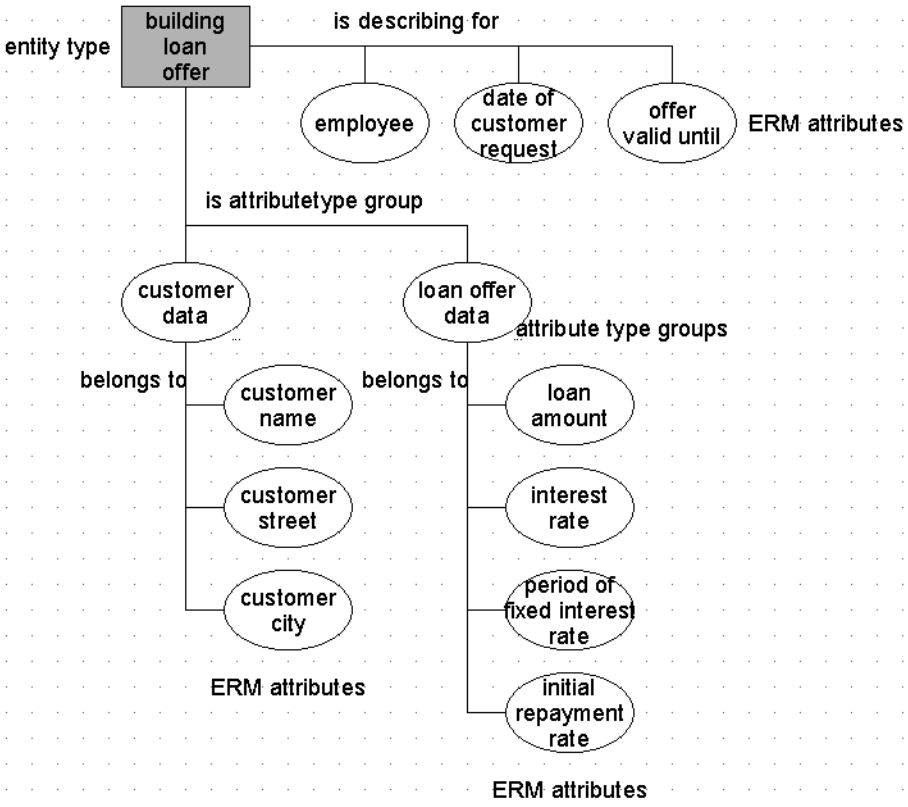
Data structures

In MQSeries Workflow models, you describe the communication of data between the activities of a workflow process by means of data structures. Every FDL data structure consists of data elements. A data structure element itself can be another data structure again. With the ARIS Bridge you can translate the following ARIS object types into data structures and their elements (Table 5 and mapping rule „DATA_OBJECT“ on page 54):

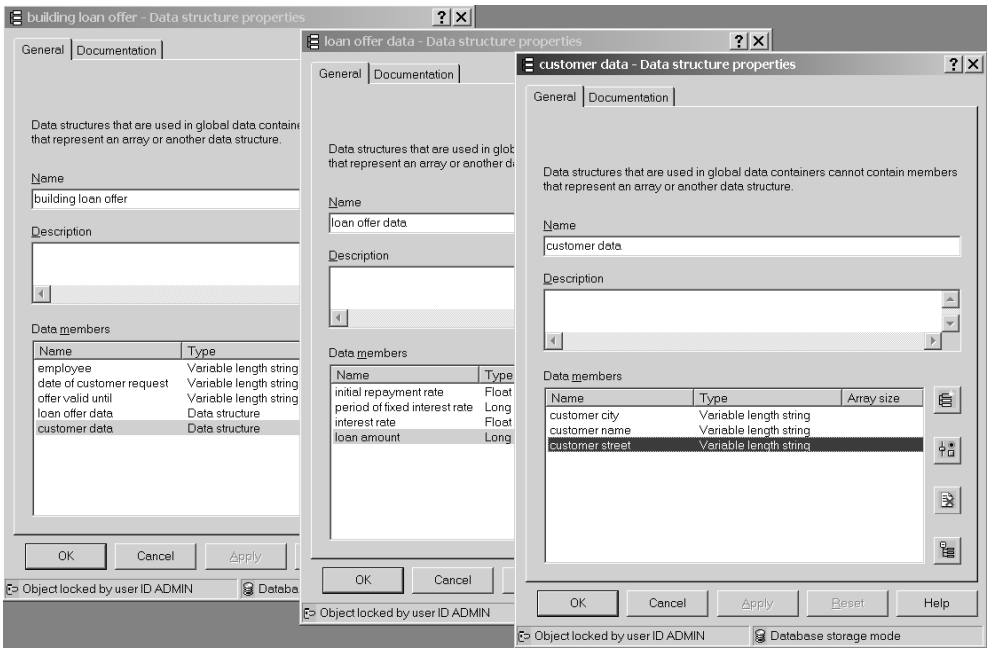
ARIS object type
Entity type (17)
Relationship type (11)
ERM attribute (19)
Cluster/Data model (14)
Attribute type group (111)
Information flow (26)

Table 5: ARIS object types that can be translated into FDL data structures

Picture 23 illustrates the example of a simple data structure that is represented in an eERM diagram. Picture 24 shows you the view of the same data structure in MQ Workflow Buildtime after its translation with the ARIS Bridge.



Picture 23: Example of a data structure

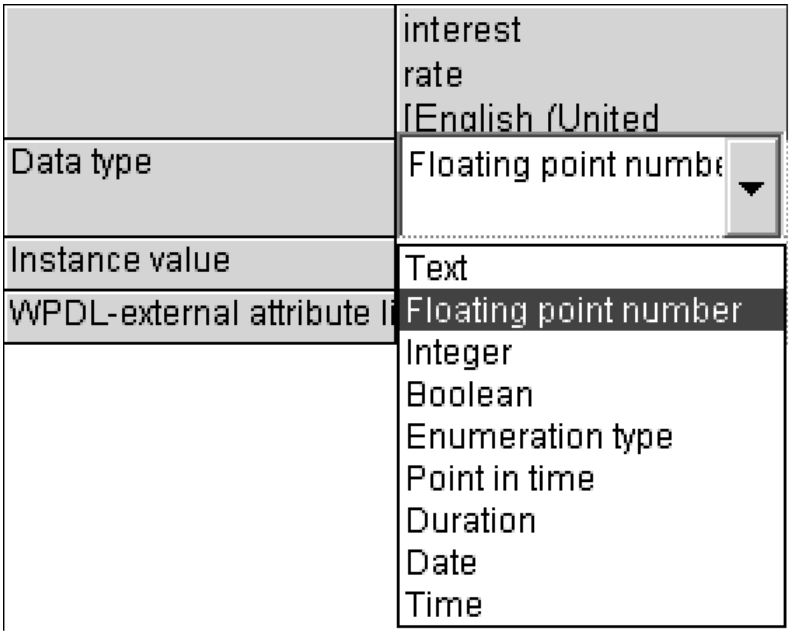


Picture 24: View of the same data structure in MQ Workflow-Buildtime

In the table of mapping rule „DATA_STRUCTURE_MEMBER“ on page 58 you can lookup the connection types, that you can use in an ARIS model, such that the data structure you designed will be converted with all its elements into the MQ Workflow model. Please, note that MQ Workflow does not allow recursive data structures.

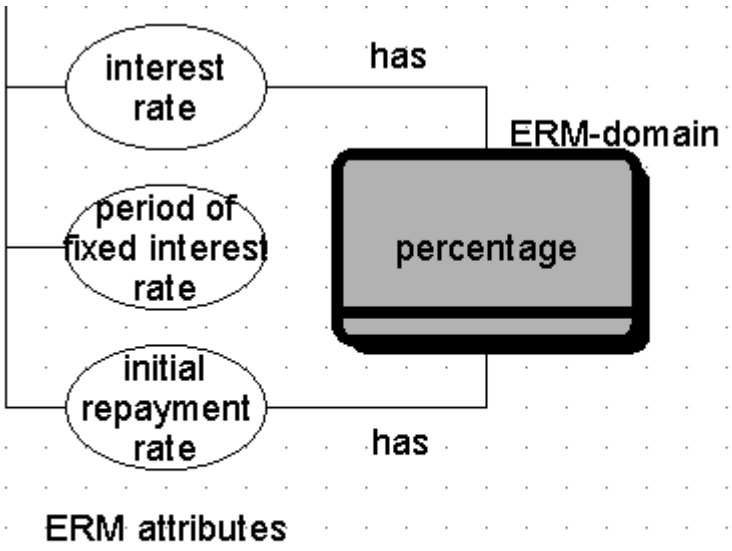
Data types

Perhaps, you noticed that the MQ Workflow representation of data structure „building loan offer” in Picture 24 obviously also derived data types from the ARIS model in Picture 23. You have several choice how to define a data type in an ARIS model. The simplest one is to select a data type for the corresponding ARIS attribute of the same name (Picture 25). You find the respective mapping rule „DATA_MEMBER_TYPE“ on page 54.

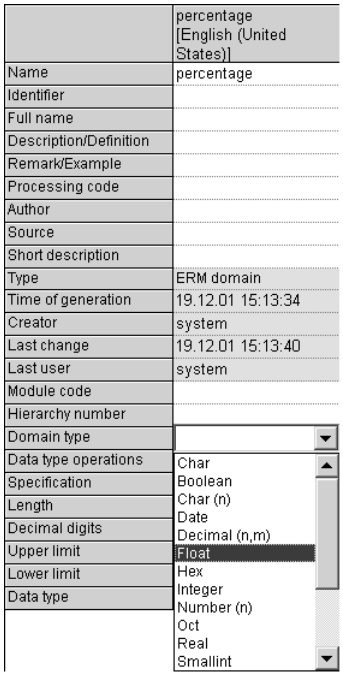


Picture 25: Defining the data type by use of ARIS attribute "Data type"

As another possibility, you may explicitly allocate an ARIS object of type „ERM-domain (20)" and specify a value for the object definition attribute „Domain type" (Picture 26 and Picture 27). You find the corresponding mapping rules „DATA_MEMBER_HAS_DOMAINTYPE" and „DOMAINTYPE_DOM_TYPE" from page 57 on.

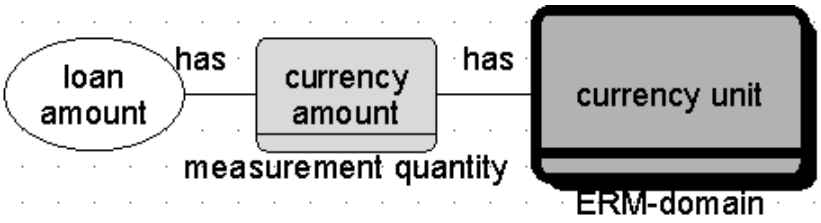


Picture 26: Defining the data type by use of an ERM-domain (1)



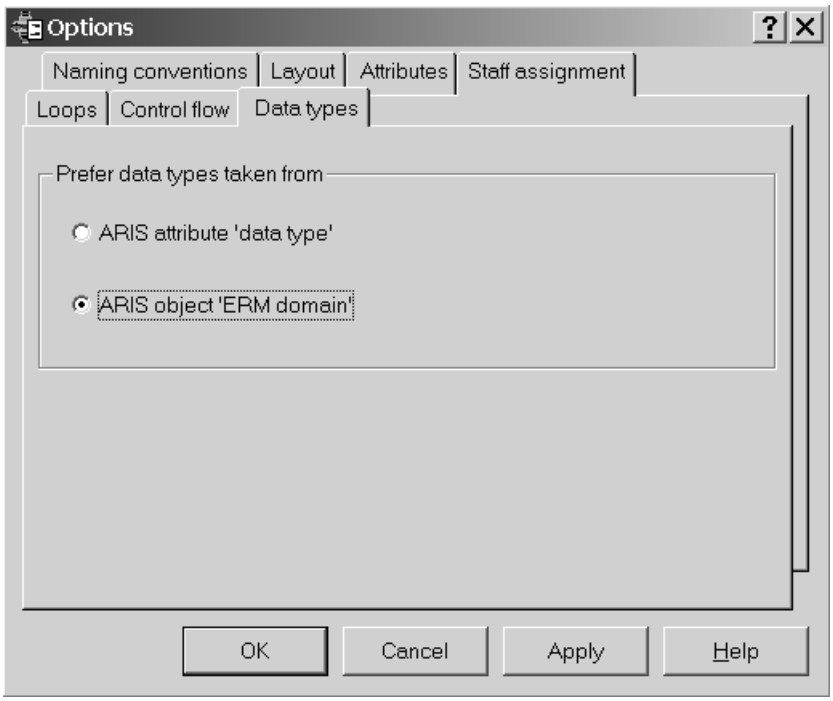
Picture 27: Defining the data type by use of an ERM-domain (2)

Eventually, you may define the data type by use of a „measurement quantity (185)"and an „ERM-domain (20)" (Picture 28 and mapping rule „UNIT_NUM_HAS_DOMAINTYPE" on page 54).



Picture 28: Defining the data type by use of a measurement quantity and an ERM-domain

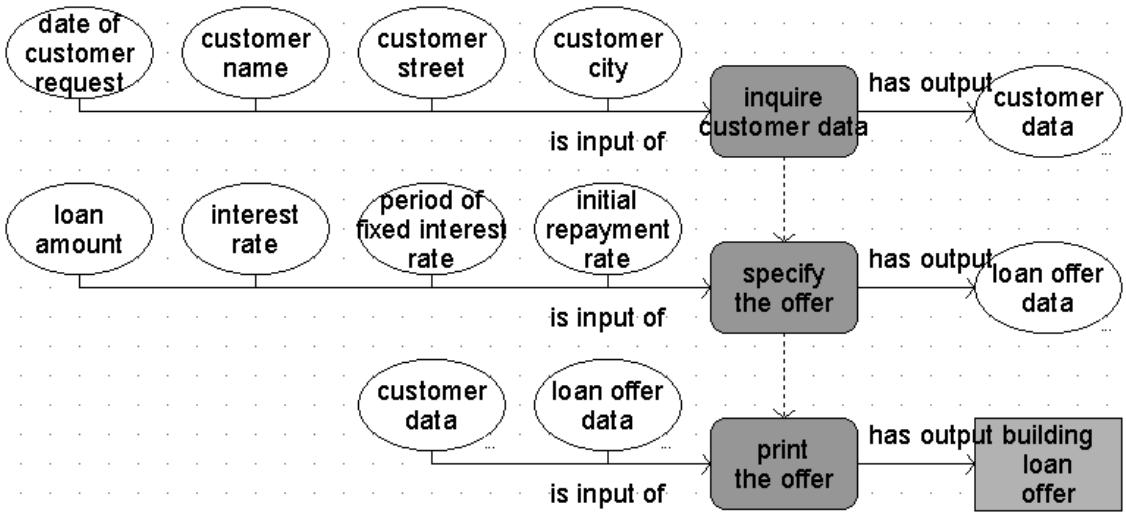
In an ARIS model you may simultaneously define different data types for the „ERM attribute (19)", using both the object definition attribute „Data type" and the „ERM-Domain (20)". In this case you must assist the ARIS Bridge with the translation task and resolve any ambiguous data type definition. You can do this in the options dialog on panel „Data types", where you enter your preference between the options „ARIS attribute ‚data type'" and „ARIS object ‚ERM domain'" (Picture 29). Despite of this choice, you may further use both definition techniques in your ARIS model. The preference choice takes effect in case of inconsistently defined data types, only.



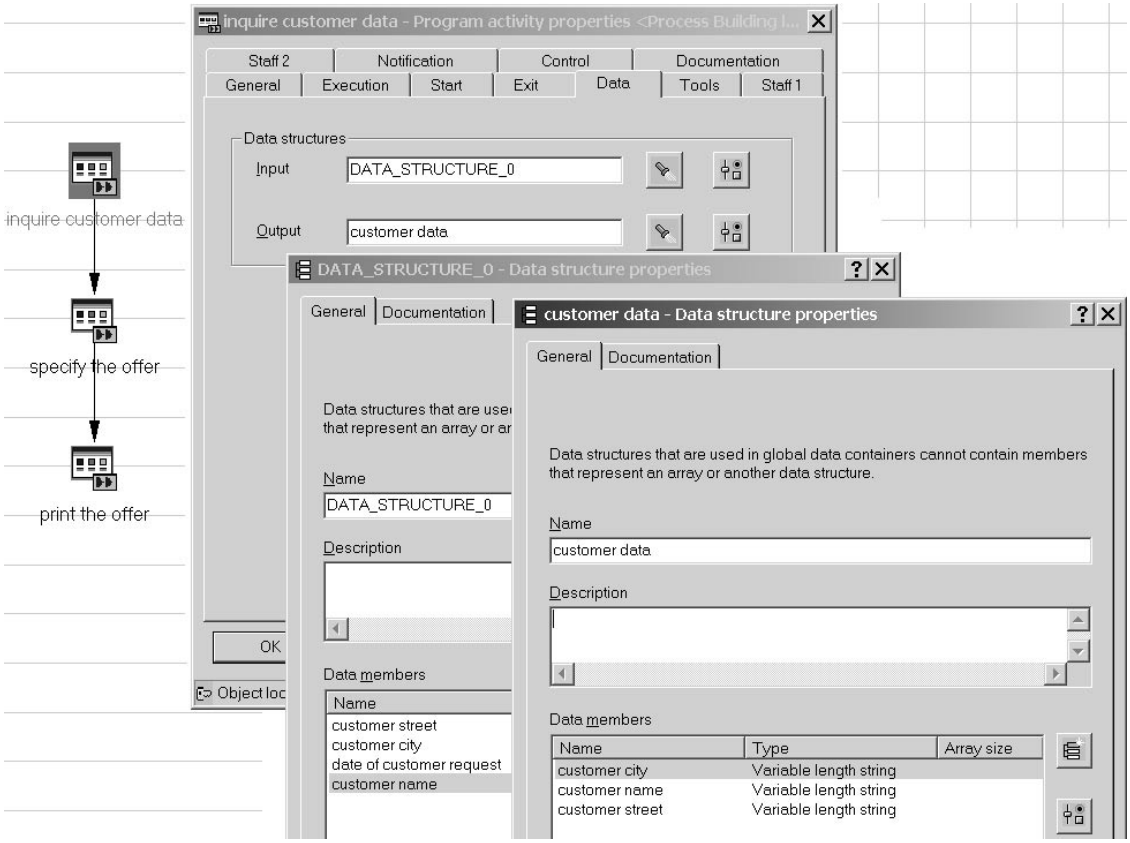
Picture 29: Option used to handle ambiguous data type definitions

Data containers

According to the MQ Workflow terminology, a data structure is also called a „data container", when it is assigned to a workflow activity or an IT resource (program) as data input or data output. Picture 30 shows an eEPC diagram with a function chain that belongs to the offering process of a building loan as well as the relationship connectors between the functions and the necessary data already introduced in Picture 23.

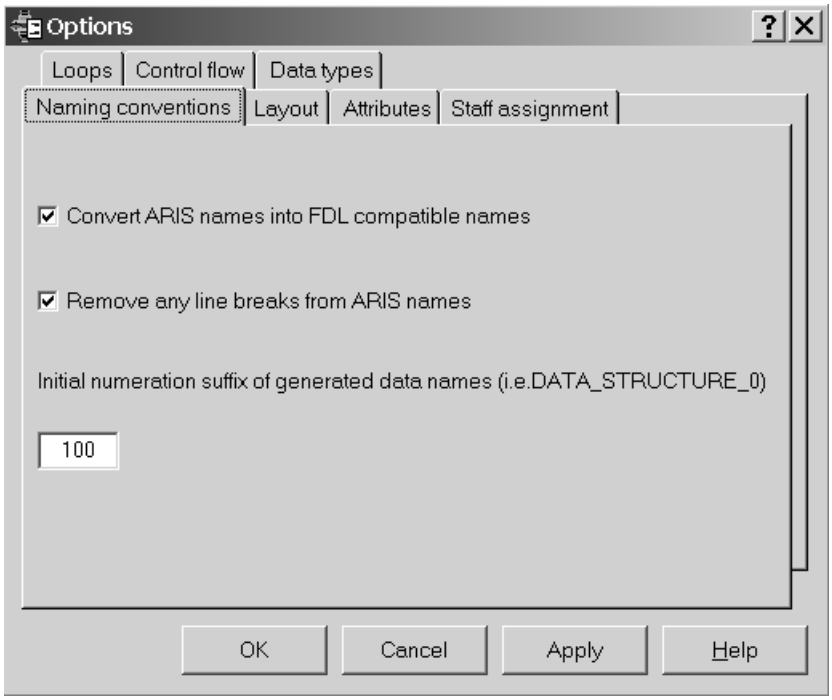


Picture 30: Assigning data structures in an eEPC diagram



Picture 31: Data assignment in the workflow representation

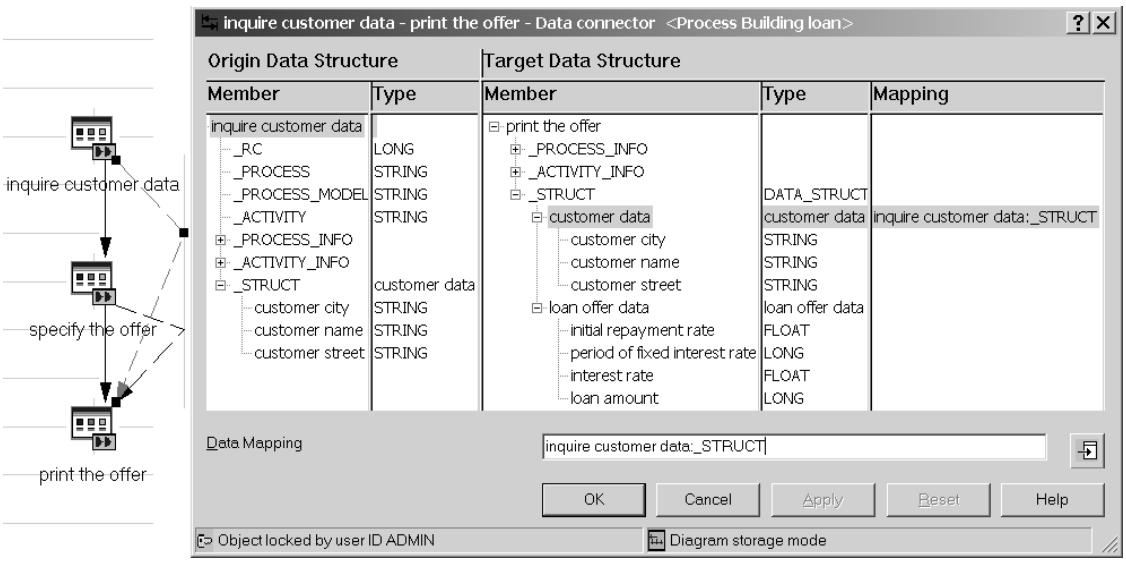
In Picture 31 you find the corresponding representation in the view of MQ Workflow Buildtime with notebooks opened for the activity „inquire customer data“ and the respective data structures assigned to it. The text fields „Input“ and „Output“ on panel „Data“ display the names of the data input and data output structures, respectively. Note that the ARIS Bridge generated a name „DATA_STRUCTURE_0“ for the data input, because there is no composite data object in the eERM diagram (cp. Picture 23) for this data aggregate. The names of generated data structures carry an indexing suffix. You may preset its initial value on the „Syntax“ panel of the options dialog (Picture 32).



Picture 32: Presetting the initial indexing suffix of generated data structure names

Data connectors

„Data connectors" represent the data flow between the activities of an MQ Workflow model. Picture 33 illustrates again the activity chain of the building loan offering, this time with data connectors displayed. Data connectors are always inserted between two workflow activities, if in the ARIS model the first function creates data that are processed as input of the second one. Mapping rule „FUNCTION_SENDS_DATA_TO_FUNCTION" on page 64 describes the translation details of this data flow connection into the MQ Workflow model.

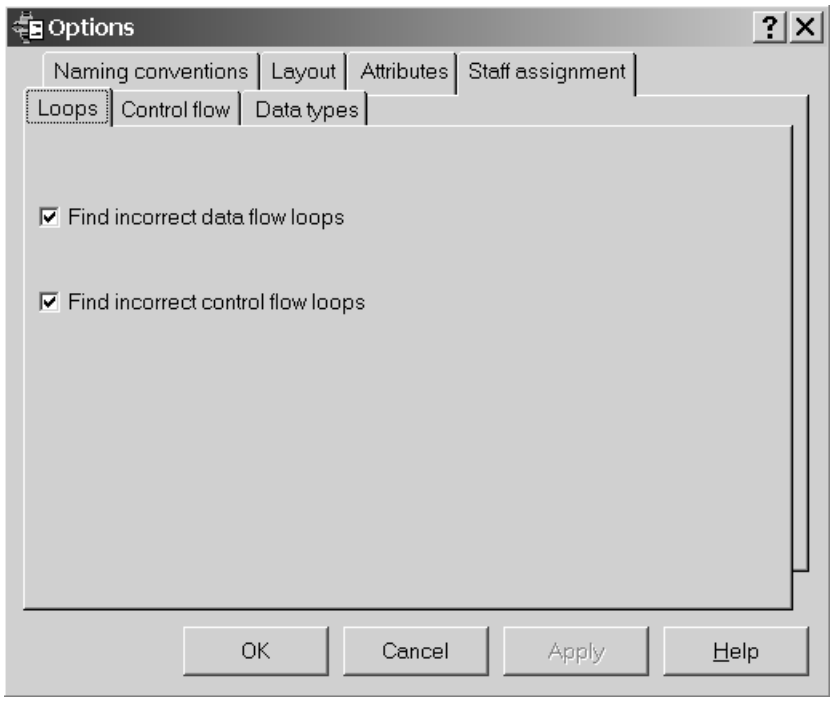


Picture 33: Data connectors and „Data Mapping"

Compare Picture 33 with Picture 31 and you will notice that there are data connectors inserted into the workflow process diagram as dashed-line arrows. The ARIS Bridge created them automatically, while it translates the ARIS model shown in Picture 30. It also created the „Data Mapping"-instruction „inquire customer data:_STRUCT", which you find in table column „Mapping" inside the dialog window opened next to the right. The dialog is displayed, if you double-click with the mouse on the data con-

nector between activities „inquire customer data" and „print the offer". The „Data Mapping"-instruction means, that the content of data structure „customer data" will be copied with all its data elements into the data input container of activity „print the offer", before it gets ready to start. The ARIS Bridge checks the data containers, which you find at both ends of each data connector, for shared data elements and inserts the respective „Data Mapping"-instructions into the FDL file.

In general, data loops or back back branches of data flow are not allowed in an MQ Workflow model. Panel "Loops" of the options dialog lets you specify, whether you want them to be searched for while translating and to be reported with a respective warning in the log file (Picture 34).

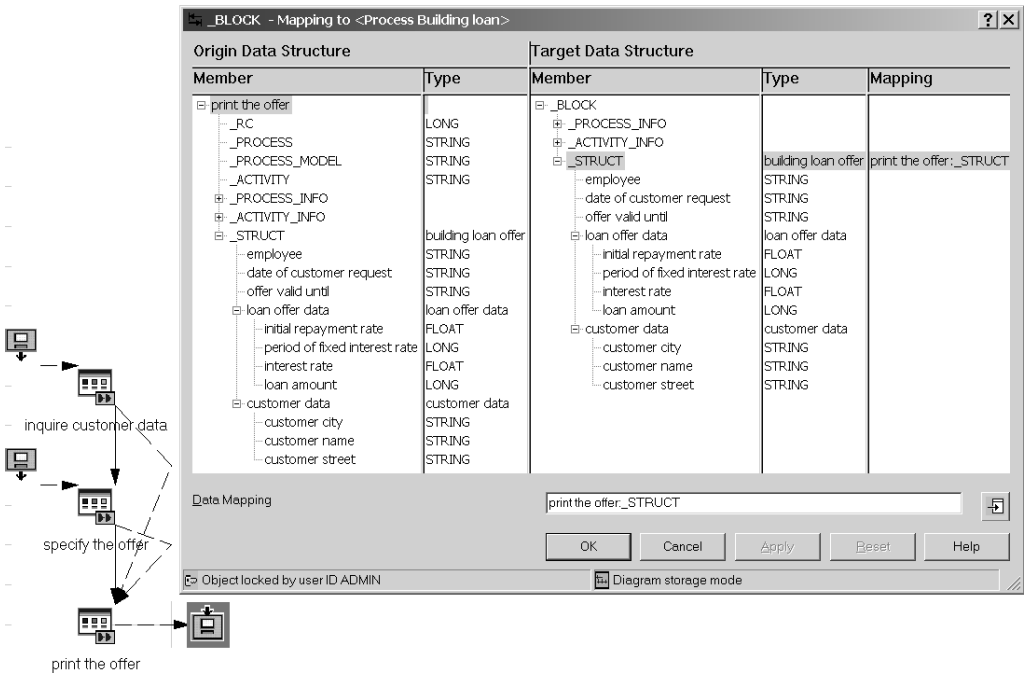


Picture 34: Option „Find incorrect data flow loops"

Data sources and data sinks

At the end of the subject „Data view", Picture 35 shows a third and now completed view of the process "building loan" after its translation into an MQ Workflow model. This time it contains in addition the previously hidden „SOURCE" and „SINK" icons, which designate so-called data sources and data sinks. They represent data structures and are constructed from such data objects of the ARIS model in Picture 30, which exclusively belong either to a function's data input or to it's data output, respectively. Each „SOURCE" icon comprises the total input data flow into the process „Building loan". Accordingly, the „SINK" icon represents a data structure, which comprises the total of output data flow leaving the process. This means, that there is exactly one data source and one data sink in every MQ Workflow process. Repeated representations of „SINK" and „SOURCE" icons in an MQ Workflow process diagram stand for multiple symbol occurrences of the same data structure. [17]

[17] „SINK" and „SOURCE" are the only object types, which may have more than one „occurrence" in an MQ Workflow diagram.



Picture 35: Data sources and data sinks

Data connectors link the „SINK“ and „SOURCE“ icons to the activities, which create or respectively receive the data. The dialog next to the right in Picture 35 displays the contents of the related data containers and the “mapping” of data.

Resource view

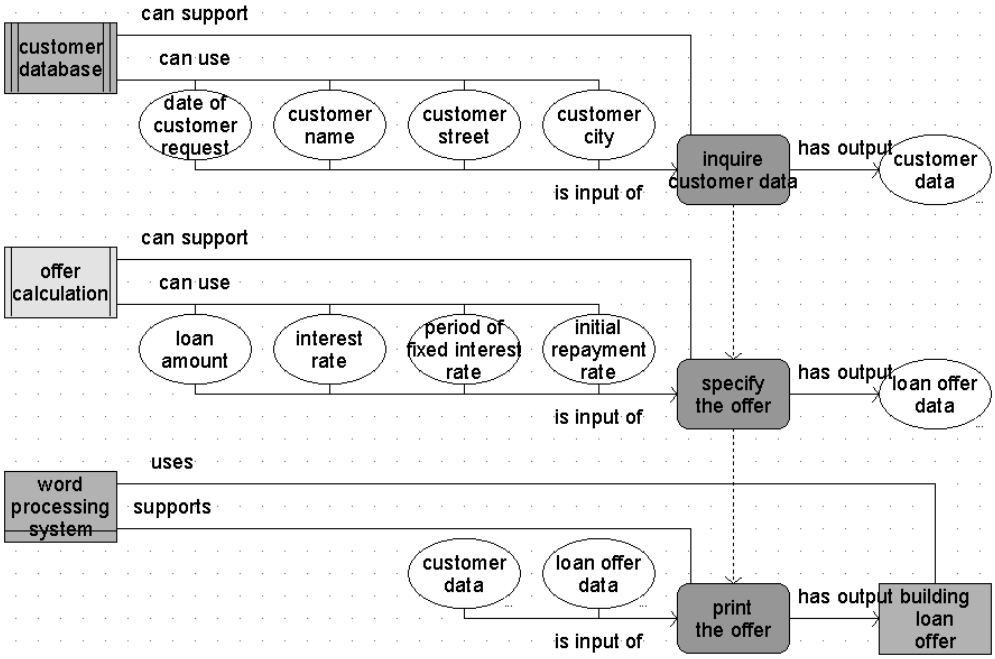
The above considered business processes revealed a pure “manual” natur. Now, we would like to improve them with the support of IT resources. In „MQ Workflow“ terminology, activities are called “program activities”, if they automatically start a program. Programs have to be registered with their properties in a workflow model. The ARIS Bridge translates the following ARIS object types into “Program”-objects (cp. mapping rule „PROGRAM“ on page 55):

Application system type (6)
IT function type (105)
Module type (37)

Table 6: ARIS objekt types, which are mapped to workflow objects of type “PROGRAM”

Picture 36 refers again to the “loan offer” sample and shows supplementary ARIS objects for IT resources. You can lookup the following mapping rules for the ARIS connection types that the ARIS Bridge supports as relationships between IT resources and data objects on the one hand and functions on the other hand:

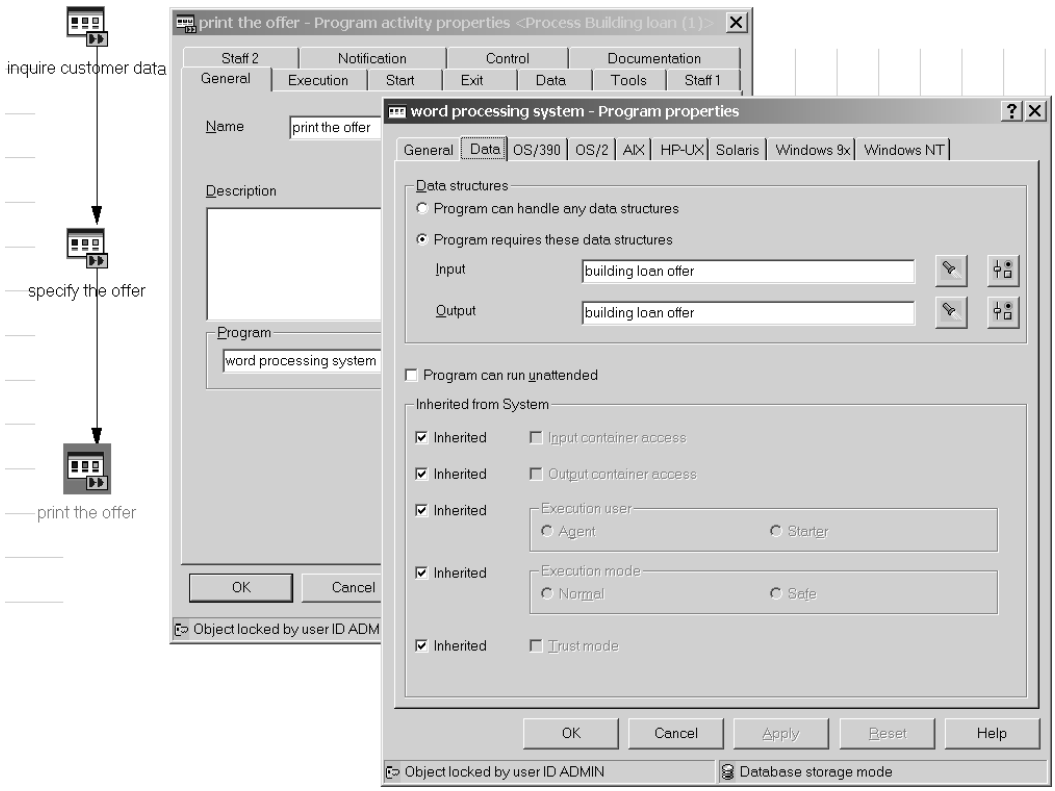
- „ACTIVITY_PROGRAM“ (page 56)
- „PROGRAM_SENDS_DATA“ (page 65)
- „PROGRAM_RECEIVES_DATA“ (page 65)



Picture 36: Integrating IT resources into the ARIS funktion chain

In a workflow model, IT resources appear as entries to the properties notebooks of the activities. In Picture 37, you find the resource „word processing system“ on notebook panel „General“ of activity „print the offer“ in text field „Program“. Next to the right, the „Program properties“ panel „Data“ displays the data structure „building loan offer“ in text fields „Input“ and „Output“.[18]

[18] Unfortunately, there are no suitable ARIS connection types applicable for the object types „Application system type (6)“,„IT function type (105)“, and „Module type (37)“, that would you allow to discriminate data input and data output.



Picture 37: Assigning programs in a workflow model

Process view

In the previous sections, you already became acquainted with some aspects of process view. You learned, which ARIS connection types you may use to relate the respective objects of the function view, the organization view, and the data view. But, this discussion was confined to the structural properties of a business process model. This section investigates the dynamic properties, which are required to perform a process.

Control connectors

The process diagrams of an MQ Workflow model display special connector types, called “control connectors”, which represent the flow of control. You already met them in the previous samples. In the following, you will learn which chains of relationships in an ARIS model the ARIS Bridge can convert into corresponding control connectors.

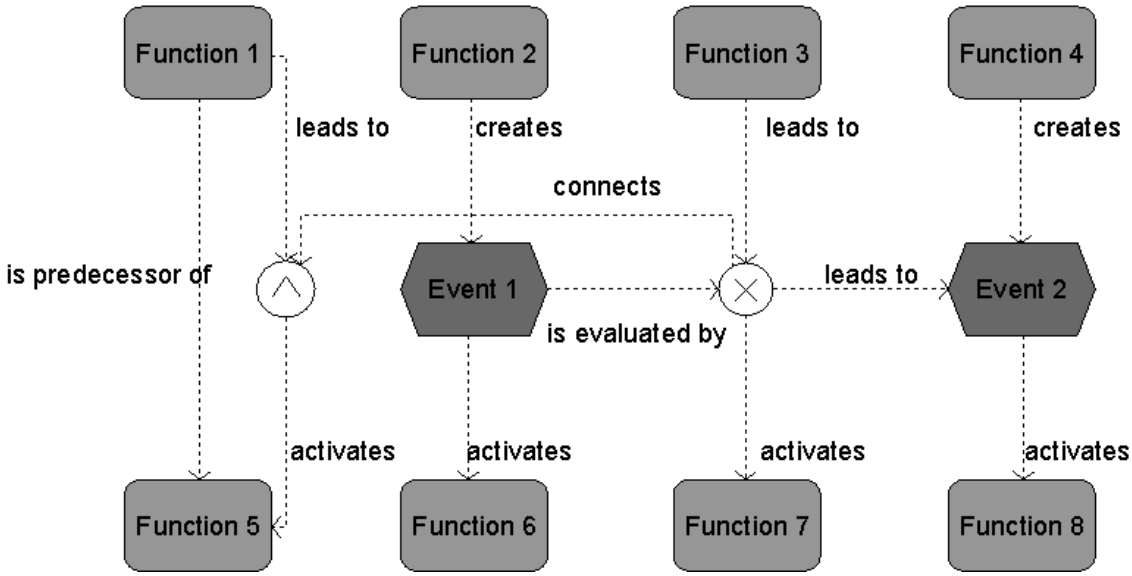
You know the ARIS connection type „is predecessor of (118)“ from eEPC models, which looks most similar to an MQ Workflow control connector. But, more frequently, you find objects of type „Event (18)“ and „Rule (50)“ as elements of control flow in an ARIS model. Sometimes, they may occur as nodes of elaborate control flow networks. Note that, in MQ Workflow, there are no objects equivalent to events and rules. On the other hand, you can very easily describe the logic of control flow with logical expressions (cp. page 41), which you assign as attributes to the workflow activities and control connectors.

Picture 38 shows a schematic ordering of all possible connections that may occur between ARIS object types „Function (22)“, „Event (18)“ und „Rule (50)“, in order to represent the flow of control. Note that it was intentionally not cared about, whether the orderings are also useful. The ARIS Bridge translates this network according to the following rule: A single control connector will be inserted between a pair of activities, if you can find one or more control flow paths between the corresponding ARIS functions (Picture 39). For instance, there is a control connector between „Function 2“ und „Function 5“, because the respective ARIS model shows the following control flow path:

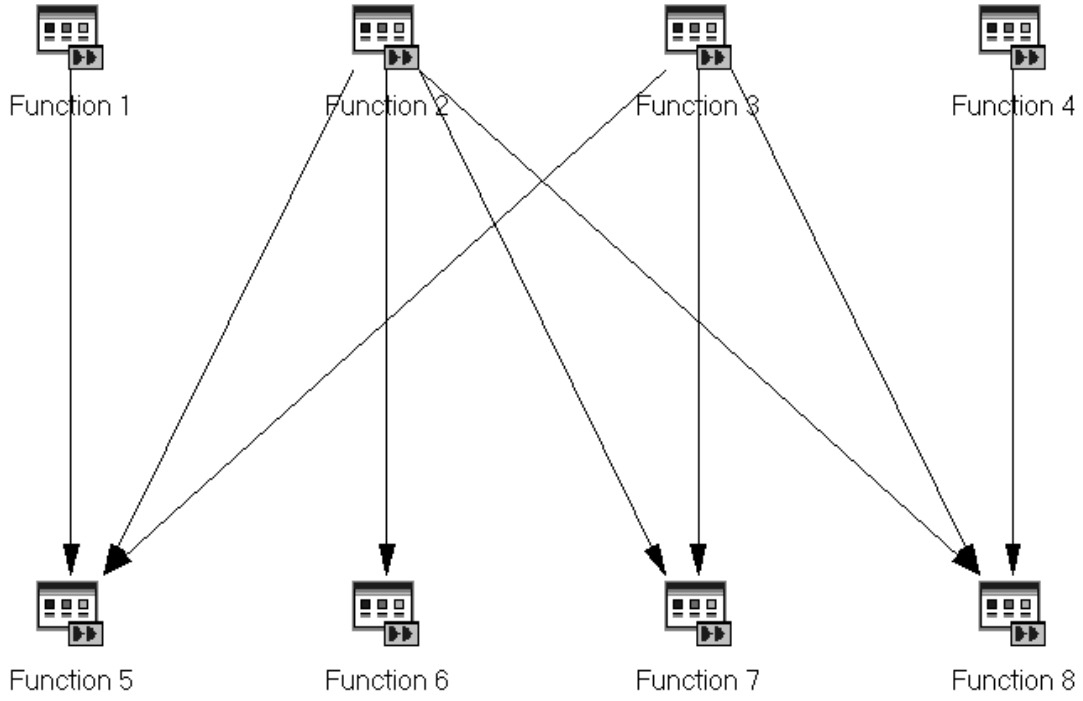
„Function 2“ → „creates“ → „Event 1“ → „is evaluated by“ → „XOR-Rule“ → „AND-Rule“ → „Function 5“

Note that there is only one control connector between „Function 1“ and „Function 5“ in the workflow process diagram, although you can find two paths:

1. „Function 1“ → „is predecessor of“ → „Function 5“
2. „Function 1“ → „leads to“ → „AND-Rule“ → „activates“ → „Func tion 5“



Picture 38: Schematic ordering of all possible connections between ARIS object types „Function (22)“, „Event (18)“ und „Rule (50)“



Picture 39: Translating the control flow network of Picture 38 into an MQ Workflow model

Note that even missing control connectors assign an implicit meaning to a workflow activity:

- Activities without entering control connectors are called „start activities”. After the start of the workflow process, they will be automatically added as workitems to the „worklists” of the assigned persons.

- Activities without leaving control connectors are called „end activities". A process is not finished before all contained end activities are finished.

According to this definitions, in Picture 39 „Function 1" until „Function 4" are start activities and „Function 5" until „Function 8" are end activities.

Start and end activities represent the control flow interfaces of a workflow process. You should notice this meaning, because the ARIS Bridge cannot correctly interpret other modelling conventions (i.e. using the "process interface"-symbols) due to ambiguous semantics.

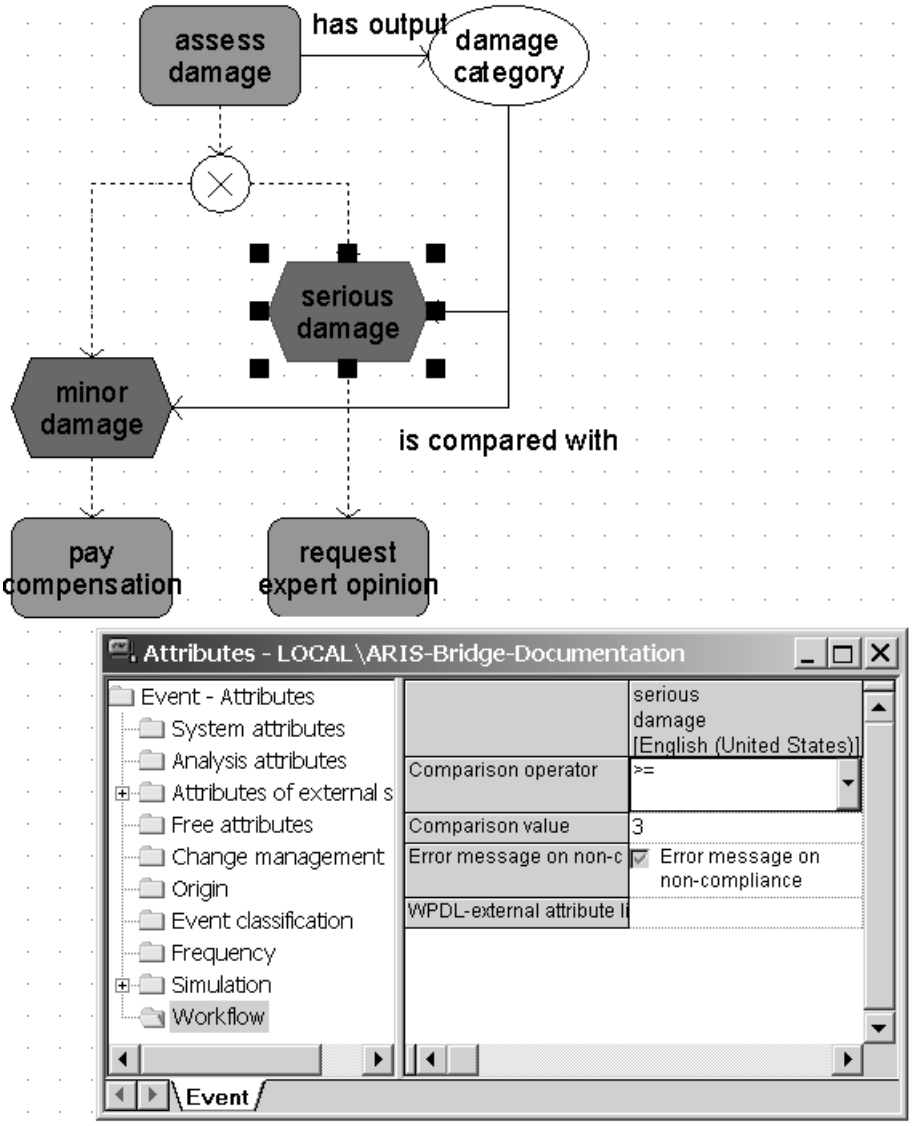
If you are familiar with modelling of MQ Workflow processes, you will know that you cannot explicitly represent control flow loops. Similar to the prohibited data flow loops, you may specify on panel „Loops" of the options dialog, whether you want the ARIS Bridge to search for incorrect control flow loops while translating and to report them with a warning in the translation log file (cp. page 36).

Control flow conditions

Transition conditions

The transition condition of a control connector describes the logical prerequisite for a flow of control between the activities that are linked to the connector.

The following example deals with the case that a security company may have to decide, either to immediately pay compensation for a damage, due to minor severity, or to request expert opinion first. Picture 40 shows the respective control flow branching, modelled in an eEPC diagram. The case is settled by comparing the found damage category with a given risk threshold. You can specify the respective comparison operators and comparison values inside the attribute group „Workflow" of the "Event (18)"-objects.



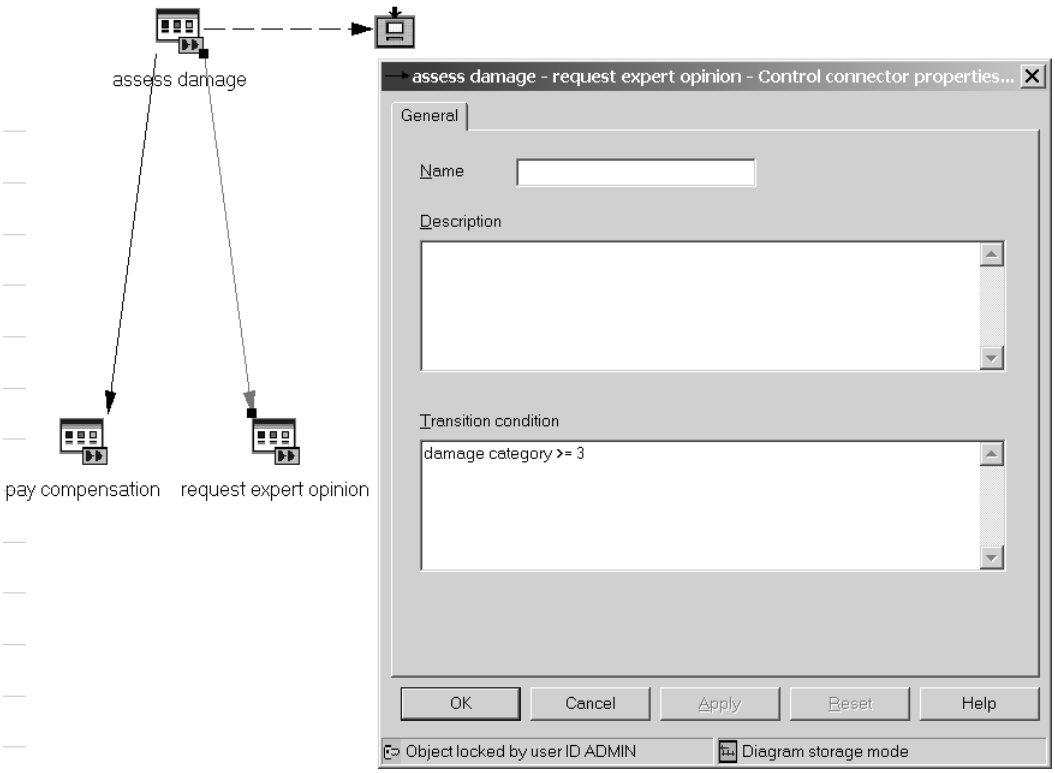
Picture 40: Transition condition with data comparison

After the ARIS Bridge translated the ARIS model, you will find the transition conditions again in the properties notebooks of the control connectors. As an example, Picture 41 shows the transition condition of the control connector between the activities „access damage“ and “request expert opinion“. The respective mapping rules are called:

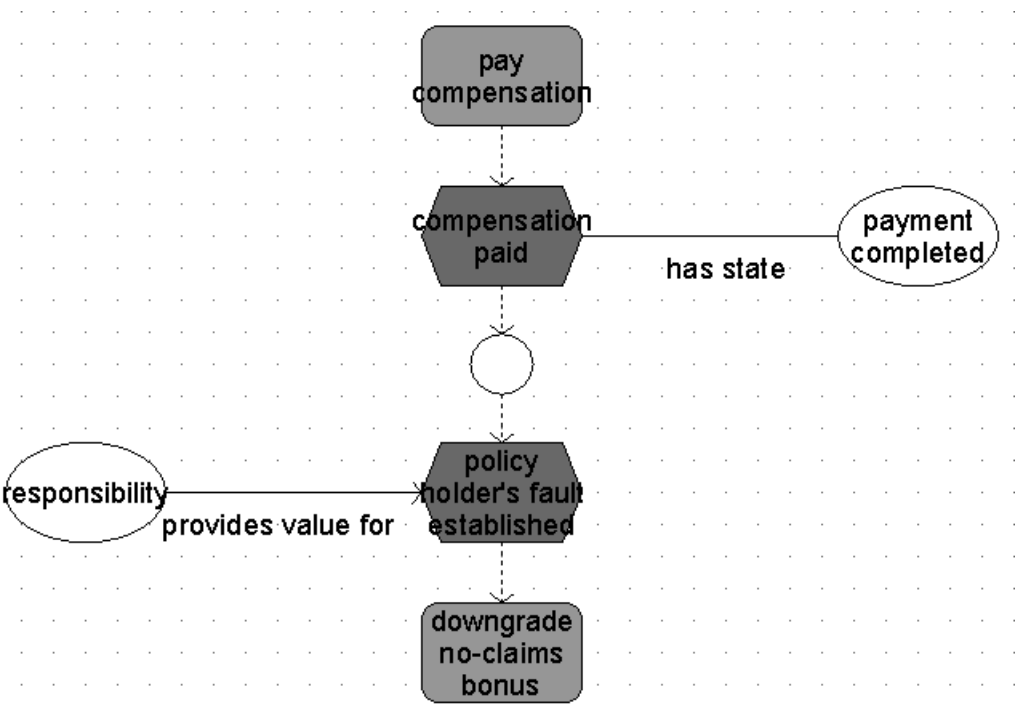
- „EVENT_IS_COMPARED_WITH_DATA“ (page 63)
- „EVENT_COMPARE_OPERATOR“ (page 58)
- „EVENT_COMPARE_VALUE“ (page 59)

Please, note that the ARIS Bridge cannot take into account the “Rule (50)”-objects of ARIS models, while it determines the control flow conditions of the translated MQ Workflow model. [19] Nevertheless, you should continue to use them for graphical illustration purposes.

[19] As an exception from this statement, the rule-object „AND operator“ is considered for „start conditions“ (cp. page 46).



Picture 41: Reprsenting a transition condition in the workflow model



Picture 42: Transition condition with a logical AND-connection (ARIS model)

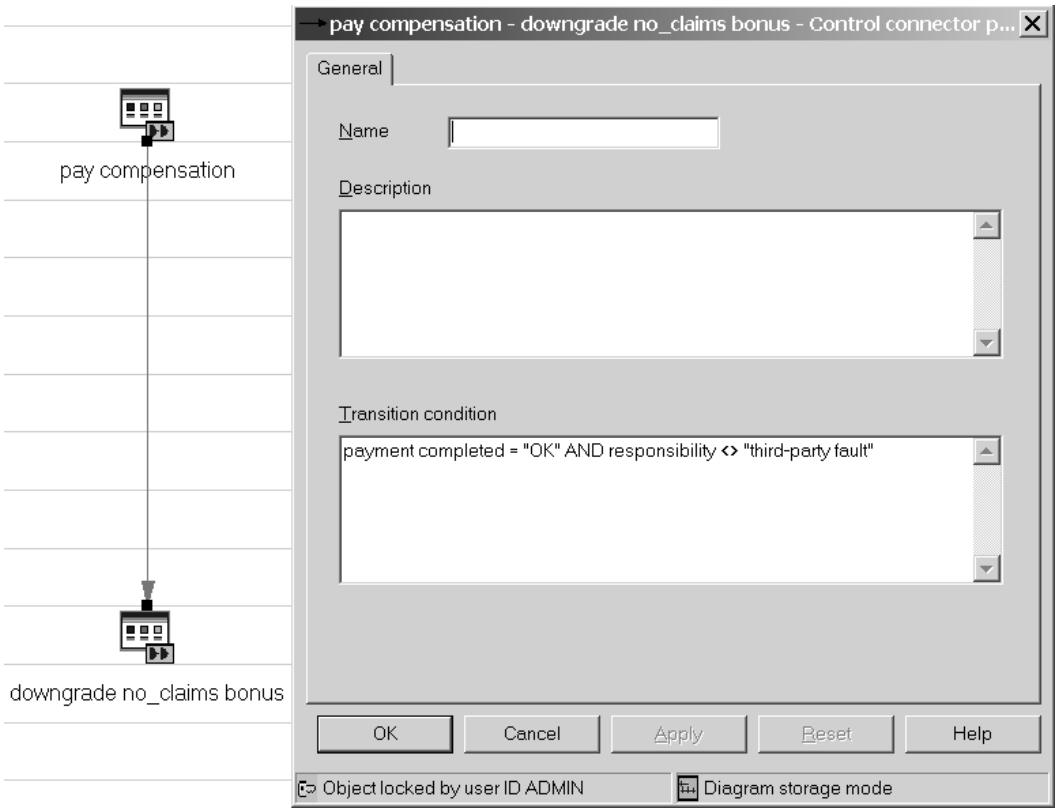
You can also model more complex conditions that are composed of partial expressions with logical „AND“ or respective „OR”-operators.

Picture 42 shows how you can represent in an ARIS model an AND-connection in the transition condition. You may consider the sequentially ordered events as the series connection of imaginary valves: The control flow can reach the next activity only, if both valves are opened.

Here are to additional comments to Picture 42:

1. The „Rule (50)"-object inserted between the two events is a „Dummy"-object. It is required, because there is no ARIS connection type available that might let you directly connect two „Event (18)"-objects.
2. The picture demonstrates that you may connect „Event (18)"-objects in different ways to data objects.[20]

In Picture 43 you find how the above construction translates into the workflow model.

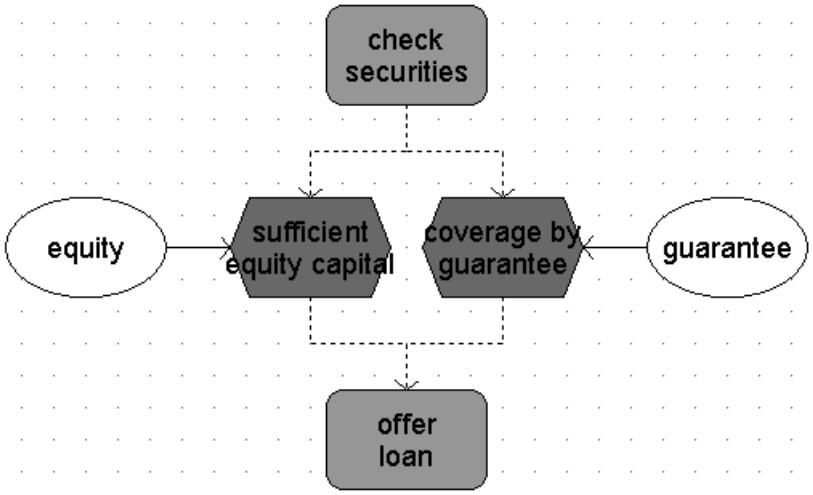


Picture 43: Transition condition with a logical AND-connection (workflow model)

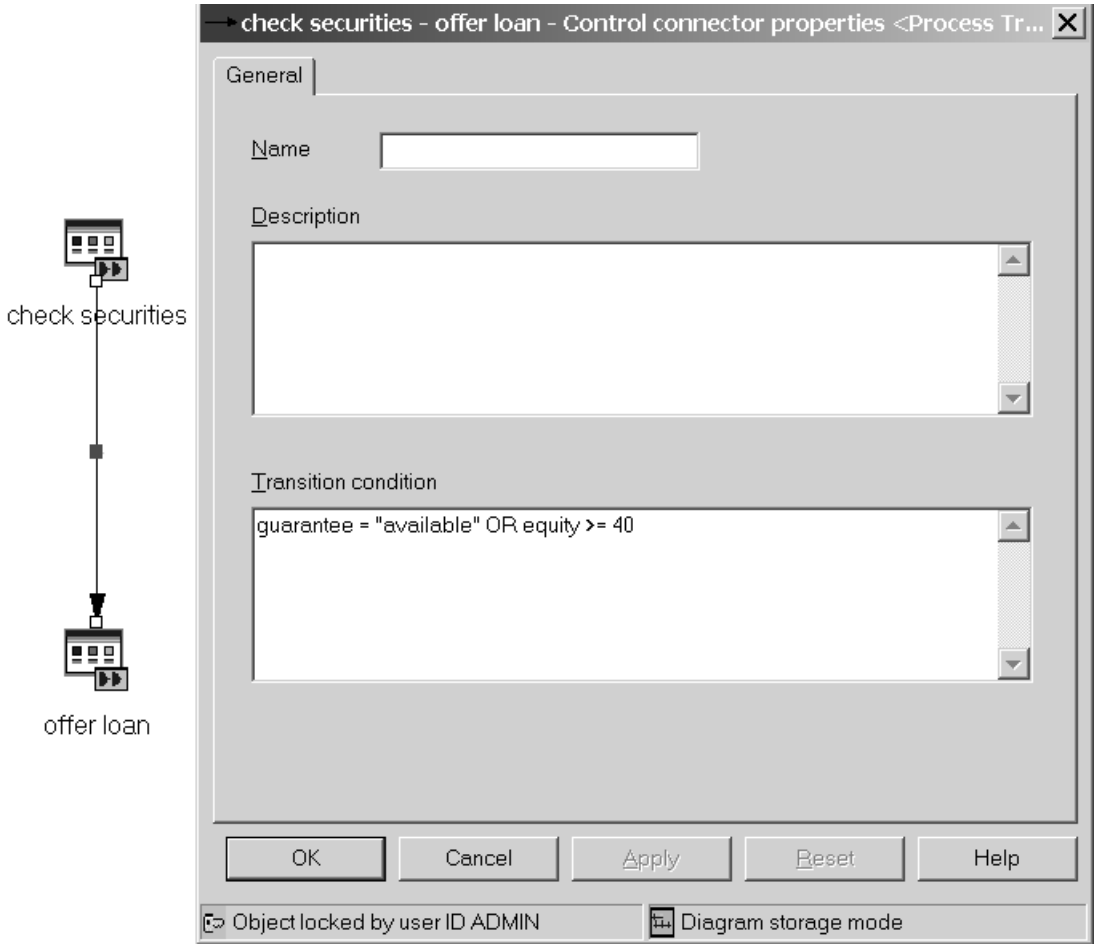
Picture 44 shows the ARIS representation of an OR-connection inside of a transition condition. In this arrangement, there are two control flow paths with a single „valve" (transition condition) assigned to each of them. You realize that it is sufficient to open one of either valves (condition is „true"), in order to direct the control flow to the next activity „offer loan". Picture 45 shows how, in the workflow model, a single control connector with an OR-typed transition condition summarizes the two control flow paths.

You may combine the control flows with AND or respective OR-connections in the ARIS model, as you like. The ARIS Bridge will translate control flow networks of any complexity into an equivalent workflow model.

[20] Please, also note that the semantics of events is always related to data and not to functions. An event describes the occurrence of a certain state of data (condition). That is why, in an ARIS diagram, there is no need to place „Events (18)" right after the „Functions (22)".

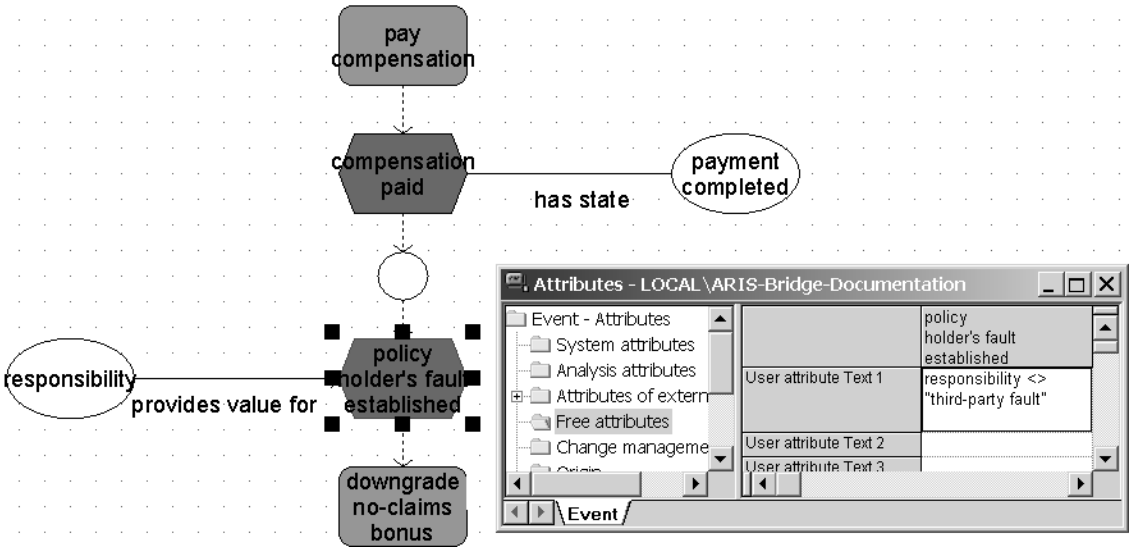


Picture 44: Transition condition with a logical OR-connection (ARIS model)



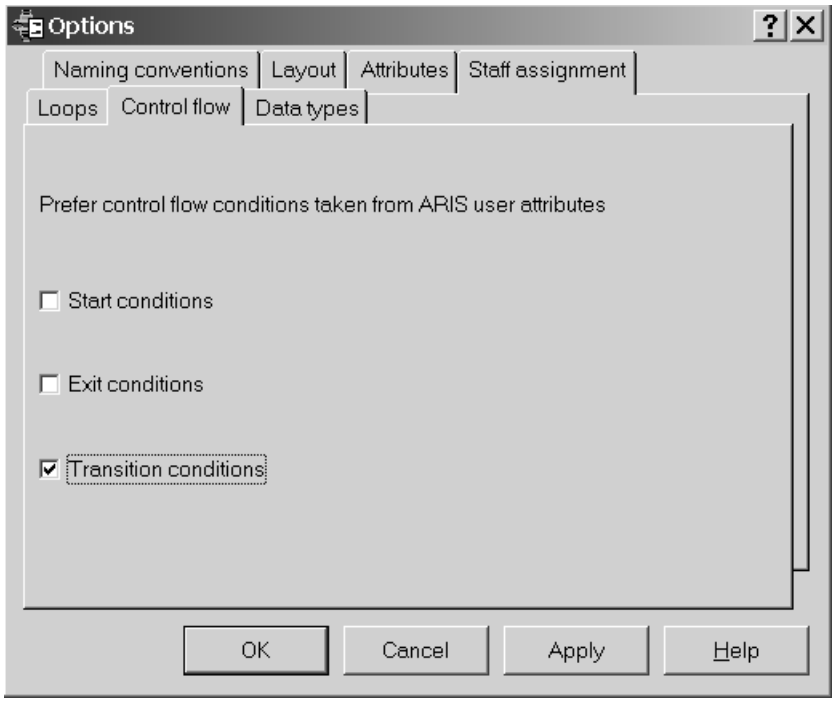
Picture 45: Transition condition with a logical OR-connection (workflow model)

In case that you consider this manner of graphical representation of control flow as too complex or inconvient, then you also may directly enter the logical expression components of the transition conditions into the „User attribute Text 1“-field of the related „Events (18)” (cp. mapping rule „CONTROL_CONNECTOR_CONDITION“ on page 57). On the other hand, it is your own responsibility to observe the correct syntax rules, according to the MQ Workflow conventions.



Picture 46: Specifying a transition condition in field "User attribute Text 1"

Provided that you simultaneously used both description techniques, you should decide on panel „Control flow" of the ARIS Bridge options dialog, which one you prefer for the translation in conflicting cases (Picture 47).



Picture 47: Option resolving ambiguous specifications of control flow conditions

Start conditions

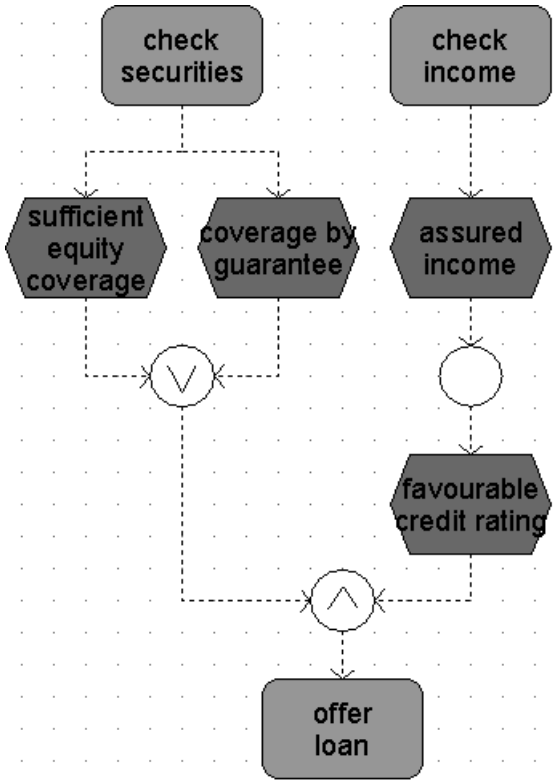
At runtime of a workflow process, an activity turns into state „ready" (activity is ready to perform), if the transition conditions of all incoming control connectors evaluate to „true". The „start condition" of a workflow activity determines the respective evaluation policy. The available policies are:

„At least one incoming connector true" — The transition condition of one of the incoming control connectors evaluates to „true".[21]

[21] Implicit OR-connection of the transition conditions.

„All incoming connectors true“ The transition conditions of all incoming control connectors evaluates to „true“.[22]

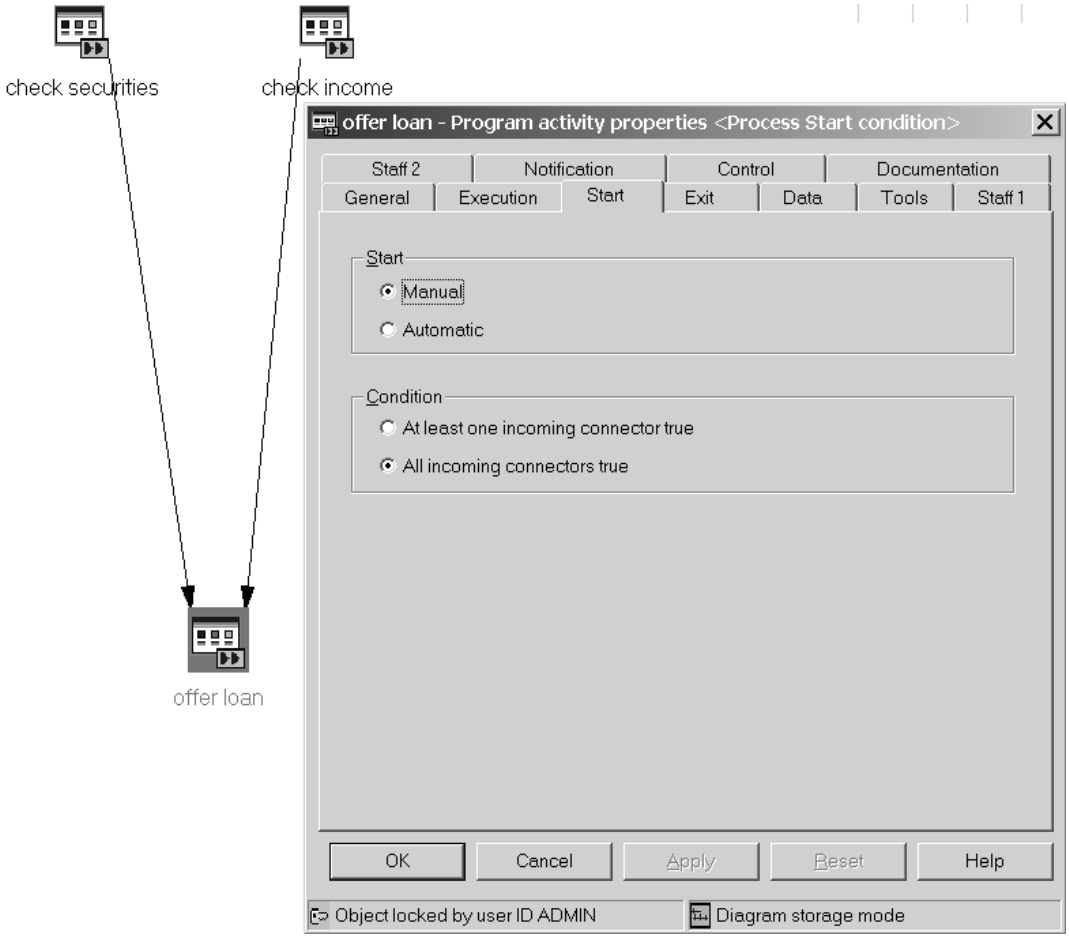
The value „All incoming connectors true“ will always be assigned to the start condition of a workflow activity, if the last common „Rule (50)”-object shared by all incoming control flow paths is an AND-connector (cp. mapping rule „FUNCTION_RECEIVES_CONTROL_FROM_RULE“ on page 63). Picture 48 shows a respective example. [23] In Picture 49 you find the start condition on notebook-panel „Start“. All other cases will evaluate to the value „At least one incoming connector true“.



Picture 48: ARIS modelling of an "All incoming connectors true" start condition

[22] Implicit AND-connection of the transition conditions.

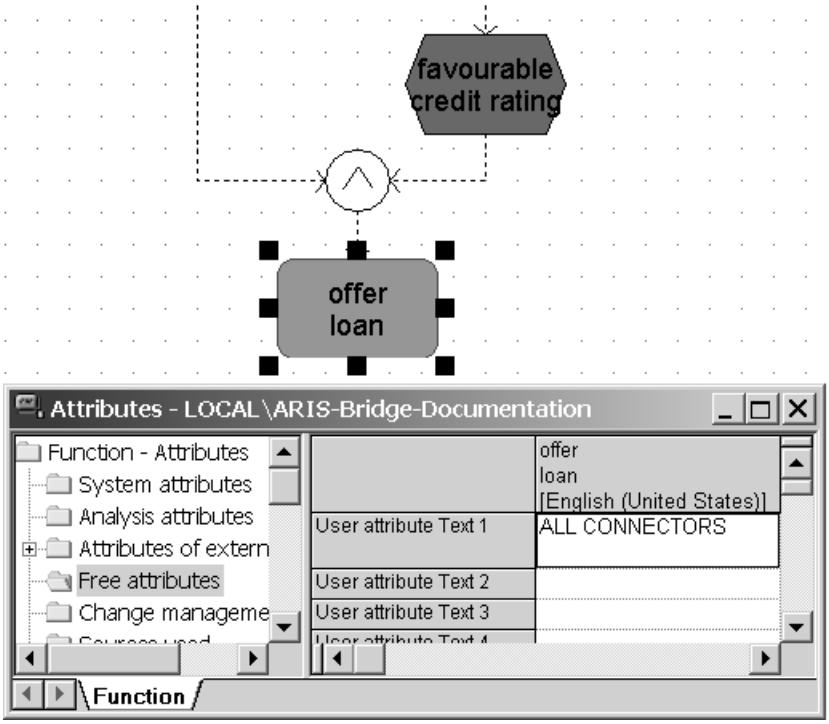
[23] You do not violate the mapping rule, if you would insert another event-symbol between the AND-connector and function „offer loan“.



Picture 49: Start condition "All incoming connectors true" in the workflow model

Similar to the transition conditions of control connectors, you also have the option to enter the start condition directly into „User attribute Text 1“ of the ARIS function (cp. Picture 50 and the mapping rule „ACTIVITY_START_CONDITION“ on page 56). But, it is again your own responsibility to observe the correct syntax rules, according to the MQ Workflow conventions. Respectively, on panel „Control flow” of the ARIS Bridge options you decide, whether you prefer this representation kind for the translation in case of conflict. (cp. Picture 47 on page 46).

Startbedingung:	**FDL-Schreibweise:**
„At least one incoming connector true“	ALL CONNECTORS
„All incoming connectors true“	AT_LEAST_ONE CONNECTOR



Picture 50: Start condition specified as a "free attribute"

Exit conditions

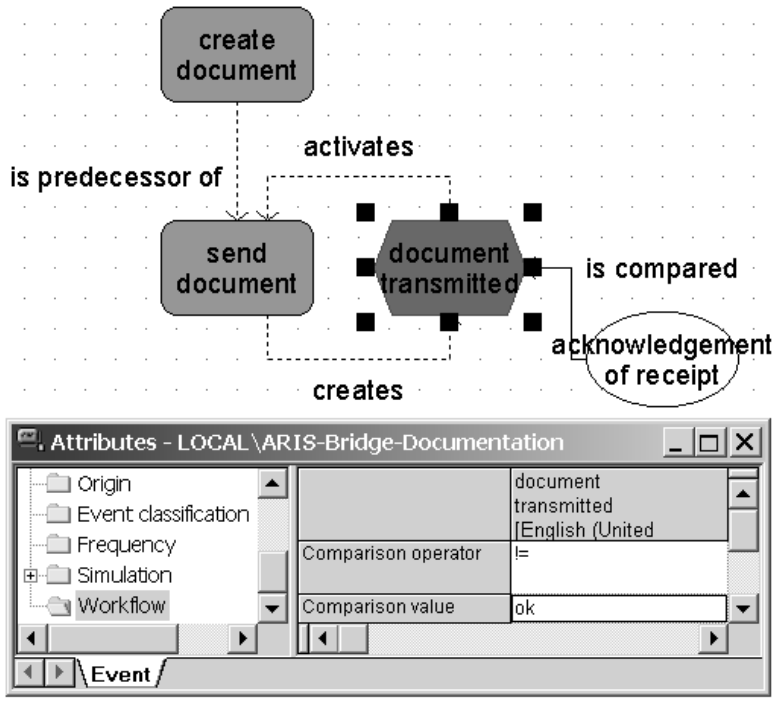
The exit condition of a workflow activity is a logical expression that must evaluate to „true", in order to terminate the activity. The evaluation result „false" means, that the activity has to be repeated.[24]

That is to say, the exit condition is an implicit representation of a control flow loop, which otherwise would be forbidden, if you modelled it explicitly in the MQ Workflow model. You can represent an exit condition in an ARIS model by way of exception and draw an otherwise „incorrect" control flow loop from the exit of the ARIS function back to its entry. You model the condition itself in the same manner as you learned about it in section „Transition conditions" from page 41 on (Picture 51). Picture 52 shows that, after translation, you can find the exit condition on panel „Exit" in the properties notebook auf activity „send document". Please, note that it equals the negated transition condition of the control flow loop, because the value „true" of the evaluated exit condition means that the control flow does not branch back to the start of the activity!

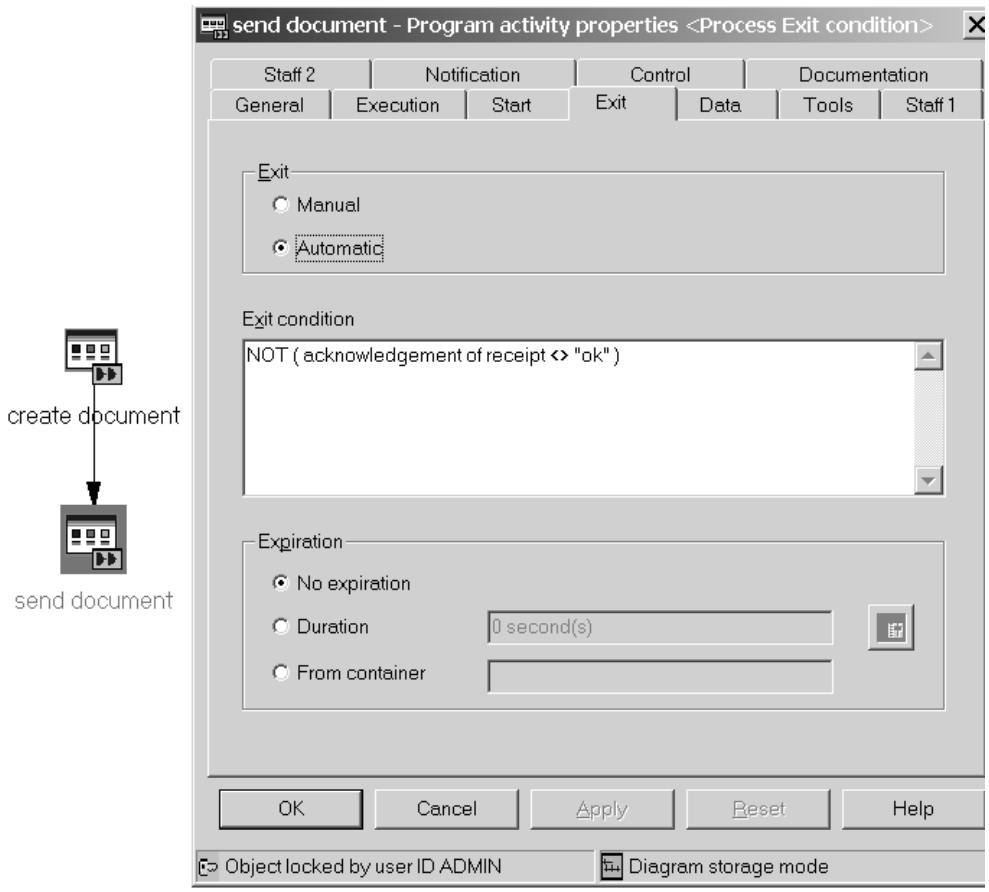
Similar to the modelling techniques of transition conditions, you may compose complex exit conditions from simpler ones by means of AND or respective OR-operators (cp. „Transition conditions" from page 41 on).

Similar to the start condition, you may replace again the graphical modelling by entering the exit condition into the text field of a free attribute („User attribute Text 2") (cp. mapping rule „ACTIVITY_EXIT_CONDITION" on page 56). Regard again the correct writing, according to FDL syntax.

[24] That is the implicit representation of „control flow loops" in an MQ Workflow model.



Picture 51: ARIS representation of an "exit condition"

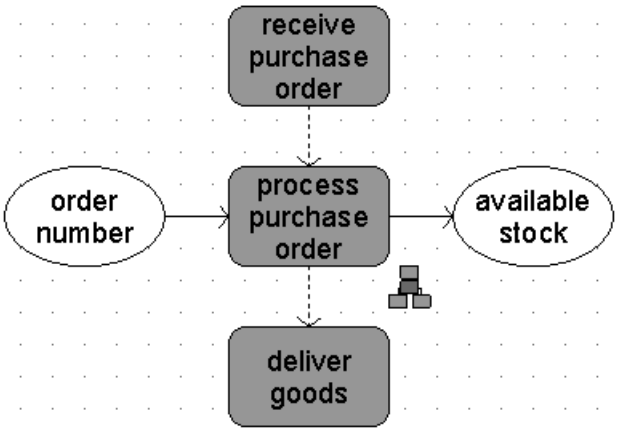


Picture 52: Exit condition of a workflow activity

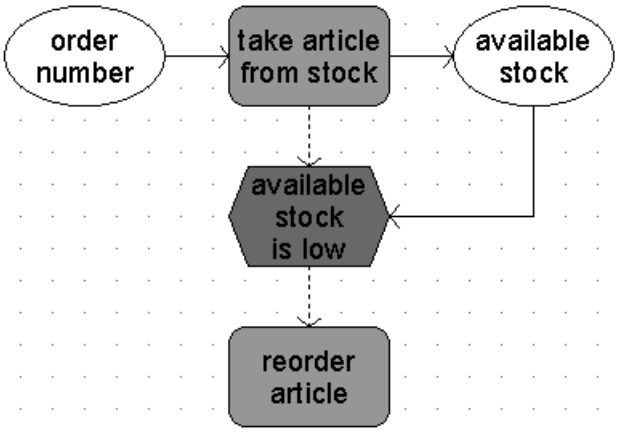
Subprocesses

A workflow process becomes a subprocess, if it implements the activity of another (superior) process. Activities, which invoke a subprocess, are called „process activities", according to MQ Workflow terminology. [25]

In an ARIS model, the ARIS Bridge translates each function with an „assignment"[26] into a process activity, provided that the assigned model can be translated into a „process" (cp. page 23).[27]



Picture 53: ARIS function with an "assignment"



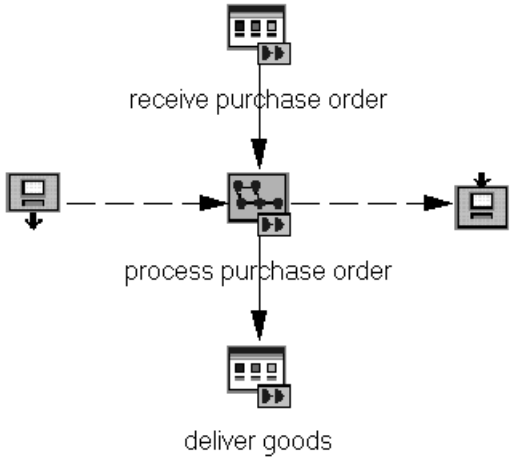
Picture 54: eEPC model assigned as a "subprocess"

Picture 53 illustrates a simple example. The small icon to the right and below of function „process purchase order" indicates that it has another ARIS model assigned. You open the respective eEPC diagram with a double-click (Picture 54). The subprocess takes over „control", as soon as function „process purchase order" is started. The data objects „order number" and „available stock" occur in both diagrams. This means, that they contribute to a vertical data flow: „order number" is passed as „data input" to the subprocess. „available stock" is returned as „data output" to the parent process.

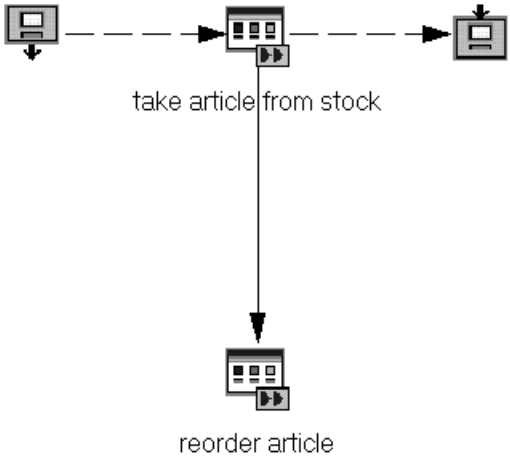
[25] Accordingly, activities, which are implemented by DP-applications (programs), are called „program activities". But, a third available activity type, called „block activity", does not occur in the workflow models, that the ARIS Bridge translates.

[26] Other kinds of modelling the branching into a subprocess are not supported.

[27] For each function only one assignment will be considered. A warning occurs in the log file for any additional assignment.



Picture 55 Workflow process containing a "process activity"



Picture 56 Workflow subprocess

In Picture 55 and Picture 56 you find the workflow processes derived from the ARIS models. A special icon marks activity „process purchase order“ as a „process activity”. The SINK and SOURCE icons represent data structures, which correspond to the data containers passed to the subprocess (cp. section „Data view“ from page 30 on).

Chapter 5 Mapping rules

Preliminary remarks

The ARIS Bridge processes three different kinds of mapping rules, while it translates the ARIS models:

1. **Object mapping**
2. **Attribute mapping**
3. **Connection mapping**

The rules are described below with the following formats:

Object mapping format

Object mapping rules describe the ARIS object definitions which the objects of an MQ Workflow model are derived from. Table 7 displays the representation of this rule type. Multiple entries in the field „object type" mean alternate kinds, how you can model the same MQ Workflow object with an equivalent ARIS object.

Rule name	PROGRAM
Rule type	Object mapping
Object type[28]	Application system type (6) IT function type (105) Module type (37)

Table 7: Object mapping format

Attribute mapping format

Attribute mapping rules describe immediate correspondence[29] between ARIS attributes and object properties inside an MQ Workflow model. Table 8 illustrates the representation format.

Rule name	ACTIVITY_DESCRIPTION
Rule type	Attribute mapping
Attribute type	Description/Definition (9)

Table 8: Attribute mapping format

Connection mapping format

Connection mapping rules describe, how simple connections and/or connection chains between ARIS object definitions are translated into corresponding constructs or properties inside an MQ Workflow model. In an ARIS model, arrows between objects represent the connections. In this context, a path through a connection network as called connection chain. A table with four columns is used to represent this rule type (Table 9 and Table 10). The headings above the columns have the following meanings:

[28] Numbers put in parentheses behind the sample table entries designate ARIS type numbers.

[29] Note that not each property of an MQ Workflow object has a corresponding ARIS attribute. But, some of them are derived by mapping ARIS object definitions into MQ Workflow properties. Accordingly, these ones are described by „connection mapping rules".

Source object type	ARIS object definition, where the connection starts from
Target object type	ARIS object definition, where the connection is directed to
Connection kind	Direction of the connection (input / output connection)
Connection type	Name and number of the ARIS connection
Target object rule	Name of the connection mapping rule which let you determine the output connections of the target object, which contribute to a connection chain

Each table row means an alternate ARIS connection (chain) for the corresponding MQ Workflow construct or property.

Rule name		ORGANIZATION_MANAGER		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Organizational unit (43)	Person (46)	input (passive) connection (2)	is Organization Manager for (395)	
Organizational unit (43)	Position (45)	input (passive) connection (2)	is disciplinary superior to (9)	POSITION_IS_OCCUPIED_BY_PERSON
Organizational unit (43)	Position (45)	input (passive) connection (2)	is technical superior to (8)	POSITION_IS_OCCUPIED_BY_PERSON

Table 9: Connection mapping format (find chained „target object rule“ in Table 10)

Rule name		POSITION_IS_OCCUPIED_BY_PERSON		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Position (45)	Person (46)	input (passive) connection (2)	occupies (210)	

Table 10: Connection mapping format (chained rule)

In the following sections, the default mapping rules[30] of the ARIS Bridge are ordered by MQ Workflow constructs, such that the above introduced mapping formats may occur side by side. [31]

Mapping rules for MQ Workflow objects

Rule name	ACTIVITY
Rule type	Object mapping
Objekttyp	Function (22)

Rule name	CONTROL_OBJECT
Rule type	Object mapping
Objekttyp	Event (18)

Rule name	DATA_OBJECT
Rule type	Object mapping
Objekttyp	Entity type (17) Connection type (11) ERM attribute (19) Cluster/Data model (14) Attribute type group (111) Information flow (26)

[30] On request, the mapping rules can be customized for you. Please, contact your IBM representative.

[31] Inside the sections, the mapping rules are listed by alphabetic order of the respective MQ Workflow constructs. MQ Workflow constructs occur in the rule names with their english FDL keywords. But, not all rule names have a corresponding FDL keyword.

Rule name	ORGANIZATION
Rule type	Object mapping
Objekttyp	Organizational unit (37)

Rule name	PERSON
Rule type	Object mapping
Objekttyp	Person (46)

Rule name	PROCESS
Rule type	Object mapping
Objekttyp	eEPC (13) eEPC (material flow) (50) eEPC (column display) (134) eEPC (table display) (154) eEPC (row display) (140) Function allocation diagram (14) Industrial process (103) Office process (100) PLOVC (138) Process chain diagram (PCD) (18) PCD (material flow) (51) UML Activity diagram (124 Value added chain diagram (12)

Rule name	PROGRAM
Rule type	Object mapping
Objekttyp	Application system type (6) IT function type (105) Module type (37)

Rule name	ROLE
Rule type	Object mapping
Objekttyp	Person type (78)

Mapping rules for MQ Workflow properties

Rule name	ACTIVITY_ActivityName
Rule type	Attribute mapping
Attribute type	ObjDefName (-1)[32]

Rule name	ACTIVITY_DESCRIPTION
Rule type	Attribute mapping
Attribute type	Description/definition (9)

Rule name	ACTIVITY_DOCUMENTATION
Rule type	Attribute mapping
Attribute type	Remark/Example (8)

[32] (-1) is no ARIS type number. It means, that the ARIS Bridge will map the ARIS object definition name itself.

Rule name		ACTIVITY_PROGRAM		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Function (22)	Application system type (6)	input (passive) connection (2)	can support (221)	
Function (22)	IT function type (105)	input (passive) connection (2)	supports (147)	
Function (22)	Module type (37)	input (passive) connection (2)	can support (221)	

Rule name		ACTIVITY_PERSON		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Function (22)	Person (46)	input (passive) connection (2)	executes (65)	
Function (22)	Person (46)	input (passive) connection (2)	is technically responsible for (10)	

Rule name		ACTIVITY_MEMBER_OF_ROLE		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Function (22)	Person type (78)	input (passive) connection (2)	executes (218)	
Function (22)	Person type (78)	input (passive) connection (2)	is technically responsible for (220)	

Rule name		ACTIVITY_ORGANIZATION		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Function (22)	Organizational unit (43)	input (passive) connection (2)	executes (65)	
Function (22)	Organizational unit (43)	input (passive) connection (2)	is technically responsible for (10)	

Rule name	ACTIVITY_START_CONDITION
Rule type	Attribute mapping
Attribute type	User attribute Text 1 (985)

Rule name	ACTIVITY_START_MODE
Rule type	Attribute mapping
Attribute type	User attribute Text 3 (1207)

Rule name	ACTIVITY_EXIT_CONDITION
Rule type	Attribute mapping
Attribute type	User attribute Text 2 (986)

Rule name	ACTIVITY_EXIT_MODE
Rule type	Attribute mapping
Attribute type	User attribute Text 4 (1208)

Rule name	ACTIVITY_OTHER_SETTINGS
Rule type	Attribute mapping
Attribute type	User attribute Text 5 (1209)

Rule name	CONTROL_CONNECTOR_DESCRIPTION
Rule type	Attribute mapping
Attribute type	Description/definition (9)

Rule name	CONTROL_CONNECTOR_CONDITION
Rule type	Attribute mapping
Attribute type	User attribute Text 1 (985)

Rule name	DATA_CONNECTOR_DESCRIPTION
Rule type	Attribute mapping
Attribute type	Description/definition (9)

Rule name	DATA_MEMBER_MemberName
Rule type	Attribute mapping
Attribute type	ObjDefName (-1)

Rule name	DATA_MEMBER_DESCRIPTION
Rule type	Attribute mapping
Attribute type	Description/definition (9)

Rule name	DATA_MEMBER_DOCUMENTATION
Rule type	Attribute mapping
Attribute type	Remark/Example (8)

Rule name	DATA_MEMBER_TYPE
Rule type	Attribute mapping
Attribute type	Data type (494)

Rule name	DATA_MEMBER_SIZE_OF_ARRAY
Rule type	Attribute mapping
Attribute type	User attribute Text 1 (985)

Rule name		DATA_MEMBER_HAS_DOMAINTYPE		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
ERM attribute (19)	ERM domain (20)	output (active) connection (4)	has (121)	
ERM attribute (19)	Unit number (185)	output (active) connection (4)	has (121)	UNIT_NUM_HAS_DOMAINTYPE

Rule name		UNIT_NUM_HAS_DOMAINTYPE		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Unit number (185)	ERM domain (20)	output (active) connection (4)	has (121)	

Rule name	DOMAINTYPE_DOM_TYPE
Rule type	Attribute mapping
Attribute type	Domain type (123)

Rule name	DATA_STRUCTURE_ObjectName
Rule type	Attribute mapping
Attribute type	ObjDefName (-1)

Rule name	DATA_STRUCTURE_DESCRIPTION
Rule type	Attribute mapping
Attribute type	Description/definition (9)

Rule name	DATA_STRUCTURE_DOCUMENTATION
Rule type	Attribute mapping
Attribute type	Remark/Example (8)

Rule name		DATA_STRUCTURE_MEMBER		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Entity type (17)	ERM attribute (19)	input (passive) connection (2)	is describing for (78)	
Entity type (17)	ERM attribute (19)	input (passive) connection (2)	is primary key for (80)	
Entity type (17)	ERM attribute (19)	input (passive) connection (2)	is external key for (79)	
Connection type (11)	ERM attribute (19)	input (passive) connection (2)	is describing for (78)	
Connection type (11)	ERM attribute (19)	input (passive) connection (2)	is primary key for (80)	
Connection type (11)	ERM attribute (19)	input (passive) connection (2)	is external key for (79)	
Attribute type group (111)	ERM attribute (19)	input (passive) connection (2)	belongs to (269)	
Entity type (17)	Attribute type group (111)	input (passive) connection (2)	is attribute type group (270)	
Connection type (11)	Attribute type group (111)	input (passive) connection (2)	is attribute type group (270)	
Cluster/Data model (14)	Entity type (17)	output (active) connection (4)	consists of (85)	
Cluster/Data model (14)	ERM attribute (19)	output (active) connection (4)	consists of (85)	
Cluster/Data model (14)	Cluster/Data model (14)	output (active) connection (4)	consists of (85)	

Rule name	EVENT_COMPARE_OPERATOR
Rule type	Attribute mapping
Attribute type	Comparison operator (497)

Rule name	EVENT_COMPARE_VALUE
Rule type	Attribute mapping
Attribute type	Comparison value (498)

Rule name	ORGANIZATION_ObjectName
Rule type	Attribute mapping
Attribute type	ObjDefName (-1)

Rule name	ORGANIZATION_DESCRIPTION
Rule type	Attribute mapping
Attribute type	Description/definition (9)

Rule name		ORGANIZATION_MANAGER		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Organizational unit (43)	Person (46)	input (passive) connection (2)	is Organization Manager for (395)	
Organizational unit (43)	Position (45)	input (passive) connection (2)	is disciplinary superior to (9)	POSITION_IS_OCCUPIED_BY_PERSON
Organizational unit (43)	Position (45)	input (passive) connection (2)	is technical superior to (8)	POSITION_IS_OCCUPIED_BY_PERSON

Rule name		ORGANIZATION_PARENT_ORGANIZATION		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Organizational unit (43)	Organizational unit (43)	input (passive) connection (2)	is disciplinary superior to (9)	
Organizational unit (43)	Organizational unit (43)	input (passive) connection (2)	is technical superior to (8)	
Organizational unit (43)	Organizational unit (43)	input (passive) connection (2)	is responsible for (211)	
Organizational unit (43)	Organizational unit (43)	input (passive) connection (2)	is superior (3)	
Organizational unit (43)	Organizational unit (43)	input (passive) connection (2)	is composed of (7)	

Rule name		ORGANIZATION_RELATED_PERSON		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Organizational unit (43)	Person (46)	input (passive) connection (2)	belongs to (6)	

Rule name	PERSON_PersonName
Rule type	Attribute mapping
Attribute type	E-mail address (509)

Rule name	PERSON_DESCRIPTION
Rule type	Attribute mapping
Attribute type	Description/definition (9)

Rule name	PERSON_FIRST_NAME
Rule type	Attribute mapping
Attribute type	First name (1243)

Rule name	PERSON_LAST_NAME
Rule type	Attribute mapping
Attribute type	ObjDefName (-1)

Rule name	PERSON_PHONE
Rule type	Attribute mapping
Attribute type	Telephone number (245)

Rule name	PERSON_SECOND_PHONE
Rule type	Attribute mapping
Attribute type	Fax number (246)

Rule name		PERSON_SUBSTITUTE		
Rule type		Connection mapping		
Source object type	Target object type	Connection kind	Connection type	Target object rule
Person (46)	Person (46)	input (passive) connection (2)	substitutes for (318)	

Rule name		PERSON_RELATED_ROLE		
Rule type		Connection mapping		
Source object type	Target object type	Connection kind	Connection type	Target object rule
Person (46)	Person type (78)	output (active) connection (4)	is of type (61)	

Rule name		PERSON_RELATED_ORGANIZATION		
Rule type		Connection mapping		
Source object type	Target object type	Connection kind	Connection type	Target object rule
Person (46)	Organizational unit (43)	output (active) connection (4)	belongs to (6)	

Rule name	PERSON_OTHER_SETTINGS
Rule type	Attribute mapping
Attribute type	User attribute Text 1 (985)

Rule name	PROCESS_ProcessName
Rule type	Attribute mapping
Attribute type	ModelName (-1)

Rule name	PROCESS_DESCRIPTION
Rule type	Attribute mapping
Attribute type	Description/definition (9)

Rule name	PROCESS_DOCUMENTATION
Rule type	Attribute mapping
Attribute type	Remark/Example (8)

Rule name	PROCESS_OTHER_SETTINGS
Rule type	Attribute mapping
Attribute type	User attribute Text 1 (985)

Rule name	PROCESS_CATEGORY_ObjectName
Rule type	Attribute mapping
Attribute type	GroupName (-1)

Rule name	PROCESS_CATEGORY_DESCRIPTION
Rule type	Attribute mapping
Attribute type	Description/definition (9)

Rule name	PROCESS_CATEGORY_ DOCUMENTATION
Rule type	Attribute mapping
Attribute type	Remark/Example (8)

Rule name	PROGRAM_ObjectName
Rule type	Attribute mapping
Attribute type	ObjDefName (-1)

Rule name	PROGRAM_DESCRIPTION
Rule type	Attribute mapping
Attribute type	Description/definition (9)

Rule name	PROGRAM_DOCUMENTATION
Rule type	Attribute mapping
Attribute type	Remark/Example (8)

Rule name	PROGRAM_OTHER_SETTINGS
Rule type	Attribute mapping
Attribute type	User attribute Text 1 (985)

Rule name	ROLE_ObjectName
Rule type	Attribute mapping
Attribute type	ObjDefName (-1)

Rule name	ROLE_DESCRIPTION
Rule type	Attribute mapping
Attribute type	Description/definition (9)

Rule name		ROLE_RELATED_PERSON		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Person type (78)	Person (46)	input (passive) connection (2)	is of type (61)	

Rule name		ROLE_COORDINATOR		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Person (46)	Person (46)	input (passive) connection (2)	is Organization Manager for (395)	

Mapping rules for MQ Workflow connectors

Rule name		ATTRIBUTETYPEGROUP_SENDS_DATA_TO_FUNCTION		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Attribute type group (111)	Function (22)	output (active) connection (4)	is input for (49)	

Rule name		CLUSTER_SENDS_DATA_TO_FUNCTION		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Cluster/Data model (14)	Function (22)	output (active) connection (4)	is input for (49)	

Rule name		ENTITY_TYPE_SENDS_DATA_TO_FUNCTION		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Entity type (17)	Function (22)	output (active) connection (4)	is input for (49)	

Rule name		ERM_ATTRIBUTE_SENDS_DATA_TO_FUNCTION		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
ERM attribute (19)	Function (22)	output (active) connection (4)	is input for (49)	

Rule name		EVENT_IS_COMPARED_WITH_DATA		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Event (18)	ERM attribute (19)	input (passive) connection (2)	has state (75)	
Event (18)	ERM attribute (19)	input (passive) connection (2)	provides value for (319)	
Event (18)	ERM attribute (19)	input (passive) connection (2)	is compared to (322)	
Event (18)	Entity type (17)	input (passive) connection (2)	has state (75)	
Event (18)	Attribute type group (111)	input (passive) connection (2)	has state (75)	
Event (18)	Cluster/Data model (14)	input (passive) connection (2)	has state (75)	

Rule name		EVENT_RECEIVES_CONTROL_FROM_RULE		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Event (18)	Rule (50)	input (passive) connection (2)	leads to (117)	

Rule name		EVENT_SENDS_CONTROL_TO_FUNCTION		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Event (18)	Function (22)	output (active) connection (4)	activates (43)	
Event (18)	Rule (50)	output (active) connection (4)	is evaluated by (48)	RULE_SENDS_ CONTROL_TO_ FUNCTION

Rule name		FUNCTION_RECEIVES_CONTROL_FROM_RULE		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Function (22)	Rule (50)	input (passive) connection (2)	activates (43)	
Function (22)	Event (18)	input (passive) connection (2)	activates (43)	EVENT_RECEIVES_ CONTROL_FROM_ RULE

Rule name		FUNCTION_RECEIVES_DATA		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Function (22)	Entity type (17)	input (passive) connection (2)	is input for (49)	
Function (22)	Connection type (11)	input (passive) connection (2)	is input for (49)	
Function (22)	ERM attribute (19)	input (passive) connection (2)	is input for (49)	
Function (22)	Cluster/Data model (14)	input (passive) connection (2)	is input for (49)	
Function (22)	Attribute type group (111)	input (passive) connection (2)	is input for (49)	
Function (22)	Information flow (26)	input (passive) connection (2)	is received from (408)	

Rule name		FUNCTION_SENDS_CONTROL_TO_FUNCTION		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Function (22)	Function (22)	output (active) connection (4)	is predecessor of (118)	
Function (22)	Event (18)	output (active) connection (4)	creates (44)	EVENT_SENDS_ CONTROL_TO_ FUNCTION
Function (22)	Rule (50)	output (active) connection (4)	leads to (116)	RULE_SENDS_ CONTROL_TO_ FUNCTION

Rule name		FUNCTION_SENDS_DATA		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Function (22)	Entity type (17)	output (active) connection (4)	has output of (50)	
Function (22)	Connection type (11)	output (active) connection (4)	has output of (50)	
Function (22)	ERM attribute (19)	output (active) connection (4)	has output of (50)	
Function (22)	Cluster/Data model (14)	output (active) connection (4)	has output of (50)	
Function (22)	Attribute type group (111)	output (active) connection (4)	has output of (50)	
Function (22)	Information flow (26)	output (active) connection (4)	sends (407)	

Rule name		FUNCTION_SENDS_DATA_TO_FUNCTION		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Function (22)	Entity type (17)	output (active) connection (4)	has output of (50)	ENTITY_TYPE_ SENDS_DATA_TO_ FUNCTION
Function (22)	Connection type (11)	output (active) connection (4)	has output of (50)	RELATION_TYPE_ SENDS_DATA_TO_ FUNCTION
Function (22)	ERM attribute (19)	output (active) connection (4)	has output of (50)	ERM_ATTRIBUTE_ SENDS_DATA_TO_ FUNCTION
Function (22)	Cluster/Data model (14)	output (active) connection (4)	has output of (50)	CLUSTER_SENDS_ DATA_TO_ FUNCTION
Function (22)	Attribute type group (111)	output (active) connection (4)	has output of (50)	ATTRIBUTETYPE GROUP_SENDS_ DATA_TO_ FUNCTION
Function (22)	Information flow (26)	output (active) connection (4)	sends (407)	INFO_FLOW_ SENDS_DATA_TO_ FUNCTION

Rule name		INFO_FLOW_SENDS_DATA_TO_FUNCTION		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Information flow (26)	Function (22)	output (active) connection (4)	is received from (408)	

Rule name		POSITION_IS_OCCUPIED_BY_PERSON		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Position (45)	Person (46)	input (passive) connection (2)	occupies (210)	

Rule name		PROGRAM_RECEIVES_DATA		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Application system type (6)	Entity type (17)	input (passive) connection (2)	is managed by (317)	
Application system type (6)	Connection type (11)	input (passive) connection (2)	is managed by (317)	
Application system type (6)	Entity type (17)	output (active) connection (4)	can use (125)	
Application system type (6)	Connection type (11)	output (active) connection (4)	can use (125)	
Application system type (6)	ERM attribute (19)	output (active) connection (4)	can use (125)	
Application system type (6)	Cluster/Data model (14)	output (active) connection (4)	can use (125)	
Module type (37)	Entity type (17)	output (active) connection (4)	can use (125)	
Module type (37)	Connection type (11)	output (active) connection (4)	can use (125)	
Module type (37)	ERM attribute (19)	output (active) connection (4)	can use (125)	
Module type (37)	Cluster/Data model (14)	output (active) connection (4)	can use (125)	
IT function type (105)	Entity type (17)	output (active) connection (4)	uses (60)	
IT function type (105)	Connection type (11)	output (active) connection (4)	uses (60)	
IT function type (105)	ERM attribute (19)	output (active) connection (4)	uses (60)	
IT function type (105)	Cluster/Data model (14)	output (active) connection (4)	uses (60)	

Rule name		PROGRAM_SENDS_DATA		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Application system type (6)	Entity type (17)	input (passive) connection (2)	is managed by (317)	
Application system type (6)	Connection type (11)	input (passive) connection (2)	is managed by (317)	
Application system type (6)	Entity type (17)	output (active) connection (4)	can use (125)	
Application system type (6)	Connection type (11)	output (active) connection (4)	can use (125)	
Application system type (6)	ERM attribute (19)	output (active) connection (4)	can use (125)	
Application system type (6)	Cluster/Data model (14)	output (active) connection (4)	can use (125)	
Module type (37)	Entity type (17)	output (active) connection (4)	can use (125)	
Module type (37)	Connection type (11)	output (active) connection (4)	can use (125)	
Module type (37)	ERM attribute (19)	output (active) connection (4)	can use (125)	
Module type (37)	Cluster/Data model (14)	output (active) connection (4)	can use (125)	
IT function type (105)	Entity type (17)	output (active) connection (4)	uses (60)	
IT function type (105)	Connection type (11)	output (active) connection (4)	uses (60)	
IT function type (105)	ERM attribute (19)	output (active) connection (4)	uses (60)	
IT function type (105)	Cluster/Data model (14)	output (active) connection (4)	uses (60)	

Rule name		RELATION_TYPE_SENDS_DATA_TO_FUNCTION		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Connection type (11)	Function (22)	output (active) connection (4)	is input for (49)	

Rule name		RULE_SENDS_CONTROL_TO_FUNCTION		
Rule type		Connection mapping		
Source object type	**Target object type**	**Connection kind**	**Connection type**	**Target object rule**
Rule (50)	Function (22)	output (active) connection (4)	activates (43)	
Rule (50)	Event (18)	output (active) connection (4)	leads to (117)	EVENT_SENDS_ CONTROL_TO_ FUNCTION
Rule (50)	Rule (50)	output (active) connection (4)	links (54)	RULE_SENDS_ CONTROL_TO_ FUNCTION