ARIS 10

MANUEL MÉTHODE

OCTOBRE 2025

This document applies to ARIS Version 10 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2010-2025 SAG ARIS GmbH, Saarbrücken, Germany and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all SAG ARIS GmbH product names are either trademarks or registered trademarks of Software GmbH and/or SAG ARIS GmbH and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software GmbH Products / Copyright and Trademark Notices of Software GmbH Products". These documents are part of the product documentation, located at https://softwareag.com/licenses and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with SAG ARIS GmbH.

Contenu

1 Introduction 1

2 Architecture de systèmes d'informations intégrés (ARIS) 3
2.1 Concept de l'architecture ARIS 3
2.2 Vues descriptives 3
2.3 Niveaux descriptifs 7

3 Modélisation au sein des vues et des niveaux du concept ARIS 12
3.1 Vue des fonctions 12
3.1.1 Règles de gestion 12
3.1.1.1 Arbre de fonctions 13
3.1.1.2 Diagramme des objectifs 19
3.1.2 Concept informatique - diagramme de type d'application 21
3.1.3 Implémentation - Diagramme d'applications 25
3.2 Vue des données 28
3.2.1 Règles de gestion 28
3.2.1.1 Modèle de base MER 28
3.2.1.2 Extensions du MER : MERe 33
3.2.1.3 Ajout d'opérateurs de construction 33
3.2.1.4 Extensions des cardinalités 38
3.2.1.5 Dépendance d'identification et d'existence 40
3.2.1.6 Modélisation de termes spécifiques dans un modèle de termes spécifiques de l'entreprise 41
3.2.1.7 Diagr. de ratt. d'attributs du modèle MERe 43
3.2.1.8 Modèle de données IE 44
3.2.1.9 Résumé des termes et des formes de représentations les plus importants du MERe 45
3.2.1.10 Modélisation de la structure Data Warehouse 46
3.2.1.11 Données de gestion de projets - Diagramme de support d'informations 47
3.2.2 Implémentation - Diagramme de table 48
3.2.3 Diagramme d'affectation de rôles (RAD - Role allocation diagram) 50
3.3 Vue organisationnelle 51
3.3.1 Règles de gestion 51
3.3.1.1 Conceptualisation organisationnelle des entreprises 51
3.3.1.2 Organigramme 53
3.3.2 Concept informatique - Topologie de réseau 59
3.3.3 Implémentation 61
3.3.3.1 Diagramme de réseau 61
3.3.3.2 Modélisation du flux de matières - Ressources techniques 63
3.4 Vue des processus 66
3.4.1 Règles de gestion 66
3.4.1.1 Relier des fonctions avec organisation - CPE 66
3.4.1.2 Commande événementielle - Chaîne de processus événementielle (CPE) 67
3.4.1.3 Diagramme de rattachement de fonctions (E/S) 76
3.4.1.4 Diagramme d'événements 78
3.4.1.5 Diagramme de chaînes de plus-value 79

3.4.1.6 Modélisation orientée objet 79
3.4.1.7 Matrice de choix des processus 79
3.4.1.8 Modélisation du flux de matières 81
3.4.1.9 CPE (flux de matières) 81
3.4.1.10 CPE (sous forme de colonnes/lignes) 82
3.4.1.11 Diagramme de contrôles d'affaires 83
3.4.1.12 eBusiness scenario diagram 85
3.4.1.13 Modèle de composition 88
3.4.1.14 Diagramme de rôles 89
3.4.1.15 Modèle express 90
3.4.1.16 Dessin de masque 92
3.4.1.17 Navigation de masque 94
3.4.1.18 Matrice des champs d'activités 95
3.4.2 Concept informatique 97
3.4.2.1 Diagramme d'accès 97
3.4.2.2 Fonctions avec données 97
3.4.2.3 Organisation avec fonctions 98
3.4.2.4 Diagramme de déroulement de programme 99
3.4.2.5 Plan de déroulement de programme (PD) 100
3.4.2.6 Diagramme de masque 101
3.4.3 Implémentation - Diagramme d'accès (physique) 104
3.4.3.1 Fonctions avec données 104
3.4.3.2 Organisation avec données 106
3.4.3.3 Organisation avec fonctions 107
3.5 Modélisation de prestation/produit 109
3.5.1 Diagramme d'échange de prestations/produits 110
3.5.2 Arbre de prestations/produits 111
3.5.3 Diagramme de rattachement des produits 112
3.5.4 Arbre de produits 115
3.5.5 Matrice de choix des produits 116

4 Unified Modeling Language (UML) dans ARIS 117
4.1 Introduction 117
4.2 ARIS UML Designer - Standard UML pris en charge 117

5 Méthodes de gestion des connaissances 119
5.1 Introduction 119
5.2 Types d'objets pour la modélisation du traitement des connaissances 120
5.2.1 Catégorie des connaissances 120
5.2.2 Connaissances documentées 123
5.3 Types de modèles pour la modélisation du traitement des connaissances 123
5.3.1 Diagramme de structure des connaissances 124
5.3.2 Carte des connaissances 124
5.3.3 Représentation du traitement des connaissances dans les processus d'entreprise 126

6 Méthode du tableau de bord prospectif 128
6.1 Introduction 128
6.2 Concept du tableau de bord prospectif 128
6.2.1 Bases de l'approche TBP 128
6.2.2 Processus de gestion stratégique et tableau de bord prospectif 129

6.2.2.1 Formuler et traduire la vision et la stratégie 130
6.2.2.2 Perspectives standard d'un tableau de bord prospectif 131
6.2.2.3 Chaîne de cause à effet 132
6.2.2.4 Définition des indicateurs précoces et retardés 133
6.2.2.5 Communication et déduction d'autres tableaux de bord prospectifs 134
6.2.2.6 Planification et directives 134
6.2.2.7 Apprentissage stratégique et feed-back 136
6.2.3 Avantages et utilité du tableau de bord prospectif 136
6.3 Développement d'un tableau de bord prospectif avec ARIS BSC 137
6.3.1 Concepts et abréviations 137
6.3.2 Élaboration des tableaux de bord prospectifs avec ARIS BSC 139
6.3.2.1 Définir les perspectives 139
6.3.2.2 Définir la structure du système de tableau de bord prospectif 139
6.3.2.3 Définir les relations de cause à effet 140
6.3.2.4 Définir les initiatives et les indicateurs pour le contrôle de l'objectif 143
6.3.2.5 Décrire les indicateurs et leurs rapports 146
6.3.3 Relations avec d'autres modèles 147

7 eBusiness scenario diagram 148
7.1 Introduction 148
7.2 La méthode du e-Business scenario diagram 150
7.2.1 L'idée 150
7.2.2 Le modèle et ses objets 150
7.2.3 Groupe d'attributs Mode de transmission 152
7.3 Evaluations par les rapports 152
7.3.1 Vérification de la sécurité des données 152
7.3.2 Support système 152
7.3.3 Flux d'informations 153
7.3.4 Collaborative business maps 153

7.4 Connexion à d'autres méthodes et composants 153

8 IT City Planning 158
8.1 Architecture d'entreprise et IT City Planning 158
8.2 A quelles entreprises s'adresse IT City Planning ? 158
8.3 IT City Planning avec ARIS 159
8.4 Vue des services 161
8.5 Les types de services et leurs données 165
8.6 Description détaillée des types de services 166
8.7 Déroulements chronologiques entre les éléments du SI 166
8.8 Vue technique 167
8.9 Les éléments informatiques et leurs données 168
8.10 Description détaillée des éléments informatiques 168
8.11 Aspects organisationnels 170
8.12 Déroulements chronologiques entre les éléments informatiques 170
8.13 Déroulements chronologiques au sein de l'architecture 171
8.14 Evaluations possibles 172

9 Business process modeling 173
9.1 Les classes de processus et le business process diagram 173
9.2 Réalisation de BPMN dans ARIS 175
9.3 Les éléments du business process diagram 176
9.3.1 Pistes et corridors 176
9.3.2 Directives de modélisation pour les pistes et les corridors 178
9.3.3 Flux de séquence 178
9.3.4 Directives de modélisation pour les liaisons de flux de séquence 178
9.3.5 Flux de message 179
9.3.6 Directives de modélisation pour les liaisons de flux de message 179
9.3.7 Association 180
9.3.8 Événements 181
9.3.9 Directives de modélisation pour les événements 182
9.3.10 Activités 183
9.3.11 Directives de modélisation pour les activités 184
9.3.12 Branchement 185
9.3.13 Directives de modélisation pour les branchements 186
9.3.14 Artifact 188
9.3.15 Sources des figures 190

10 Modeling BPMN 2.0 191
10.1 Introduction 191
10.1.1 Initial situation and objective 191
10.1.2 Purpose of this chapter 191
10.2 BPMN core elements and their implementation in ARIS 192
10.2.1 Infrastructure 192
10.2.2 Foundation 193
10.2.3 Common Elements 196
10.2.3.1 Artifacts 196
10.2.3.2 Association 200
10.2.3.3 Group 200
10.2.3.4 Text annotation 201
10.2.3.5 Callable Elements 202
10.2.3.6 Event 202

10.2.3.7 Expression 203
10.2.3.8 Flow Element 203
10.2.3.9 Flow Elements Container 204
10.2.3.10 Gateways 205
10.2.3.11 Message 206
10.2.3.12 Message flow 207
10.2.3.13 Participant 209
10.2.3.14 Resource 213
10.2.3.15 Sequence flow 213
10.2.3.16 Elements not included in the current implementation 217
10.3 BPMN diagrams and ARIS model types: An overview 218
10.4 Process 219
10.4.1 Activities 221
10.4.1.1 Resource assignment 224
10.4.1.2 Performer 224
10.4.1.3 Activity type: Task 224
10.4.1.4 Human interactions 229
10.4.1.5 Activity type: Subprocess 229
10.4.1.6 Subprocess type: Subprocess 230
10.4.1.7 Type de sous-processus : sous-processus d'événement 232
10.4.1.8 Subprocess type: Transaction 234
10.4.1.9 Subprocess type: Ad hoc subprocess 235
10.4.1.10 Subprocess type: Call Activity 237
10.4.1.11 Global task 239
10.4.1.12 Loop characteristics 239
10.4.1.13 Loop characteristics representations 239
10.4.1.14 Standard and multi-instance loop characteristics and complex behavior definition 240
10.4.2 Items and Data 244
10.4.2.1 Data object 244
10.4.2.2 Data store 246
10.4.3 Events 248
10.4.3.1 Catch events and throw events 250
10.4.3.2 Start event 252
10.4.3.3 Intermediate events 253
10.4.3.4 End event 254
10.4.3.5 Event definitions 255
10.4.4 Gateways 263
10.4.4.1 Exclusive gateway 265
10.4.4.2 Inclusive gateway 265
10.4.4.3 Parallel gateway 267
10.4.4.4 Complex gateway 267
10.4.4.5 Event-based gateways 268
10.4.5 Lanes 269
10.5 Collaboration 271
10.5.1 Pool and participant 272
10.5.2 Object types and connection types reused from a process 273
10.5.3 Message flow 273
10.6 Conversation 274
10.6.1 Conversation container 274

10.6.2 Conversation nodes 275
10.6.3 Participant 276
10.6.4 Artifacts 277
10.6.5 Conversation link 277
10.6.6 Message flow in a conversation 278
10.6.7 Model assignments 278
10.7 Enterprise BPMN collaboration diagram 279

11 Gestion de l'expérience client (CXM) 280
11.1 Paysage du parcours client 280
11.2 Cartographie du parcours client 282
11.3 Diagramme d'affectation des points de contact du client 285
11.4 Cartographie des points de contact du client 286
11.5 Connexion entre CXM et BPM 287
11.5.1 Possibilités d'analyse 288
11.5.1.1 Rapport 288
11.5.1.2 Requêtes 290
11.5.1.3 Get full customer journey overview 290
11.5.1.4 Find customer touchpoints clustered by associated risk 292
11.5.1.5 Find customer touchpoints clustered by associated ownership 292
11.5.1.6 Find customer touchpoints clustered by associated channel 293
11.5.1.7 Find risks and initiatives for all customer touchpoints 294
11.5.1.8 Find risks and initiatives for bad customer touchpoints only 295
11.5.1.9 Find all processes related to customer journeys 296

12 Cas d'application 297
12.1 Documentation générale de l'entreprise 300
12.2 Gestion de base de données/Data Warehousing 302
12.3 Gestion de réseau et de matériel informatique 303
12.4 Calcul de coûts de processus 304
12.5 Gestion de la qualité 305
12.6 Mesures de réorganisation 306
12.7 Introduction à SAP-R/3 308
12.8 Développement et introduction de logiciels 309
12.9 Gestion des connaissances 311
12.10 Gestion du flux de travail 313

13 Bibliographie 314
13.1 Bibliographie générale 314
13.2 Bibliographie relative aux sujets 316
13.2.1 Unified Modeling Language dans ARIS 316
13.2.1.1 Spécification UML 316
13.2.1.2 Utilisation d'UML 316
13.2.1.3 UML et la modélisation de processus d'entreprise 316
13.2.2 Méthodes de gestion des connaissances 317
13.2.2.1 Gestion des connaissances en général 317
13.2.2.2 Utilisation d'ARIS pour la gestion des connaissances 317
13.2.3 Méthode du tableau de bord prospectif 317
13.2.4 IT City Planning 317
13.2.5 Business process modeling 318

14 Informations légales et d'assistance 319
14.1 Étendue de la documentation 319
14.2 Support 319

1 Introduction

Par le passé, la conception des systèmes et leur intégration recouvraient le plus fort potentiel d'optimisation. Les activités se sont pourtant recentrées sur la création de solutions répondant aux demandes spéciales de certains secteurs durant ces dernières années. La disponibilité accrue des systèmes d'information décentralisés et la possibilité de les incorporer dans des infrastructures intégrées de systèmes d'information ont engendré de nouveaux potentiels de baisse des coûts dans le domaine de la conception organisationnelle des entreprises.

Les structures organisationnelles étaient auparavant subdivisées en fonctions et établies de façon centrale, parce que la plupart étaient basées sur des environnements hôtes centralisés dotés de capacités limitées. Les entreprises manquaient donc de flexibilité. Au début, seules quelques personnes ont pris conscience ou accordé de l'attention aux nouvelles perspectives découlant de l'essor de la décentralisation des services informatiques et du développement parallèle de nouveaux concepts architecturaux pour les systèmes d'information (par exemple, les modèles client/serveur ou la gestion des workflows).

Aujourd'hui, la concurrence toujours plus vive a transformé ce potentiel pour en faire le principal sujet de préoccupation dans toutes les entreprises. La flexibilité des structures axées en permanence sur les processus commerciaux internes est devenue le facteur concurrentiel décisif pour les entreprises. Toutefois, seule une vision globale de l'ensemble des processus commerciaux permet à une entreprise d'identifier, rationaliser et prendre en charge des processus interconnectés au sein d'infrastructures optimisées de systèmes d'information. La gestion de ces nouvelles structures est nettement plus complexe que la gestion d'environnements d'entreprise centralisés. Relever un tel défi exige une affectation sans équivoque des responsabilités, une transparence maximale des structures, une communication homogène à tous les niveaux de l'entreprise et une gestion rationalisée des projets sur la base des objectifs commerciaux définis.

Les méthodes de modélisation d'entreprise aident les chefs d'entreprise à accomplir leurs tâches complexes. Quant aux modèles d'entreprise, ils constituent une condition préalable indispensable pour analyser les processus commerciaux, harmoniser les projets avec les objectifs commerciaux globaux et utiliser les infrastructures des systèmes d'information sous forme de systèmes composites répartis et intégrés afin de soutenir de manière optimale ces structures organisationnelles allégées.

De ce fait, la modélisation de la situation réelle d'une entreprise (et par la même occasion, l'examen de ces processus commerciaux globaux) tend à devenir le premier sujet de préoccupation. La diversité et l'augmentation du nombre déjà conséquent des méthodes de modélisation utilisées causent plus de complexité et de confusion. Par conséquent, des efforts ont été déployés pour définir des concepts-cadres standardisés (architectures) au sujet des méthodes de modélisation et de développement.

L'un de ces concepts est l'**Architecture of Integrated Information Systems** (**ARIS**©) développée par Scheer (cf. Scheer, "Architecture of Integrated Information Systems", 1992). Ce concept architectural permet d'évaluer et d'organiser les méthodes en se concentrant sur leurs points principaux. Il sert aussi de cadre d'orientation pour les projets de développement complexes, car il contient, du fait de ses propres éléments structurels, un modèle de procédure implicite dédié au développement de systèmes d'information intégrés.

Une architecture de ce type entraîne naturellement plus de standardisation dans l'utilisation des méthodes. En se basant sur cette architecture, les méthodes de modélisation récentes et plus anciennes ont été combinées pour créer une méthode holistique de modélisation des processus commerciaux.

De plus, l'architecture ARIS intègre des produits tels qu'ARIS Architect au sein de la gamme de ARIS. Ces produits aident les consultants et les entreprises à créer, analyser et évaluer des processus commerciaux sur le plan de leur réingénierie. ARIS Designer permet d'enregistrer et de modéliser aisément les processus commerciaux concernés dans les départements opérationnels.

Ce manuel fournit une première introduction au sujet des méthodes de modélisation intéressantes. Il présente par ailleurs les approches et les méthodes qui utilisent toute la gamme des produits ARIS, y compris les compléments système. Ce manuel constitue également un excellent support pour les utilisateurs qui s'occupent des méthodes de modélisation, mais qui n'ont pas l'intention d'étudier les questions ou les problèmes liés à l'emploi de ces outils.

2 Architecture de systèmes d'informations intégrés (ARIS)

2.1 Concept de l'architecture ARIS

La conception de l'**AR**chitecture de **S**ystèmes d'**I**nformation Intégrés (ARIS) repose sur un concept d'intégration dicté par une vision globale des processus de l'entreprise. La conception de l'architecture se base tout d'abord sur un modèle développé pour les processus d'entreprise et contenant toutes les caractéristiques principales nécessaires à la description de processus d'entreprise. Le modèle complexe qui en résulte est décomposé en plusieurs vues. Cette décomposition par vues permet alors de procéder à la description du contenu de ces vues à l'aide de méthodes spécialement adaptées à chacune d'elles sans qu'il soit nécessaire de tenir compte des multiples relations et liens que ces vues peuvent entretenir entre elles. Les relations entre les vues sont ensuite prises en compte et regroupées sans redondances pour une vue générale des chaînes de processus.

La deuxième étape permettant de réduire la complexité de l'architecture consiste à distinguer différentes descriptions. Les diverses méthodes descriptives appliquées aux systèmes d'information sont classées selon un concept Life Cycle en fonction de leur degré de rapprochement avec les techniques de traitement de l'information. Ceci permet d'obtenir une description approfondie de tous les aspects, depuis la problématique de gestion d'entreprise jusqu'à la transposition technique.

Le concept ARIS fournit ainsi un cadre dans lequel des systèmes d'information intégrés peuvent être développés et optimisés et dans lequel la transposition de ces systèmes peut être décrite. C'est en particulier l'importance du niveau descriptif spécialisé qui permet au concept ARIS de jouer un rôle d'orientation lors de l'élaboration, de l'analyse et de l'évaluation de chaînes de processus économiques. Une description plus précise de l'architecture de systèmes d'information intégrés est fournie par Scheer (voir Scheer, Architecture of Integrated Information Systems, 1992, ainsi que Scheer, ARIS - Business Process Frameworks, 1998).

2.2 Vues descriptives

Le point de départ de l'observation est un processus d'entreprise tel que représenté à la figure suivante.

Le processus est déclenché par l'événement **La commande client est arrivée**. Cet événement déclenche la fonction (opération) **Accepter la commande client**. L'exécution de cette opération requiert les descriptions de l'état de l'environnement pertinent du processus. Les données concernant le client et les articles sont ici particulièrement pertinentes. Le traitement de l'opération peut modifier l'état des objets environnementaux ; ainsi, les données

sur le stock magasin des articles peuvent être mises à jour à l'aide des nouvelles données de réservation.

Les processus sont exécutés par des employés qui peuvent être affectés à des services. Le service utilise des ressources déterminées des techniques de traitement de l'information (ordinateur personnel, imprimante, etc.) pour exécuter ses tâches.

Le processus **Accepter la commande client** est suivi de l'événement **Commande confirmée** qui, à son tour, entraîne d'autres opérations telles que **Créer le plan de production** et **Suivre la commande**. L'objet **Commande** a maintenant un autre état.En effet, l'objet **Commande entrée** est devenu une **Commande confirmée**. L'exécution de la fonction **Accepter la commande client** a pour résultat la création d'une prestation qui, conjointement aux ressources personnelles et matérielles, sera utilisée comme entrée lors du traitement des processus suivants.

Illustration 1: Modèle de processus d'entreprise

Les composants requis pour la description complète d'un processus d'entreprise sont les opérations, les événements, les prestations (états), les utilisateurs, les unités organisationnelles et les ressources des techniques de traitement de l'information. Si tous les effets sur tous les éléments du processus devaient être pris en compte pour chaque opération observée, le modèle serait extrêmement complexe et la description serait redondante.

Afin de réduire la complexité, la vue d'ensemble est décomposée en différentes vues (cf. figure suivante) représentant des champs de modélisation et de conception (cf. Scheer, Architektur integrierter Informationssysteme 1992, pp. 13 et suiv.). Vous pouvez les éditer indépendamment l'un de l'autre. La décomposition des vues est conçue de sorte que les composants d'une vue ont une relation très étroite entre eux mais que l'interconnexion est plus vague entre les vues.

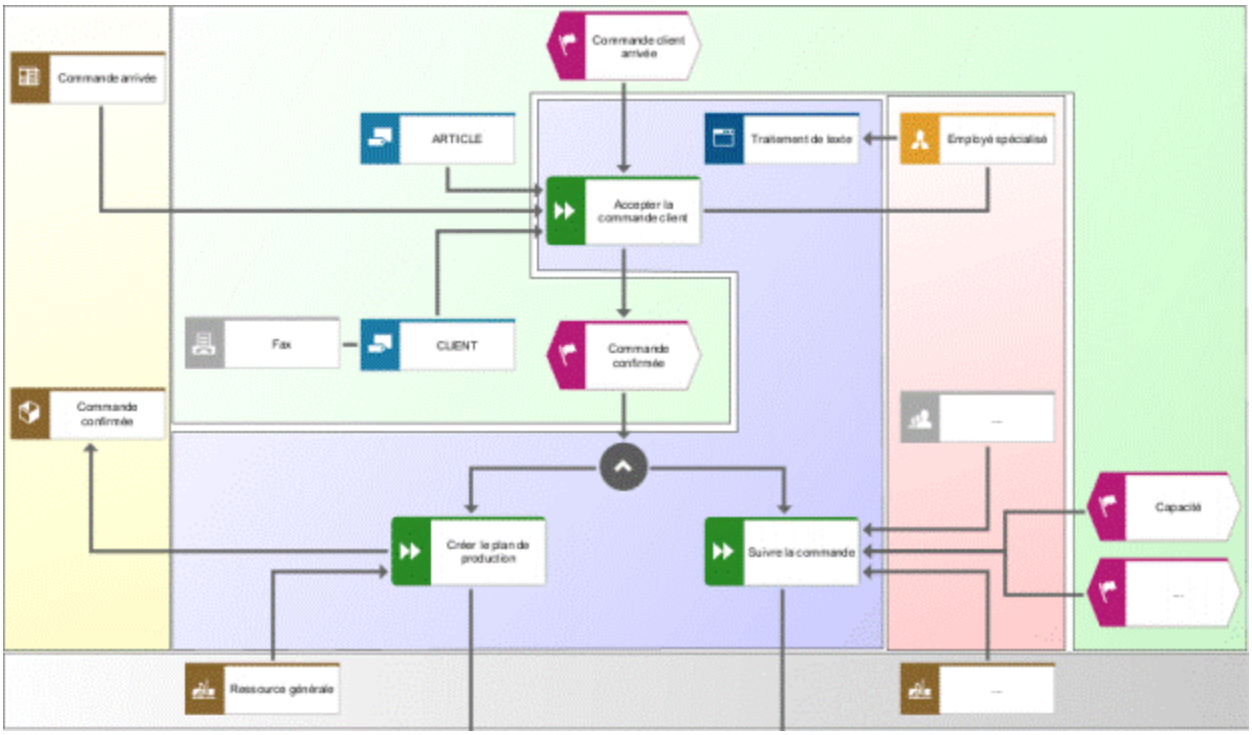


Illustration 2: Vue d'un modèle de processus

Les événements tels que **La commande client est arrivée**, **La facture est établie** définissent l'apparition d'une modification de l'état des objets d'informations (données). Ils sont décrits dans la vue des données de l'architecture ARIS.

Les états dans l'environnement des objets, par exemple dans l'environnement de la commande, sont représentés par le biais de prestations/produits. Une prestation/produit peut être soit une prestation en nature, soit une prestation de service. Les prestations de services permettant la création et la mise à disposition d'informations sont appelées prestations de services d'informations. La mise à disposition de moyens financiers fait également partie des prestations/produits. Les rapports entre les prestations/produits sont décrits dans la **vue de prestations/produits** de l'architecture ARIS.

Les fonctions à exécuter (opérations) ainsi que les rapports existant entre elles représentent la seconde vue, la **vue des fonctions**. Elle regroupe la description des fonctions, la liste des différentes sous-fonctions appartenant à l'ensemble, ainsi que les relations hiérarchiques entre les fonctions.

La **vue organisationnelle** regroupe les utilisateurs et les unités organisationnelles ainsi que leurs relations et leurs structures.

Les ressources des techniques de traitement de l'information représentent un autre champ d'observation, la **vue des ressources**. Celle-ci n'est cependant pertinente pour l'observation d'un processus d'entreprise que comme condition cadre pour la description des autres composants plutôt orientés économique. Les descriptions des composants des autres vues (données, fonctions, organisation) sont donc décrites en fonction de leur proximité des ressources pour les techniques de traitement de l'information. Les ressources sont donc traitées aux niveaux descriptifs du concept informatique et de l'implémentation (cf. chapitre **Niveaux descriptifs** (page 7)). Le modèle Life-Cycle défini par l'observation des niveaux remplace ainsi la vue des ressources et devient un champ descriptif à part entière.

La décomposition des processus en plusieurs vues est certes l'objectif de la simplification mais elle entraîne également la disparition des relations entre les éléments de processus entre les vues. C'est pourquoi une vue supplémentaire, la **vue de gestion**, est ajoutée ; elle décrit les relations entre les vues. La prise en compte de ces relations dans une vue propre permet de saisir toutes les relations systématiquement et sans redondances.

La vue de gestion est un composant fondamental d'ARIS. Elle représente également une des principales différences entre le concept ARIS et les autres concepts d'architecture (pour la comparaison avec d'autres systèmes d'architecture, cf. Scheer, Architecture of Integrated Information Systems, p. 24 et sqq.).

ARIS dispose ainsi de cinq vues que nous approcherons également dans les autres descriptions de méthodes.

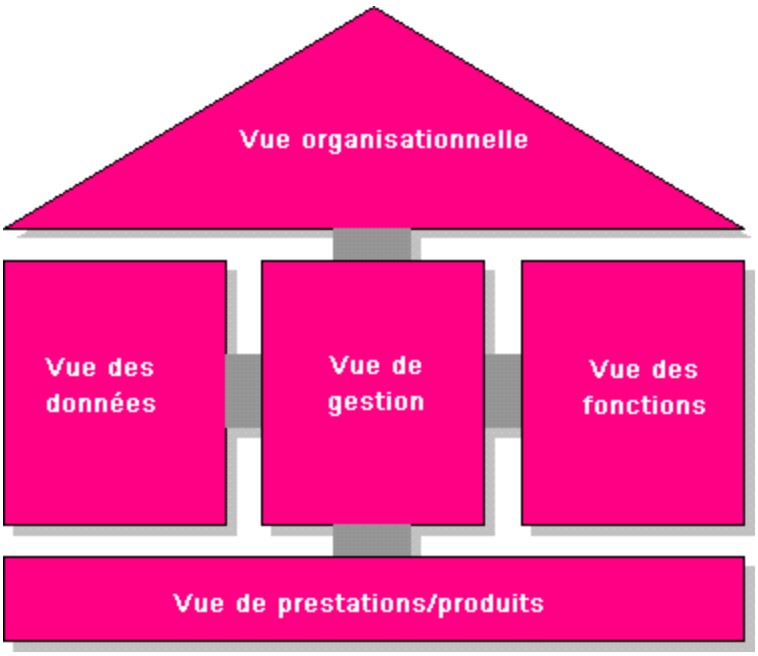


Illustration 3: Vues de décomposition du modèle de processus

2.3 Niveaux descriptifs

Comme nous l'avons déjà expliqué, la vue des ressources d'ARIS est représentée par un concept Life Cycle des descriptions d'un système d'information.

Les modèles Life-Cycle sous forme de concepts par étapes ou par phases décrivent l'historique d'un système d'information. Le modèle Life Cycle d'ARIS n'a cependant pas la signification d'un modèle de procédé pour le développement d'un système d'information. Il définit plutôt différentes descriptions qui se distinguent par leur proximité aux techniques de traitement de l'information.

ARIS fait une distinction dans la subdivision en trois parties représentée dans la figure suivante (cf. Scheer, Architektur integrierter Informationssysteme 1992, pp. 16 et suiv.).

Le point de départ de l'observation est l'**énoncé économique du problème**. La description regroupe des états généraux très proches des objectifs spécifiques et fortement orientés langage spécialisé. Les possibilités offertes par la technique de traitement de l'information pour gérer les processus et les décisions économiques sont également prises en compte. C'est pourquoi seules les méthodes descriptives semi-formelles sont prises en compte pour la représentation. Elles ne peuvent pas servir de point de départ à une transposition formalisée lors de l'implémentation à cause de leur manque de détail et du vocabulaire spécifique au domaine spécialisé.

Dans les **règles de gestion**, le concept d'application de gestion d'entreprise devant être géré est donc représenté dans une langue descriptive suffisamment formalisée afin qu'il puisse servir de base à une transposition cohérente dans les techniques de traitement de l'information. Cette opération est également désignée comme modélisation (sémantique). La règle de gestion est liée très étroitement à l'énoncé économique du problème, comme l'indique la largeur de la flèche de la figure suivante.

Si le champ conceptuel des règles de gestion est transféré dans les catégories de la transposition informatique, le niveau du **concept informatique** est atteint. Ce sont les méthodes ou les transactions exécutant les fonctions spécialisées qui sont définies ici et non ces dernières. Ce niveau peut être défini comme le lieu d'adaptation de la description spécifique aux canevas descriptifs généraux des techniques de traitement de l'information. La règle de gestion et le concept informatique ne sont liés que de loin. Vous pouvez modifier un concept informatique sans aucun effet sur la règle de gestion. Cela ne signifie cependant pas que la règle de gestion et le concept informatique peuvent être développés indépendamment l'un de l'autre. Au contraire, le contenu économique doit être fixé à la fin de la description au niveau de la règle de gestion de manière à ce que les arguments purement informatiques, tels que le comportement de la prestation d'un système informatique, n'aient aucune influence sur les contenus spécifiques.

Au niveau de l'**implémentation**, le concept informatique est converti en composants informatiques matériels et logiciels concrets. La relation à la technique du traitement de l'information est ainsi établie.

Les descriptions sont caractérisées par différents cycles de modification. La fréquence de modification la plus faible est au niveau des règles de gestion et la plus forte au niveau de l'implémentation.

Le niveau d'implémentation est étroitement lié au développement de la technique de traitement de l'information et subit constamment des modifications de par ses cycles innovateurs fréquents.

Le niveau de la règle de gestion revêt une grande importance dans la mesure où il représente à la fois le concept d'application économique à long terme et le point de départ des étapes de générations ultérieures pour le passage à l'implémentation technique. Les règles de gestion ont la durée de vie la plus longue et documentent l'utilité spécifique du système d'information de par leur grande proximité avec l'énoncé économique du problème. C'est la raison pour laquelle la vue est au premier plan du développement des règles de gestion ou des modèles sémantiques. Les modèles sémantiques constituent la liaison entre l'utilisateur et la première transposition de son énoncé en un langage semi-informatique.

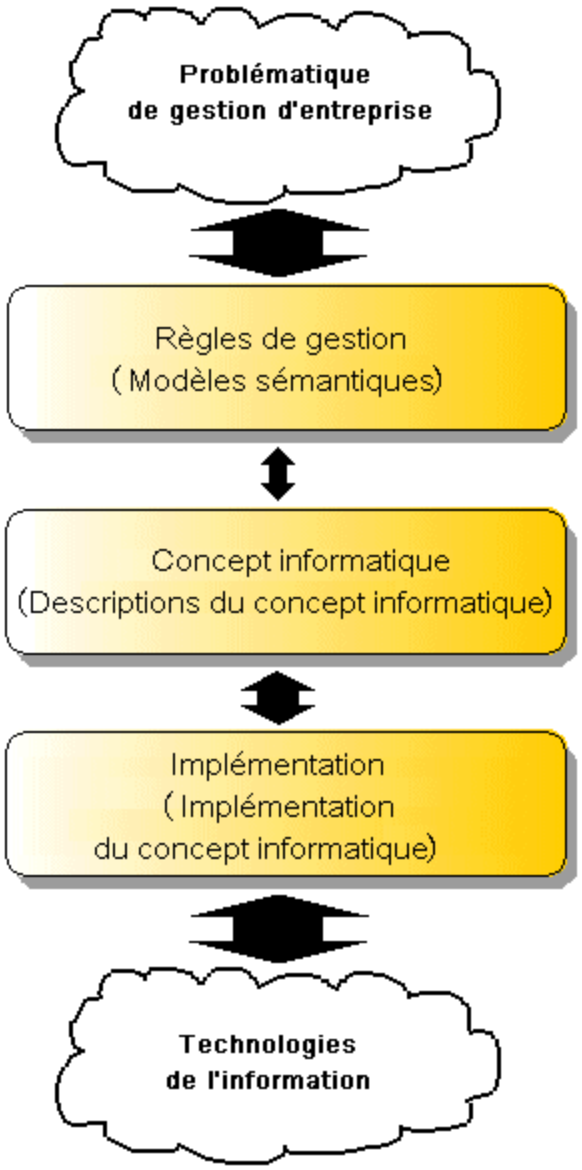


Illustration 4: Descriptions d'un système d'information

Avec la constitution des vues et des descriptions (solution de départ économique comprise), nous avons développé le concept ARIS. Chacune des vues descriptives est décrite pour les trois niveaux (cf. figure suivante) : **règles de gestion**, **concept informatique** et **implémentation**.

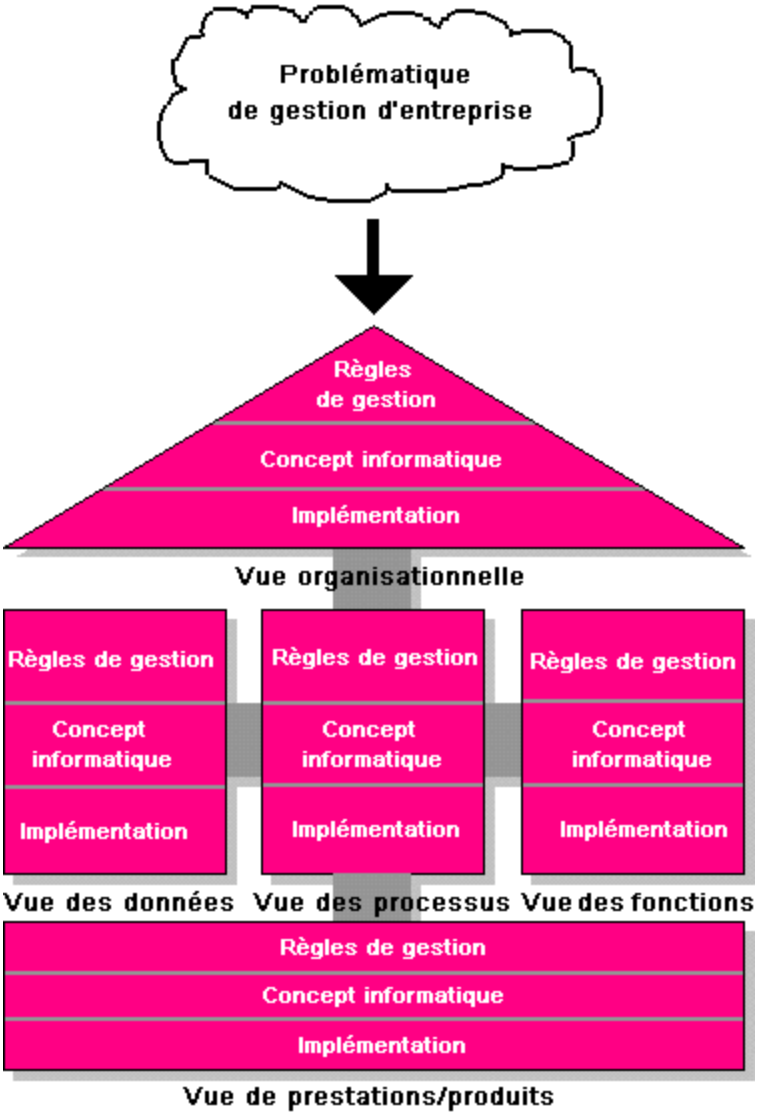


Illustration 5: Concept ARIS

Le concept ARIS que nous avons développé permet d'établir des champs descriptifs définis par les vues et les niveaux descriptifs de l'architecture. Il en résulte treize composants, dont l'énoncé économique du problème servant de départ à l'observation. Il s'agit désormais de sélectionner et d'expliquer les méthodes descriptives adéquates pour chaque champ d'observation.

Les critères de sélection des méthodes sont les suivants (cf. Scheer, Wirtschaftsinformatik 1994, p. 18) :

- la simplicité et la clarté des moyens de représentation,

- la qualification pour les contenus spécifiques à exprimer,
- la possibilité de mettre en œuvre des méthodes uniformes pour toutes les applications à représenter,
- le degré de connaissance existant ou supposé des méthodes, ainsi que
- l'indépendance des méthodes de développement technique des techniques de traitement de l'information.

Toutes les méthodes des champs descriptifs seront décrites par la suite.

3 Modélisation au sein des vues et des niveaux du concept ARIS

3.1 Vue des fonctions

3.1.1 Règles de gestion

Souvent, les méthodes de modélisation traitent les fonctions dans le contexte des objets des autres vues descriptives d'ARIS. Elles représentent, par exemple, le rapport entre données et fonctions pour spécifier le processus de transformation d'une fonction à l'aide des données d'entrée/de sortie de la fonction.

Cependant, en ce qui concerne l'architecture ARIS, les champs observés sont strictement séparés (cf. Scheer, Architektur integrierter Informationssysteme 1992, p. 62). Pour cette raison, la vue des fonctions ne traite que les moyens de représentation qui montrent les rapports entre les fonctions. Les rapports entre fonctions et données, par exemple, sont représentés dans la vue des processus d'ARIS.

Une fonction est une tâche ou activité spécialisée exécutée sur un objet pour réaliser un ou plusieurs objectifs de l'entreprise (cf. Scheer, Architektur integrierter Informationssysteme 1992, p. 63).

Les fonctions sont représentées par des rectangles aux coins arrondis :

Illustration 6: Représentation de la fonction Vérifier la demande client

En général, une telle fonction doit être créée en se basant sur un objet d'information, comme une demande du client ou un ordre de fabrication. Il devrait également apparaître lors de la désignation d'une fonction. La figure ci-dessus apporte plus de précisions. **Demande client** définit l'objet, **vérifier** définit l'activité qui est effectuée sur cet objet. Aux niveaux supérieurs, cependant, il suffit d'utiliser un substantif pour désigner la fonction (Approvisionnement, Production, Ventes).

3.1.1.1 Arbre de fonctions

Les fonctions peuvent être décrites à différents niveaux de consolidation. Le niveau de consolidation supérieur est constitué de fonctions rassemblées sous la forme de processus d'entreprise ou de chaînes de processus. Il peut s'agir par exemple du traitement d'une commande client depuis le traitement de la demande du client jusqu'à l'expédition. Un tel processus d'entreprise est une fonction complexe qui peut être décomposée en sous-fonctions en vue de réduire sa complexité. Ainsi, le terme "fonction" peut être utilisé à tous les niveaux hiérarchiques. Cependant, des termes tels que procédure, processus, fonction partielle ou fonction élémentaire sont souvent employés pour une meilleure compréhension du niveau hiérarchique.

La décomposition des fonctions peut s'effectuer sur plusieurs niveaux hiérarchiques. Dans les arbres de fonctions sémantiques, les fonctions élémentaires forment le niveau inférieur.

Les fonctions élémentaires sont des fonctions qu'il n'y a pas lieu de décomposer plus avant pour répondre aux besoins de la gestion d'entreprise.

Cette décomposition est représentée dans des arbres de fonctions ou des modèles de hiérarchie.

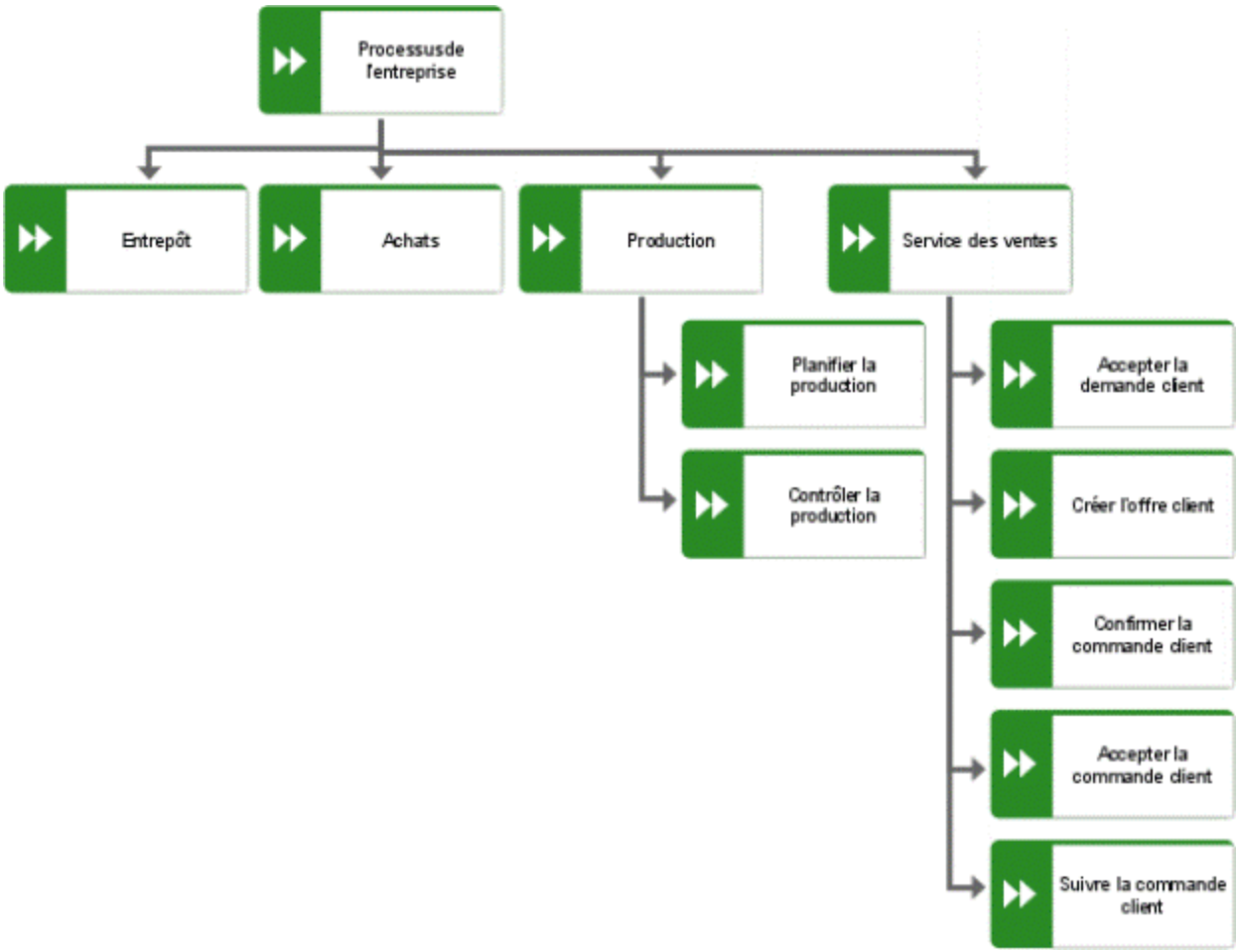


Illustration 7: Arbre de fonctions (extrait)

Dans un arbre de fonctions, les fonctions peuvent être regroupées suivant différents critères (cf. Brombacher/Bungert, Praxis der Unternehmensmodellierung, 1992). Parmi les critères fréquemment utilisés, on peut citer le traitement du même objet (orienté objet), la décomposition selon l'appartenance à un même processus (orienté processus) ou le regroupement de fonctions selon les opérations (orienté traitement).

La figure suivante illustre l'exemple d'une décomposition orientée objet. La fonction supérieure **Traiter l'ordre de fabrication** est subdivisée dans les fonctions **Créer ordre de fabrication**, **Modifier ordre de fabrication**, **Annuler ordre de fabrication**, **Activer ordre de fabrication**, **Confirmer ordre de fabrication** et **Surveiller ordre de fabrication**. Les fonctions décrivent différentes opérations (créer, modifier, annuler...) mais toujours au même objet **Ordre de fabrication**.

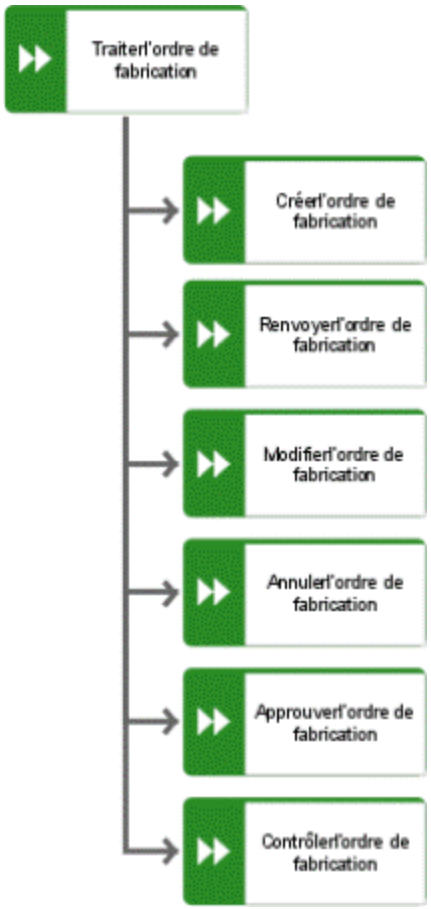


Illustration 8: Arbre de fonctions orienté objet

Si les arbres de fonctions représentent des résultats de modélisation de processus d'entreprise, la représentation d'arbres de fonctions orientés processus s'impose. La figure suivante illustre l'exemple d'une décomposition de fonction orientée processus.

Les fonctions **Accepter la commande client**, **Vérifier la commande client**, **Créer données client**, **Vérifier solvabilité du client**, **Vérifier disponibilité du produit** et **Confirmer ordre client** font partie des fonctions du processus d'entreprise **Traiter ordre de fabrication**. Contrairement à une décomposition orientée objet, les opérations sont exécutées sur différents objets (ordre client, disponibilité du produit).

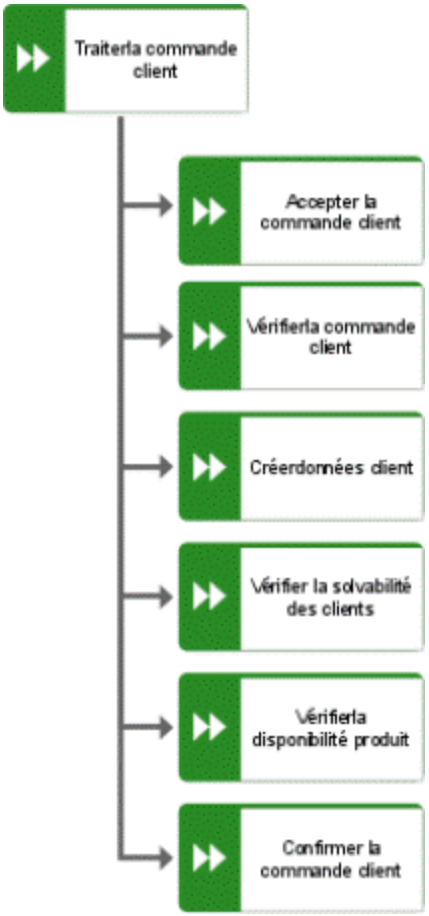


Illustration 9: Arbre de fonctions orienté processus

L'expression "Groupement orienté opération" signifie que toutes les fonctions exécutant les mêmes opérations (vérifier, créer, effacer) sur différents objets d'informations sont regroupées. Vous trouverez un exemple pour l'opération **Modifier** dans la figure suivante. Les fonctions représentées peuvent être présentes dans différents processus et traiter différents objets. La façon d'opérer sur les différents objets est toutefois toujours la même.

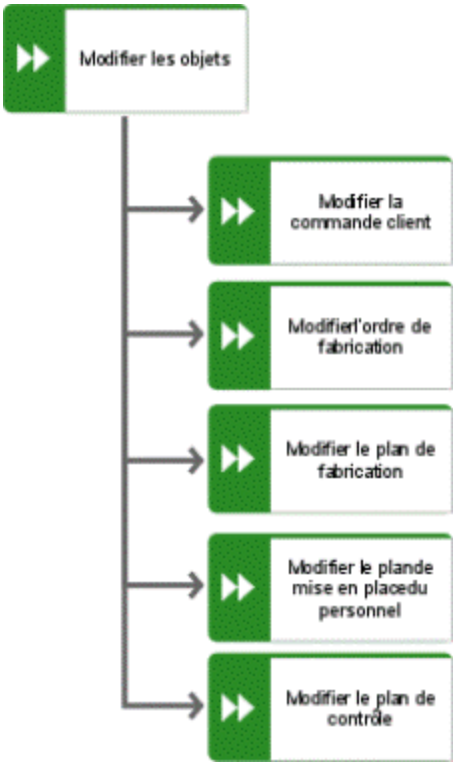


Illustration 10: Arbre de fonctions orienté traitement

La représentation de fonctions dans un arbre de fonctions permet certes une simplification, elle reste cependant statique. En plus de la représentation statique, il peut être intéressant de représenter l'enchaînement pour le déroulement temporel des fonctions. Les chaînes de processus événementielles (CPE) permettent de représenter les enchaînements temporels logiques. Les CPE contiennent, outre les fonctions, des événements servant de liens entre les fonctions. Les événements doivent être affectés à la vue des données d'ARIS. La séparation stricte entre les différentes vues d'ARIS fait que les chaînes de processus événementielles sont décrites dans la vue de gestion d'ARIS (cf. chapitre **Règles de gestion** (page 12)).

Dans la description relative aux règles de gestion des fonctions, la propriété de la décomposition des fonctions en fonctions partielles présente un intérêt particulier tout comme certaines autres propriétés d'une fonction, en particulier celles influant sur l'agencement des déroulements d'entreprise.

Ainsi, pour chaque fonction, il faut définir globalement au préalable si une saisie de l'utilisateur est nécessaire ou si la fonction peut être exécutée de façon quasi automatique. Des fonctions d'un même type pouvant être exécutées sans saisie de l'utilisateur peuvent être liées et traitées sans être ouvertes (procédure en différé).

Les informations sur la structure quantitative d'une fonction (exemple : nombre de demandes traitées en un jour) et la durée totale de l'exécution de la fonction constituent d'autres points de repère pour la réorganisation des processus d'entreprise. La durée totale peut encore être subdivisée en différentes étapes (temps de rodage, temps de traitement, temps d'attente). Dans ARIS, vous pouvez enregistrer ces informations dans les attributs du type d'objet **Fonction**. Vous trouverez une liste des types d'attributs disponibles dans l'aide **Référence de la Méthode ARIS**.

3.1.1.2 Diagramme des objectifs

Avant toute modélisation, analyse et optimisation de processus d'entreprise (Business Process Reengineering), il faut savoir quels sont les objectifs de l'entreprise vis à vis de la modélisation de processus d'entreprise.

Les objectifs (de l'entreprise) peuvent être définis et hiérarchisés dans les diagrammes des objectifs.

Un objectif permet de définir les futures stratégies de l'entreprise, qui sont à exécuter en tenant compte des facteurs de succès et de la réalisation de nouveaux processus d'entreprise.

Les facteurs de succès possibles pour atteindre un objectif peuvent être définis, classés et affectés aux objectifs dont ils soutiennent la réalisation.

Les facteurs de succès définissent les aspects à respecter pour atteindre l'objectif de l'entreprise correspondant. Ils sont affectés aux objectifs de l'entreprise dans le diagramme des objectifs.

La liaison de ce type de modèle aux autres types de modèles des règles de gestion a lieu grâce au type d'objet **Fonction**. Il est possible de représenter la fonction (le processus d'entreprise) de l'entreprise qui soutient la réalisation de l'objectif, et ce pour chaque objectif. Pendant la phase de modélisation et d'optimisation de processus d'entreprise, il est recommandé de prendre en compte la définition des priorités pour les objectifs définis ici et les fonctions affectées, lors de la définition de modèles de procédés.

La figure illustre un exemple de diagramme cible.

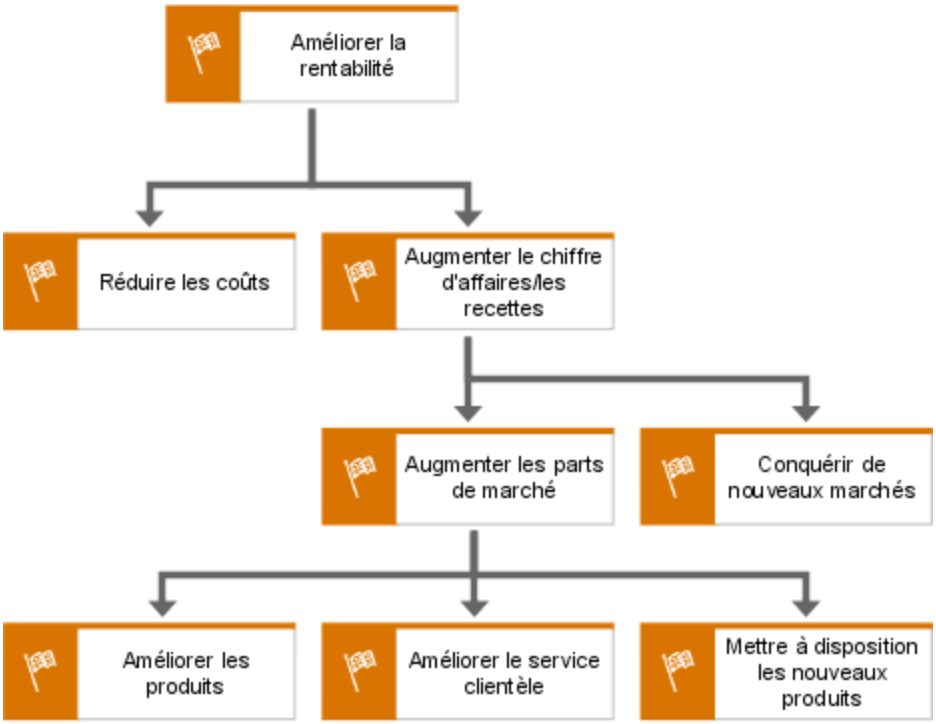


Illustration 11: Diagramme des objectifs

3.1.2 Concept informatique - diagramme de type d'application

Le concept informatique de la vue des fonctions comprend la définition de types d'application, de types de module, la structure modulaire des types d'application, le projet d'étapes, ainsi que la définition des présentations d'entrée et de sortie sous forme de concepts de listes et de masques.

La définition du concept informatique permet de répondre aux questions centrales suivantes :

- Comment soutenir les fonctions définies dans les règles de gestion en utilisant les types d'applications, les types de modules et les fonctions informatiques ?
- Quelle est la structure modulaire des types d'applications et des types de modules ?
- Quels sont les listes et les masques nécessaires à l'exécution d'une fonction ?
- Quelles sont les listes pouvant être créées avec un type d'application ou un type de module ?Quels sont les masques utilisés par les types d'applications et les types de modules ?
- Sur quelle base technologique (système d'exploitation, interface utilisateur ou systèmes de gestion de bases de données) repose un type d'application ?
- Quels objectifs de l'entreprise lui font utiliser un type d'application particulier ?

Le **Type d'application** est donc le type d'objet central du concept informatique de la vue des fonctions.

Contrairement à une application concrète qui n'est d'abord prise en compte qu'au niveau d'implémentation de la vue des fonctions et qui représente un système d'application particulier, identifiable dans l'entreprise grâce à un numéro de licence, un type d'application est créé par une caractérisation de toutes les applications fonctionnant sur la même base technologique.

Un type d'application représente l'assimilation à un type de différentes applications basées sur une technologie strictement identique.

Exemple : ARIS Architect représente un type d'application. Il est possible d'acquérir plusieurs licences et donc plusieurs applications différentes de ce type d'application.

Les types d'applications sont représentés graphiquement comme suit :

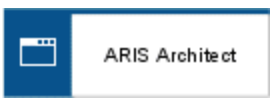


Illustration 12: Représentation graphique d'un type d'application

Les types d'application ont, en règle générale, une structure modulaire. La représentation de cette structure modulaire est permise par le diagramme d'application. Les types

d'applications sont composés de types de modules. La figure suivante vous en montre un exemple :

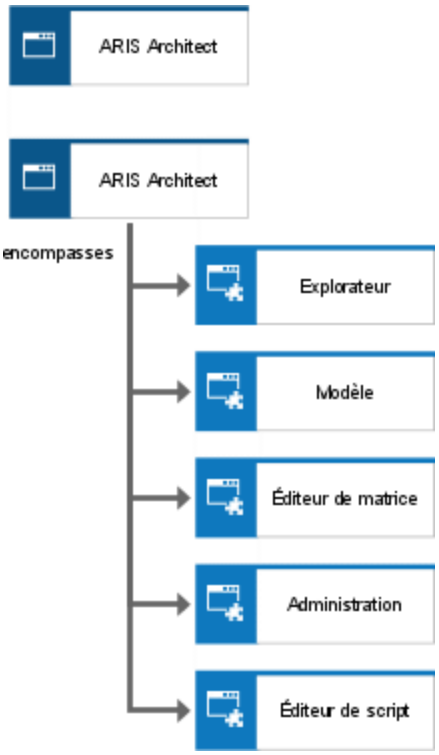


Illustration 13: Structure modulaire d'un type d'application

Dans l'exemple ci-dessus, ARIS Architect est composé des types de modules **Explorateur**, **Modèle**, **Éditeur de matrice**, **Administration** et **Éditeur de script** . Tout comme dans les types d'applications, il est question dans les types de modules de l'assimilation à un type de différents modules basés sur une technologie strictement identique. Les types de modules font partie des composants des types d'applications. Ils représentent en soi des composants pouvant fonctionner de manière indépendante.

Un type de module est un composant d'un type d'application pouvant fonctionner de manière indépendante. Les types de modules représentent l'assimilation à un type de différents modules basés sur une technologie strictement identique.

Les types d'applications et les types de modules peuvent être arbitrairement classés de manière hiérarchique. Les types de modules peuvent être divisés au niveau le plus bas en types de fonctions informatiques.

Au niveau d'une transaction, les types de fonctions informatiques sont les plus petites unités d'un type de module. Ils sont réalisés par différents sous-programmes et doivent toujours être exécutés complètement pour qu'une étape particulière de travail soit traitée.

Illustration 14: Représentation graphique du type de fonction informatique

Le diagramme de type d'application permet également de définir les fonctions des règles de gestion qui sont soutenues par les types d'applications et les types de modules définis. Cette affectation forme le lien entre les règles de gestion et le concept informatique de la vue des fonctions. Vous trouverez un exemple illustrant cela dans la figure suivante.

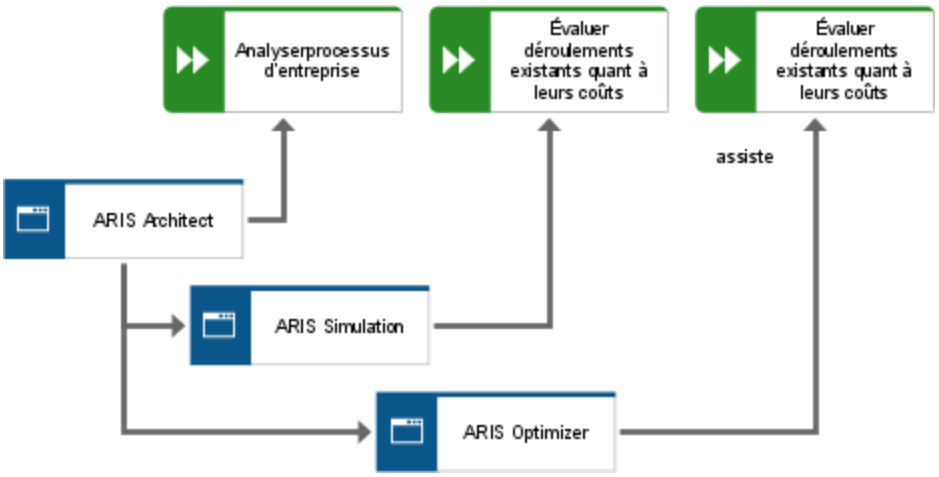


Illustration 15: Affectation des fonctions aux types d'application

Pour une définition plus précise de la base technologique des types d'applications et des types de modules, il est possible de leur affecter les différents types d'interfaces utilisateurs, les systèmes de gestion de base de données et les systèmes d'exploitation qui les font fonctionner ainsi que les langues de programmation qui permettent leur réalisation. Il est possible de trouver des désignations multiples, car il s'agit de types et non pas d'exemples concrets. C'est ainsi que les interfaces utilisateur **Windows 7** et **Windows 8** peuvent être affectées à un type d'application, ce qui signifie que la version du type d'application est compatible avec les deux interfaces utilisateur. Une relation univoque est nécessaire, uniquement lors de l'affectation des interfaces utilisateur à un exemple concret (à un type d'application) au niveau d'implémentation de la vue des fonctions. Cela décrit exactement la configuration des différentes licences du type d'application acquis par l'entreprise.

Cette figure illustre un exemple des affectations possibles dans le diagramme de type d'application.

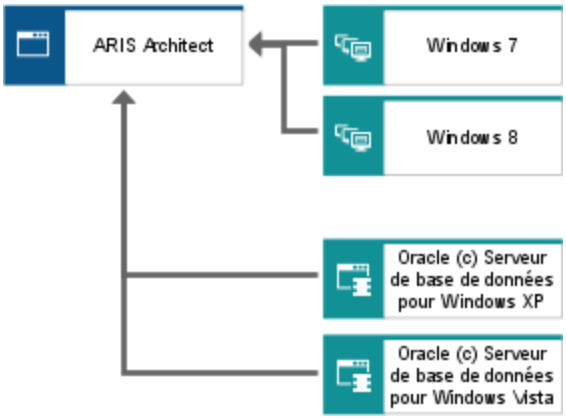


Illustration 16: Configuration du type d'application

Le traitement d'une fonction spécifique avec le soutien d'une application implique l'utilisation de différents masques d'écran et la création ou l'utilisation de différentes listes proposées par l'application correspondante. Les objets **Liste** et **Masque** permettent de représenter ce phénomène ; ils peuvent être affectés à la fonction spécialisée ou aux types d'applications et de modules.

Si vous souhaitez définir au préalable les déroulements généraux sans qu'il n'y ait de rapport avec les types d'applications concrets, les objets **Concept de liste**, et **Concept de masque** peuvent également être utilisés pour spécifier les masques et listes requis. Les deux types d'objets spécifient tout d'abord de manière générale le type de liste ou de masque devant être utilisé (par ex. **Entrée données client**) sans créer de rapport concret avec les listes ou les masques d'un type d'application. Ensuite, les concepts de listes et les concepts de masques peuvent être reliés avec les listes et les masques concrets. L'affectation détermine alors les possibilités de réalisation existantes. Un exemple est illustré dans la figure suivante.

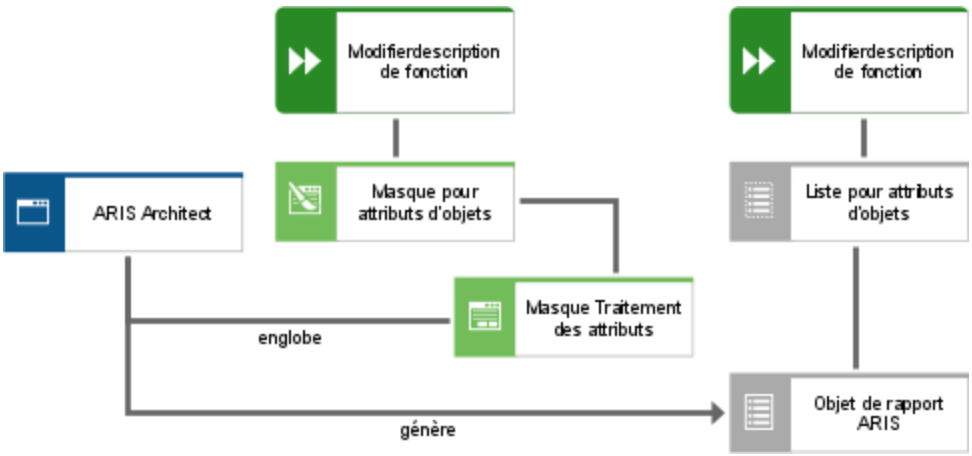


Illustration 17: Affectation de masques et listes

Vous trouverez une liste des types d'objets et des relations disponibles dans le diagramme de type d'application dans l'aide **Référence de la Méthode ARIS**.

3.1.3 Implémentation - Diagramme d'applications

Dans le diagramme de type d'application, il est possible d'affecter les applications et les modules concrets aux types d'applications et de modules du concept informatique. Il s'agit ici des exemplaires d'un type d'application dont dispose une entreprise, qui sont p. ex. identifiables grâce à leur numéro de licence.

Une application (module) est un exemplaire d'un type d'application (type de module) qui peut être identifié précisément grâce au numéro de licence.

Les applications et les modules sont représentés graphiquement comme suit :

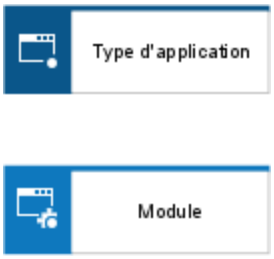


Illustration 18: Représentation graphique de l'application et du module

Comme il peut exister plusieurs licences pour un type d'application (type de module) dans une même entreprise, il est possible d'affecter plusieurs applications (modules) à un type d'application (type de module) dans un diagramme d'application.

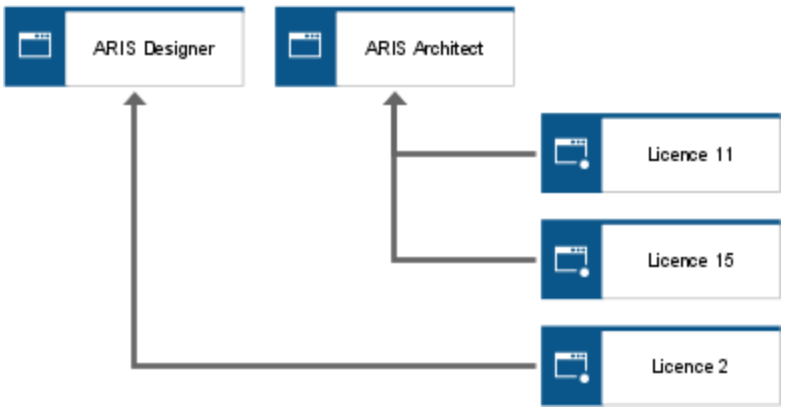


Illustration 19: Affectation des applications aux types d'applications

Le diagramme d'applications contient la structure modulaire réelle d'une d'application. Si tous les éléments modulaires d'un type d'application sont consignés dans le concept informatique, l'application n'a qu'une seule licence et les éléments modulaires de chaque licence peuvent être définis de façon univoque. Il peut donc exister, dans une même entreprise, plusieurs applications d'un même type ayant des structures modulaires différentes.

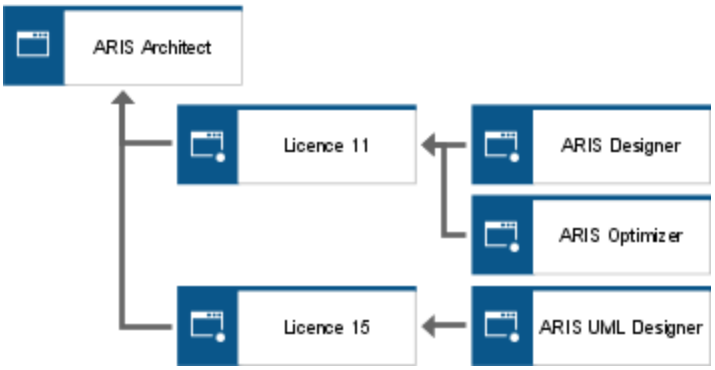


Illustration 20: Structures modulaires différentes de deux applications du même type

Le niveau d'implémentation permet, outre la représentation des applications et des modules réellement disponibles, la définition du transfert technique (physique) des applications sous forme de fichiers de programme.

Pour cela, il est possible de représenter, dans le diagramme de type d'application, les types de sous-programmes nécessaires à la réalisation d'un type d'application ou d'un type de module.

Un sous-programme est tout fichier de programme sur un support de données (p. ex. un fichier EXE ou COM) acquis lors de l'achat d'une licence. Un type de sous-programme apparaît en assimilant plusieurs sous-programmes basés sur une technologie strictement identique à un type.

L'affectation des types de sous-programmes à un type d'application et l'affectation des différents sous-programmes aux types de sous-programmes sont illustrées dans la figure suivante.

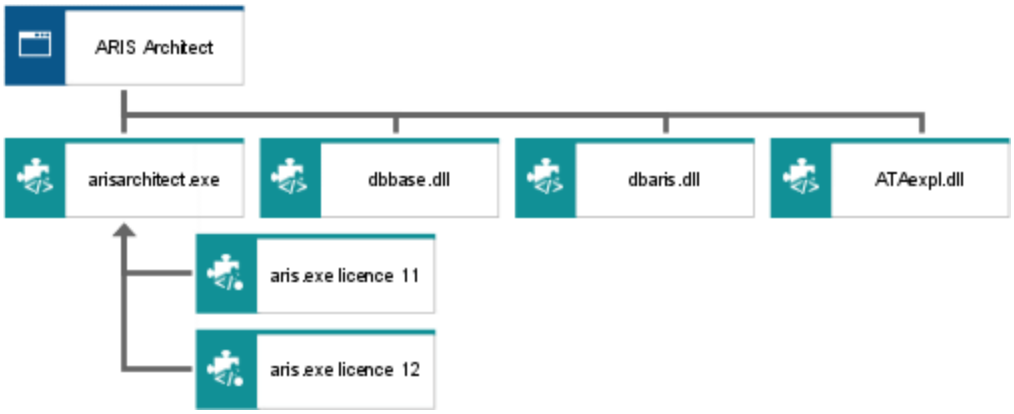


Illustration 21: Affectation de types d'application, types de sous-programme et sous-programmes

Le type d'application **ARIS Architect** se compose entre autres des types de sous-programmes **arisarchitect.exe**, **dbbase.dll**, **abaris.dll** et **ATAexpl.dll**. Au sein d'une

entreprise, il peut exister plusieurs exemplaires (sous-programmes) de chaque type de sous-programmes, grâce à l'achat de plusieurs licences ou bien grâce à la création de copies de secours.

Les types de sous-programmes et les parties de programmes peuvent être arbitrairement classés de manière hiérarchique. Pour une spécification technologique plus précise du programme, il est possible de représenter, dans le diagramme de type d'application, les accès des types de sous-programmes à des bibliothèques de programmes.

Vous trouverez une liste des types d'objets et des relations disponibles dans le diagramme de type d'application dans l'aide **Référence de la Méthode ARIS**.

3.2 Vue des données

3.2.1 Règles de gestion

Les règles de gestion de la vue des données contiennent la description du modèle de données sémantique du domaine de recherche concerné. Selon le principe de décomposition d'ARIS, les objets permettant la spécification des événements de départ et de fin, ainsi que les descriptions d'état du cadre pertinent d'une chaîne de processus, sont décrits.

Par rapport à la modélisation de fonctions, la modélisation de données est très exigeante du point de vue de la méthode. Dans la vue de fonction, un seul objet, la fonction, est pris en compte. En outre, seuls les ordres inférieurs et supérieurs sont représentés comme relations entre les fonctions.

La méthode de conception la plus répandue pour les modèles de données sémantiques est le modèle entité-relation (MER) de Chen (cf. Chen, Entity-Relationship Model 1976). Cette méthode de modélisation fait appel à des termes aussi divers que type d'entité, type de relation, attribut, etc. De nombreuses relations existent entre ces objets ; elles sont en outre bien plus difficiles à classifier que pour la modélisation de fonctions.

Vous trouverez ci-dessous une présentation de la modélisation avec des modèles entité-relation (MER). Nous vous expliquerons d'abord les objets et les relations du modèle de base selon Chen. Ce modèle de base sera ensuite complété par quelques règles dans le chapitre suivant.

3.2.1.1 Modèle de base MER

Le modèle de base fait une distinction entre les entités, les attributs et les relations. Le niveau de type peut fondamentalement être distingué du niveau de valeur.

Les entités sont des objets réels ou abstraits, qui présentent un intérêt pour la partie considérée des tâches exécutées au sein d'une entreprise.

Ainsi, un processus d'entreprise peut être un objet pris en compte. Les objets de données intéressants sont alors, selon le modèle de décomposition d'ARIS, les objets du cadre et les objets qui spécifient les événements. Voici des exemples d'entités d'un processus **Traitement de la commande client** :

- Client 1235,
- Article 471,
- Commande 11.

Les entités sont décrites plus précisément par des attributs (propriétés) déterminés. Un client peut donc être spécifié plus précisément par son nom, son prénom et son adresse.

Si des entités similaires sont regroupées en quantités, elles sont appelées types d'entités et leurs différentes valeurs sont les entités.

L'appartenance d'entités à une même catégorie peut être décrite par des attributs identiques. Le client **Dupont** et le client **Durant** peuvent être regroupés sous le type d'entité **Clients**, et l'article **4710** ainsi que l'article **4712** sous le type d'entité **Article**. Les types d'entités sont représentés par des rectangles dans le modèle ERM (cf. figure ci-dessous). Les types d'entités sont caractérisés par du texte en majuscules dans le paragraphe suivant.

Illustration 22: Exemples de types d'entités

Les attributs sont des propriétés permettant de décrire les types d'entités.

Les valeurs d'attributs sont des valeurs concrètes d'attributs de différentes entités. Par exemple, le client **1235** peut être décrit par les valeurs d'attributs **Durant**, **Eric**, **Strasbourg** etc. Les attributs correspondants sont **Nom**, **Prénom** et **Domicile**.

En règle générale, les attributs sont représentés par un ovale ou un cercle. Ci-dessous, nous utiliserons l'ovale comme moyen de représentation. La figure suivante affiche par exemple des attributs pour le type d'entité CLIENT.

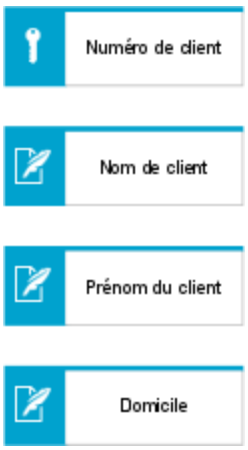


Illustration 23: Exemples d'attributs du type d'entité Clients

Une différenciation entre les types d'entités et les attributs est souvent très difficile et peut uniquement être faite dans le contexte de la modélisation. L'adresse d'un client peut donc par ex. être regroupée en une entité et non pas en attribut de CLIENT. Dans ce cas, un type

d'entité propre, ADRESSE, avec une relation vers CLIENTS, serait modélisé. Pour déterminer s'il s'agit d'un type d'entité ou d'un attribut, une caractéristique de différenciation simple existe : les entités possèdent des attributs, mais les attributs n'en possèdent pas. Donc, si un attribut est créé dans un MER que l'on souhaite décrire plus tard par d'autres attributs, il deviendra un type d'entité. Il est également pertinent de se poser la question suivante : Souhaitez-vous créer des relations entre un objet et d'autres types d'entités ? Si c'est le cas, l'objet considéré est également un type d'entité.

Une relation est une connexion logique entre des entités.

Les relations ne peuvent donc exister que si des entités existent aussi.

Si des relations similaires sont regroupées en quantités, ces relations sont appelées types de relations.

Un type de relation entre FOURNISSEUR et PIECE pourrait par ex. s'appeler FOURNIT. Les types de relations sont également écrits en majuscules dans le texte suivant. Dans le MER, les types de relations sont représentés par un losange et reliés aux types d'entités par des liaisons (cf. figure suivante).

Illustration 24: Exemple de type de relation

Lors de la désignation des types de relations, il n'est pas rare que seule une direction de lecture puisse entraîner des connexions pertinentes. L'exemple ci-dessus illustre la relation **Fournisseur fournit pièce**. Dans le sens de lecture inverse, le résultat serait **Pièce fournit fournisseur**, ce qui n'aurait aucun sens. S'il est impossible de déterminer le sens de lecture spécifique, veuillez sélectionner attentivement les termes supérieurs.

Il existe différentes catégories de types de relations. Les caractéristiques de différenciation sont, d'une part, le nombre de types d'entités qui les relient et d'autre part, le degré de complexité d'une relation.

Le nombre des types d'entités connectés par un type de relation peut être différencié par des relations à 1, 2 ou n chiffres.

Le degré de complexité ou la cardinalité indique le nombre d'entités d'un type d'entité qui peuvent être en relation avec une entité d'un autre type d'entité.

Les relations à différencier sont illustrées dans la figure suivante (cf. Scheer, Wirtschaftsinformatik 1994, p. 34).

Il existe quatre types de relations différents (cardinalités) :

- Relation 1:1,
- Relation 1:n
- Relation n:1 et
- Relation n:m.

Dans le cas d'une relation 1:1, chaque entité de la première quantité se voit attribuer exactement une entité de la deuxième quantité.

Une relation 1:n exprime que chaque entité de la première quantité se voit attribuer exactement une entité de la deuxième quantité, chaque entité de la deuxième quantité peut malgré tout posséder des relations vers plusieurs entités de la première quantité.

La relation n:1 exprime la même chose mais dans le sens inverse.

Si plusieurs entités de la seconde quantité sont affectées à chaque entité de la première quantité et inversement, la relation est une relation n:m.

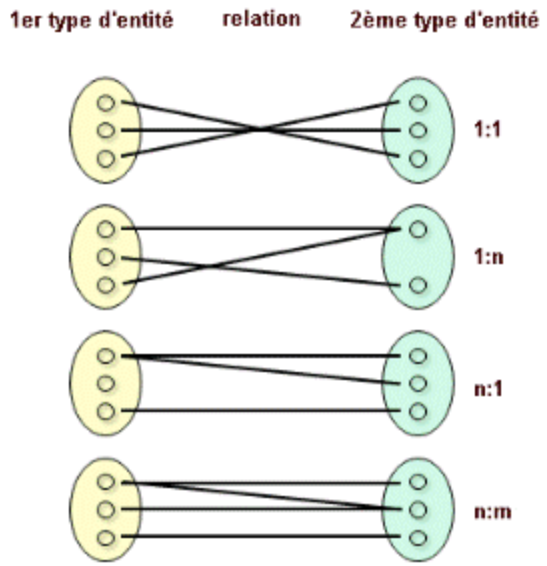


Illustration 25: Cardinalités des relations entre deux types d'entités

Les cardinalités du type de relation (type d'attribut **Degré de complexité**) sont écrites aux liaisons du modèle entité-relation.

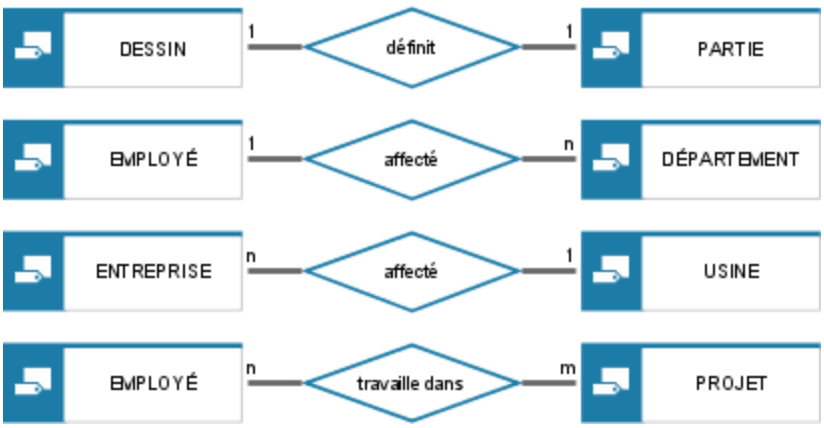


Illustration 26: Représentation des cardinalités dans MER

La cardinalité d'une entité indique, pour le type d'entité en question, le nombre maximum de relations d'un type de relation spécifique qu'elle peut avoir. Dans l'exemple de la relation n:1 de la figure ci-dessus, une entreprise du type d'entité ENTREPRISE peut apparaître dans plusieurs relations AFFECTE, car une entreprise est composée de plusieurs sites ; en

revanche, un site concret peut être associé au maximum à une relation AFFECTE, il doit donc être affecté univoquement à une entreprise.

l'ouvrage original de Chen indique une nouvelle interprétation de la cardinalité. Mais la notation représentée ici permet des formulations plus univoques, surtout lors de la représentation de relations entre plusieurs types d'entités. Afin d'éviter toute confusion, l'ouvrage original de Chen n'est pas discuté en détails ici.

Etant donné que des relations peuvent exister entre différentes entités d'un type d'entité, il est possible de représenter deux liaisons parallèles entre un type d'entité et un type de relation. Des noms de rôles peuvent alors être attribués aux liaisons pour permettre de les différencier. Un exemple de relations récursives est illustré dans la figure suivante. Une partie supérieure est composée de plusieurs parties inférieures, Une partie inférieure, en revanche, peut également être utilisée comme composant dans plusieurs parties supérieures.

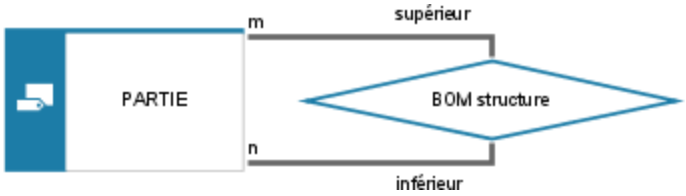


Illustration 27: MER pour une nomenclature

Les types d'entités ainsi que les types de relations peuvent être décrits par des attributs (cf. la figure suivante).

Les plages de valeurs des attributs sont appelées domaines.

Les affectations des éléments du domaine aux éléments des types d'entités et de relations représentent également des relations ; elles peuvent être représentées par une liaison caractérisée par le nom.

Il doit exister une relation 1:1 entre un type d'entité et au moins un domaine. Les valeurs de ce domaine identifient de manière univoque les différentes entités. C'est pourquoi elles sont aussi nommées attribut clé du type d'entité.

Dans l'exemple de la figure suivante (cf. Scheer, Wirtschaftsinformatik 1994, page 33), les entités de CLIENT sont identifiées de manière univoque grâce à l'attribut clé **Numéro client**.

Les relations sont identifiées par la fusion des attributs clé des entités connectées. Les attributs clé du type de relation HABITER sont donc le numéro client et le numéro adresse.

Les valeurs de domaines ayant une relation 1:n aux types d'entités ou de relations définissent les attributs descriptifs des différents objets de données.

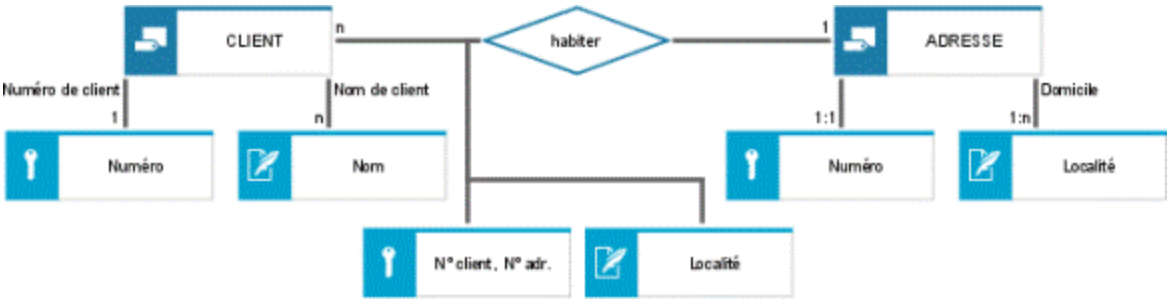


Illustration 28: Affectation d'attributs dans le MER

3.2.1.2 Extensions du MER : MERe

Le modèle de base de Chen a connu de nombreuses extensions au cours des dernières années. Ci-dessous, nous aborderons uniquement les extensions importantes pour la modélisation de la vue de données dans le concept ARIS.

3.2.1.3 Ajout d'opérateurs de construction

Les opérateurs de construction offrent une aide formelle lors de la création d'un modèle de données. Ils garantissent que les structures de données ne sont pas créées 'au hasard' et donnent à l'observateur d'une structure de données précise des indications sur le processus de conception. A partir de termes déjà introduits, de nouveaux termes sont créés par l'utilisation des opérateurs de construction. La création représente un processus intellectuel qui se déroule au niveau des connaissances techniques. L'examen des états de fait économiques d'un point de vue des structures de données entraîne soit la structuration d'états de fait connus sous un nouveau point de vue, soit l'apport de nouveaux rapports qui n'avaient pas encore été pris en compte.

Dans la multitude des différentes bases pour l'extension de la modélisation MRE, quatre opérateurs de construction ont fait leurs preuves (cf. Scheer, Wirtschaftsinformatik 1994, page 35 et suiv.)

- Classification,
- Généralisation,
- Agrégation et
- Groupement.

CLASSIFICATION

Lors de la classification, des objets semblables (entités) sont reconnus et affectés à un terme (type d'entité). Un objet est identique à un autre objet lorsqu'il est décrit par les mêmes caractéristiques (attributs).

La classification entraîne donc la recherche déjà décrite de types d'entités.

Illustration 29: Classification de clients

GÉNÉRALISATION/SPÉCIALISATION

La généralisation permet de regrouper des objets semblables dans un type d'objet générique.

Dans la figure suivante, les types d'entités **Client** et **Fournisseur** sont généralisés sous le terme générique **Partenaire commercial**. Les propriétés (décrites par les attributs), qui sont communes aux deux objets source, sont transmises au type d'objet généralisé. Seuls les attributs dont les types d'objets source diffèrent doivent donc encore être décrits. La création du nouveau type d'entité **Partenaire commercial** est représentée graphiquement par le triangle qui est également désigné comme relation is-a.

Si un terme générique est décomposé en termes partiels (**Partenaire commercial** est séparé en **Client** et en **Fournisseur**), on parle alors de spécialisation.

Il s'agit de la forme inverse de la généralisation. Les propriétés de l'objet généralisé sont transmises aux objets spécialisés. En plus de ces attributs transmis, les types d'objets spécialisés peuvent également posséder des attributs propres. La représentation de la spécialisation ne se différencie graphiquement pas de la représentation de la généralisation.

Cela s'exprime dans le graphique par le fait que les lignes de connexion ne sont pas représentées de façon orientée.

Illustration 30: Généralisation/Spécialisation

Alors que la spécialisation supporte surtout une procédure du haut vers le bas lors de la création d'un modèle de données, la généralisation est utilisée pour le support d'une procédure du bas vers le haut.

Dans le cadre de la spécialisation, l'intégralité ou la disjonction (alternative) des quantités partielles en cours de création peuvent être spécifiées lors de la constitution de quantités partielles.

Des quantités partielles non-disjointes existent lorsque la valeur d'un objet peut être représentée dans l'une ou l'autre des quantités partielles. Dans l'exemple ci-dessus, un client peut donc simultanément être un fournisseur. Si une valeur ne peut toujours être affectée qu'à exactement une quantité partielle, des quantités disjointes existent.

Une spécialisation complète existe lorsque tous les types d'objets spécialisés possibles sont affichés pour un type d'objet généralisé. Le type d'entité **Humain** peut, par exemple, être spécialisé dans les types d'entités **Femme** et **Homme** (cf. figure suivante). La spécialisation du point de vue du sexe serait donc complète.

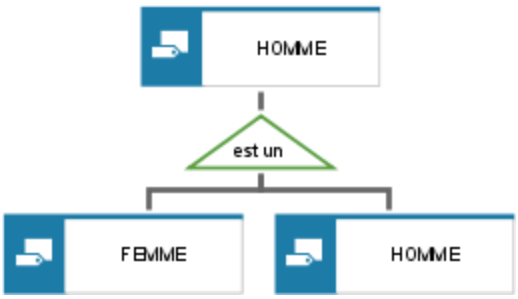


Illustration 31: Spécialisation décrite intégralement

Les combinaisons de ces critères entraînent les quatre valeurs de spécification plus complète suivantes pour une généralisation/spécialisation :

- disjoint/complet
- disjoint/incomplet,
- non disjoint/complet,
- non disjoint/incomplet.

AGRÉGATION

L'agrégation décrit la création de nouveaux types d'objets par le regroupement de types d'objets existants. Le nouveau type d'objet peut contenir de nouvelles propriétés.

Dans le MER, l'agrégation est exprimée par la création de types de relations (cf. figure suivante). Le nouvel objet **Gamme opératoire de commande** est créé par l'agrégation des types d'entité **Ordre de fabrication** et **Gamme opératoire**.

Illustration 32: Exemple d'agrégation

L'opérateur d'agrégation peut également être appliqué à des relations. Un type de relation existant est alors considéré comme type d'entité et peut donc à nouveau être lui-même point de départ pour la création de nouvelles relations. Vous en trouverez un exemple dans la figure suivante.

A partir des types d'entités **Ordre de fabrication** et **Gamme opératoire**, une première agrégation du type de relation **Gamme opératoire de commande** a été créée. Les attributs clé **Numéro d'ordre de fabrication** (NOF) et **Numéro de gamme opératoire** (NGO) constituent la clé complexe de la gamme opératoire de commande. Plusieurs phases de fabrication peuvent alors être affectées à la gamme opératoire de commande. Pour cette raison, la relation **Phase de fabrication de commande** est créée entre le type de relation **Gamme opératoire de commande** et le type d'entité **Gamme opératoire**. Etant donné que les relations peuvent uniquement être créées entre des types d'entités, le type de relation **Gamme opératoire de commande** doit être réinterprété. Le losange est alors entouré, comme vous pouvez le voir dans la figure suivante. Le type de relation réinterprété en cours de création est traité par la suite comme un type d'entité 'normal'. Pour continuer à représenter graphiquement la création du type de relation, les liaisons des types d'entités concernées par la création du type de relation sont tirées jusqu'au losange. Les liaisons sortantes du type de relation réinterprété et permettant la création de nouvelles relations sont seulement menées jusqu'aux bords du rectangle et ne touchent pas le losange dans l'intérieur du symbole.

Bien qu'il soit fondamentalement possible de remplacer les clés complexes par des clés simples, l'utilisation de la clé complexe permet de mieux comprendre le processus de création d'un modèle de données.

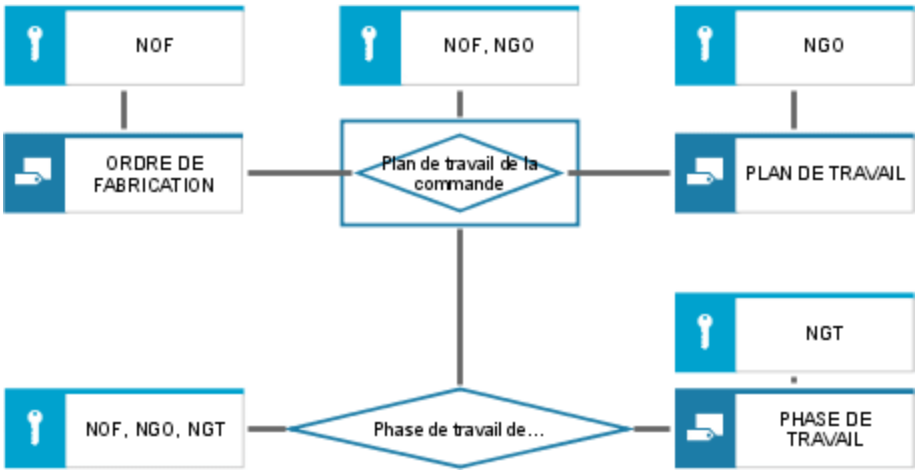


Illustration 33: Agrégation avec types de relations réinterprétés

Un MER permet de décomposer des rapports complexes en une structure claire. Cependant, comme la relation au complexe global n'est pas toujours très visible, des objets complexes sont introduits sous forme de clusters/modèles de données.

Un cluster/modèle de données donne un aperçu logique d'un regroupement de types d'entités et de relations, nécessaire à la description d'un objet complexe, d'un modèle de données.

Outre les types d'entités et de relations, un cluster/modèle de données peut également faire lui-même partie d'un cluster/modèle de données. Contrairement aux types d'entités et de relations, le cluster/modèle de données est donc hiérarchisable et favorise ainsi principalement une procédure du haut vers le bas lors de la création de modèle de données. La création de clusters/modèles de données peut également être d'une grande aide pour la fusion et consolidation de modèles partiels dans le cadre d'une procédure du bas vers le haut.

La figure suivante illustre la représentation graphique d'un cluster/modèle de données.

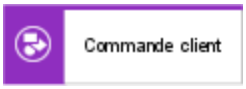


Illustration 34: Cluster de données (symbole graphique)

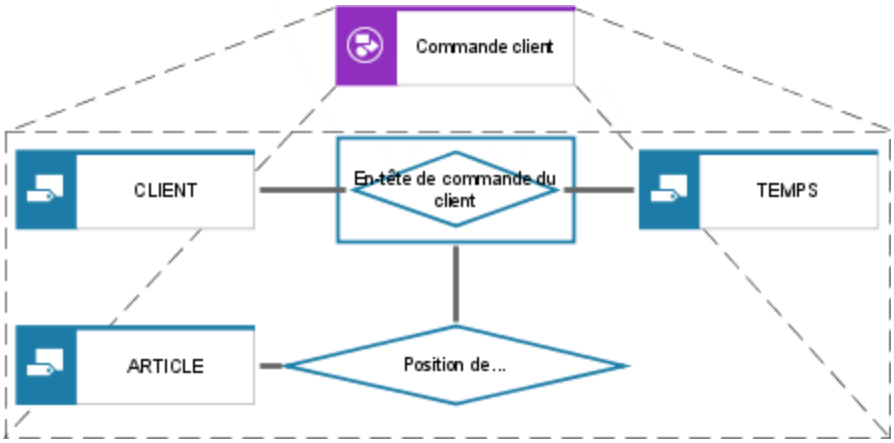


Illustration 35: Vue de cluster de données sur plusieurs objets

Le cluster/modèle de données est une vue logique sur plusieurs types d'entités et de relations. Les types d'entités et de relations **Client**, **Temps**, **En-tête de commande**, **Article** et **Poste de commande** sont nécessaires à la description de l'objet **Commande client**.

GROUPEMENT

Lors du groupement, des éléments d'une quantité d'entités sont rassemblés par groupes.

C'est ainsi que sont regroupés par ex. les moyens d'exploitation en groupe de moyens d'exploitation dans la figure suivante. Le groupe de moyens d'exploitation est un objet autonome qui peut être décrit plus précisément par d'autres attributs (nom du groupe de moyens d'exploitation, nombre de moyens d'exploitation). D'autres exemples sont le groupement de postes de travail en services ou le regroupement de lignes de commandes en commandes.

Illustration 36: Groupement

3.2.1.4 Extensions des cardinalités

Jusque là, lors de l'entrée des cardinalités, seule la limite supérieure des valeurs de relations possibles était affichée. Les cardinalités dans la figure suivante expriment qu'un projet au maximum peut se voir affecter plusieurs (m) collaborateurs, et qu'un collaborateur au maximum peut être concerné par plusieurs (n) projets.

Illustration 37: Limite inférieure/supérieure (1)

En plus de la saisie de la limite supérieure, la limite inférieure peut également être intéressante pour la spécification du nombre minimal des valeurs de relations. Pour cela, les cardinalités peuvent être représentées sous forme de paire de lettres (a, b) (cf. Scheer, Wirtschaftsinformatik). Dans la figure suivante, la paire de lettres (a1, b1) signifie que chaque projet peut participer à **au moins** a1 et **au plus** b1 valeurs de relations du type **travaille dans**, ce qui veut dire que chaque projet peut se voir attribuer **au moins** a1 et **au plus** b1 collaborateurs. La paire de lettres (a2, b2) signifie qu'un collaborateur peut participer **au moins** à a2 et **au plus** à b2 projets.

Illustration 38: Limite inférieure/supérieure (2)

Chaque relation est donc déterminée par deux degrés de complexité (minimum, maximum). Comme valeur pour la limite inférieure, on utilise souvent uniquement 0 et 1, la limite supérieure a comme plage de valeurs 1 <= max <= * (* signifie ici un chiffre au choix).

Une limite inférieure min = 0 signifie qu'une entité peut participer à une relation mais qu'elle n'y est pas obligée. La limite inférieure min = 1 signifie qu'une entité doit participer à au moins une relation.

Dans la figure suivante, la limite inférieure signifie qu'un collaborateur peut participer à une relation, mais qu'il n'y est pas obligé (min = 0), mais qu'un projet doit participer à au moins une relation (min = 1). Cela signifie que des collaborateurs peuvent exister sans qu'ils soient affectés à un projet ; en revanche, un projet peut exister uniquement si au moins un collaborateur lui a été affecté.

Illustration 39: Limite inférieure/supérieure (3)

Si seules les valeurs 0 ou 1 sont acceptées comme valeurs minimales et 1 ou * comme valeurs maximales, on distingue quatre types de notation (min, max) : (1,1), (1,m), (0,1) et (0,m). Dans ce cas, l'écriture abrégée suivante peut également être utilisée (cf. Schlageter/Stucky, Datenbanksysteme 1983, page 51) :

- 1 (correspond à (1,1))
- c (correspond à (0,1))
- m (correspond à (1,m))
- cm (correspond à (0,m)).

La figure suivante illustre l'exemple de la figure précédente avec cette notation.

Illustration 40: Limite inférieure/supérieure (4)

3.2.1.5 Dépendance d'identification et d'existence

L'extension des cardinalités par la saisie d'une limite inférieure et supérieure discutée dans le chapitre **Extensions du MER : MERe** (page 33) permet de définir certaines dépendances entre les objets de données.

Les types de relation et les types de relations réinterprétés n'existent pas par eux-mêmes par définition, mais par l'existence des types d'entités qui les relient entre eux. Ils sont donc dépendants de l'identification et de l'existence d'autres types d'entités.

Mais il existe également des types d'entités, qui, bien que possédant un attribut clé propre, dépendent de l'existence d'autres types d'entités. Ces dépendances sont par ex. basées sur l'opération de groupement. Dans la figure suivante, un service est uniquement valable lorsqu'au moins un poste de travail lui est affecté. Et inversement, la définition d'un poste de travail est uniquement pertinente lorsqu'il est affecté à un service. Comme dans la figure suivante, ces dépendances existentielles sont exprimées par l'entrée des degrés de complexités. Lors d'une notation (min., max.) par (1,1) et (1,*). La définition de dépendances existentielles dans le modèle de données entraîne la condition de l'intégrité référentielle de données dans une conversion ultérieure. Pour simplifier, le respect de cette condition garantit que les contenus d'une base de données sont toujours dans un état cohérent après certaines transactions. Pour l'exemple, cela signifie qu'un service ne peut être supprimé que lorsque tous les postes de travail affectés à ce service sont également supprimés.

Illustration 41: Dépendance existentielle

3.2.1.6 Modélisation de termes spécifiques dans un modèle de termes spécifiques de l'entreprise

Une des difficultés courantes de la modélisation - en particulier de la modélisation de données - est la diversité des termes qui existent pour des objets d'informations dans des grandes entreprises. Le service d'achat comprendra le terme **Commande** autrement que la production. L'acceptation des informations relevées peut être grandement facilitée par l'utilisation de termes propres à l'entreprise ou au service. C'est pourquoi la Méthode ARIS offre des modèles de termes spécifiques qui facilitent tant la gestion des différents termes dans le sens d'une gestion des synonymes pour les objets de données que la gestion des relations entre les objets des modèles de données (type d'entité, type de relation, ...) et les termes spécifiques à l'entreprise relevés.

Pour représenter ces relations, le type d'objet du Terme spécifique a été introduit. Plusieurs termes spécifiques peuvent être affectés à chaque objet d'information du modèle de données. Un exemple est illustré dans la figure suivante.

Illustration 42: Termes spécifiques (1)

Les termes spécifiques peuvent être en relation entre eux et peuvent également être hiérarchisés. La figure suivante illustre l'utilisation des différents types de liaisons entre les termes spécifiques.

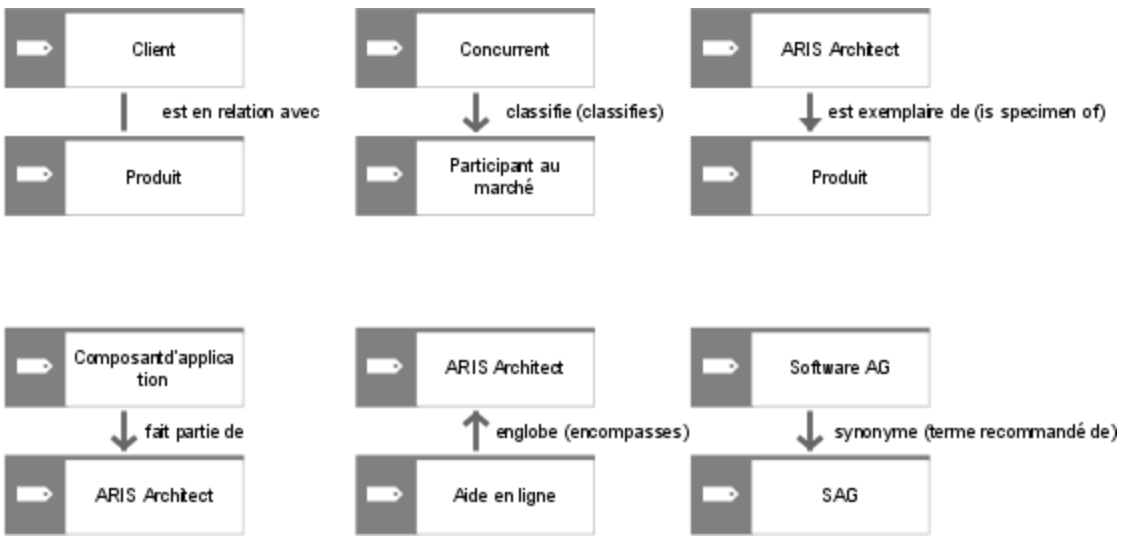


Illustration 43: Termes spécifiques (2)

Les termes spécifiques définis dans le modèle de termes spécifiques peuvent maintenant être employés dans d'autres types de modèles dans lesquels sont utilisés des objets d'informations, p. ex. des chaînes de processus permettant de représenter les données d'entrée/de sortie d'une fonction.

3.2.1.7 Diagr. de ratt. d'attributs du modèle MERe

Les modèles de données sous forme de représentations MERe, dans lesquels sont représentés seulement les types d'entités et les types de relations, ont en règle générale des structures relativement complexes. Si les attributs MER sont enregistrés dans ces modèles, ceux-ci deviennent illisibles.

Les diagrammes de rattachement d'attributs MERe permettent d'associer les affectations d'attributs MER dans un modèle propre à chaque type d'entité et de relation. Le type d'objet du MERe (type d'entité ou de relation) peut être intégré dans ce modèle par une copie de valeur et les relations avec des attributs MER peuvent être modélisées. Il est possible de distinguer s'il s'agit, pour l'attribut MER relié, d'un attribut clé, d'une clé externe ou d'un attribut descriptif. Un exemple est illustré dans la figure suivante.

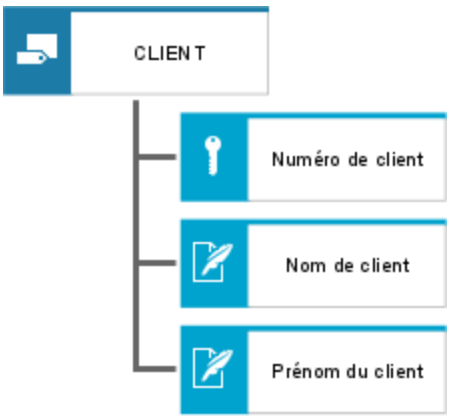


Illustration 44: Affectation d'attributs MER à un type d'entité

En plus de la représentation et de l'affectation d'attributs MER, il est également possible de représenter, dans ce type de modèle, les groupes de types d'attributs et leurs affectations.

Un groupe de types d'attributs représente le groupement d'attributs MER d'un type d'entité qui présentent une étroite homogénéité sémantique. A titre d'exemple, les attributs MER d'un type d'entité, qui forment une clé secondaire dans leur ensemble, peuvent être regroupés dans un groupe de types d'attributs.

Les groupes de types d'attributs sont représentés comme suit :

Illustration 45: Représentation d'un groupe de types d'attributs

Vous trouverez une liste des relations possibles dans le diagramme de rattachement d'attributs du MER dans l'aide **Référence de la Méthode ARIS**.

3.2.1.8 Modèle de données IE

Le modèle de données IE correspond à la notation de la modélisation de données de CASE-Tools Information Engineering Facility (IE) de la société Texas Instruments Inc.

Il s'agit également d'une notation qui ne connaît pas de type d'objet particulier pour les relations entre types d'entités.

La figure illustre un exemple de modèle de données en notation IE.

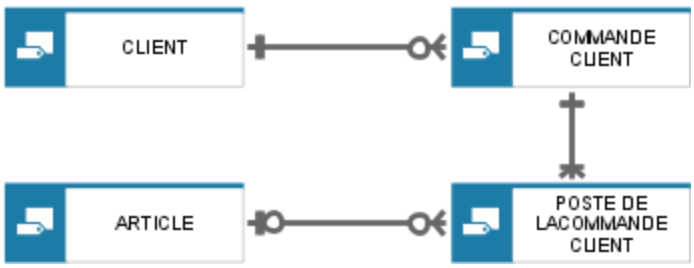


Illustration 46: Modèle de données en notation IE

3.2.1.9 Résumé des termes et des formes de représentations les plus importants du MERe

Les termes et formes de représentation des éléments structurels et des opérateurs de construction du modèle entité-relation étendu (MERe) sont résumés dans la figure suivante (cf. Scheer, Wirtschaftsinformatik 1994, p. 45).

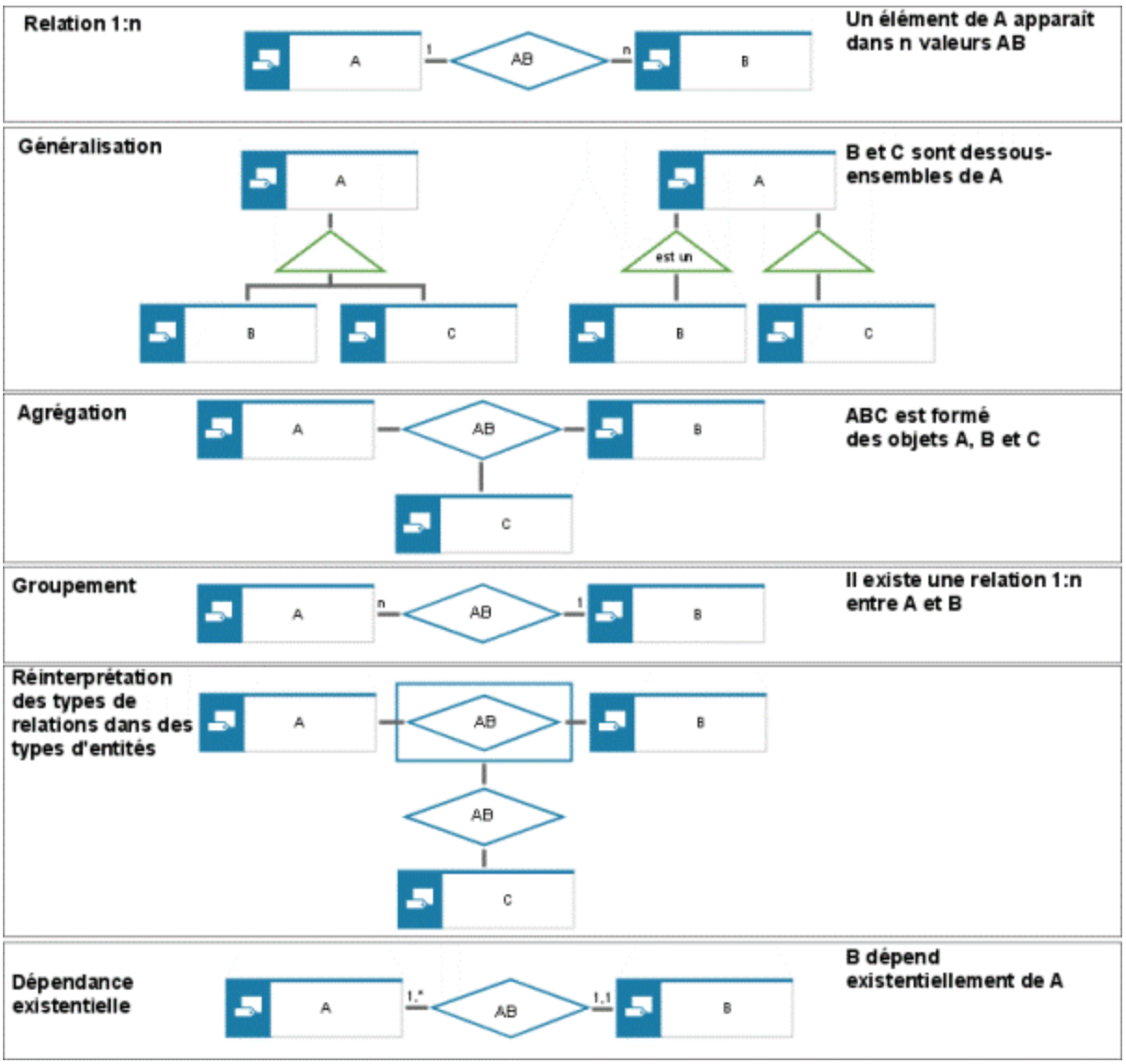


Illustration 47: Termes et formes de représentations du MERe

3.2.1.10 Modélisation de la structure Data Warehouse

Le diagramme de structure Data Warehouse sert à décrire la structure d'une Data Warehouse. Il représente avant tout la description statique, c'est-à-dire la relation des données entre elles et leurs sites. Dans l'architecture ARIS, cette manière de décrire est réalisée avec la vue des données. Les relations des informations ainsi que leur emplacement les unes par rapport aux autres sont considérées comme aspect central. Les dimensions de données sont décrites par l'Info cube. La coopération entre les dimensions est représentée par le schéma étoiles (Star scheme) (voir figure suivante). Une dimension peut donc servir comme clé pour en relier d'autres. Les objets des différentes dimensions peuvent adopter certaines valeurs qui sont cataloguées dans des tables des faits et peuvent être déterminées exactement par des indicateurs. Les dépendances sont décrites par des tables de dimensions avec leurs attributs clés et leurs caractéristiques. Les relations de hiérarchie des caractéristiques entre elles sont décrites à l'aide des structures d'arbres. Finalement l'affectation de dimensions aux tables de données de base peut être effectuée à l'aide du diagramme de structure.

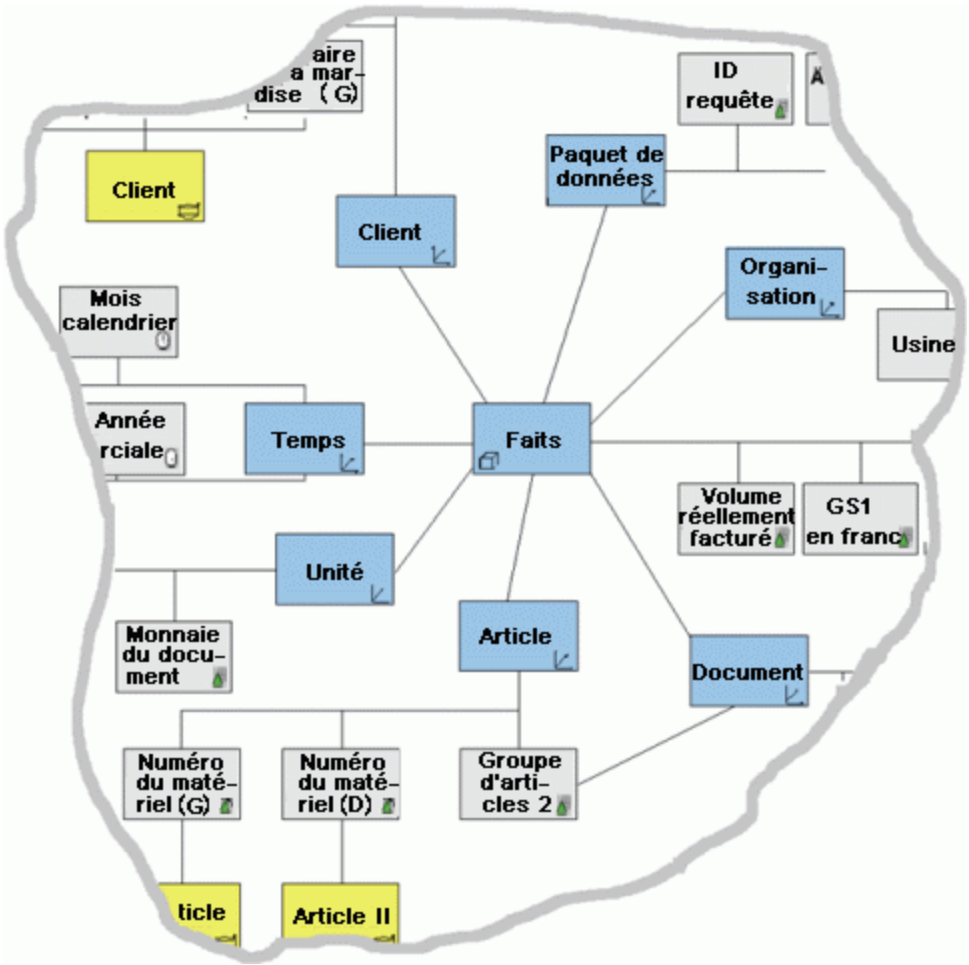


Illustration 48: Data Warehouse en schéma étoiles

3.2.1.11 Données de gestion de projets - Diagramme de support d'informations

Le diagramme de support d'informations est une partie optionnelle de la gestion de projet avec ARIS. Il est affecté aux règles de gestion de la vue des données et permet de documenter les données d'entrée et de sortie consignées sous forme de documents, de protocoles et de modèles ARIS.

Les documents réellement nécessaires (par exemple, un fichier de traitement de texte, peuvent être représentés explicitement et appelés par les attributs **Lien 1** à **Lien 4** depuis ARIS.

Illustration 49: Diagramme de support d'informations

3.2.2 Implémentation - Diagramme de table

Le diagramme de table permet la description de la table et des champs d'un système de base de données. La représentation graphique de tables et de champs est illustrée dans la figure suivante.

Illustration 50: Représentation graphique de tables et de champs

Pour chaque table, les champs individuels affectés à cette table peuvent être représentés. Pour une spécification plus précise il est possible d'affecter un indexe de tri et le domaine à chaque champ individuel. Un exemple est illustré dans la figure suivante.

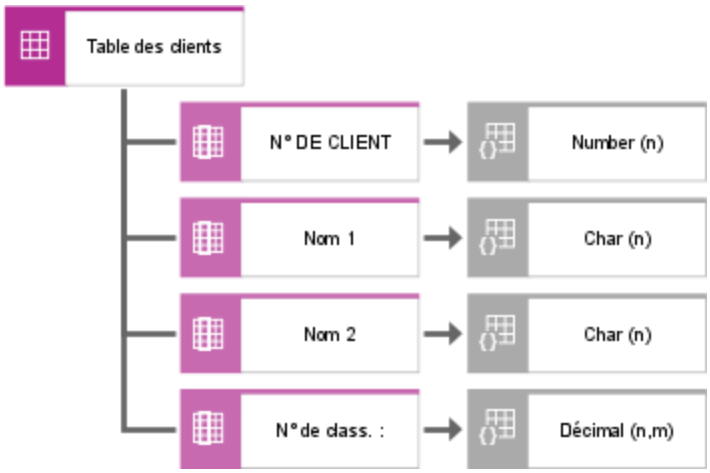


Illustration 51: Affectations de champs

Des relations multilatérales entre des tables et des relations ou des types d'entités peuvent apparaître. Dans le diagramme de table, ces relations peuvent être représentées par la sélection de la liaison correspondante.

La conversion ou la documentation des tables et champs de bases de données utilisés au sein de l'entreprise n'étant pas obligatoirement réalisée grâce à la définition d'un schéma de relations, il est possible de représenter les relations de réalisation entre des relations (ou des attributs) et des tables (ou des champs), ainsi qu'entre des types d'entités (ou des attributs MER) et des tables (ou des champs).

Il est possible de montrer soit les relations et les attributs qui sont réalisés grâce aux tables et champs représentés, soit (sans passer par les définitions de relations) les types d'entités, les

types de relations et les attributs MER représentés par ces tables et champs. Vous trouverez des exemples illustrant les deux formes de représentation dans la figure suivante.

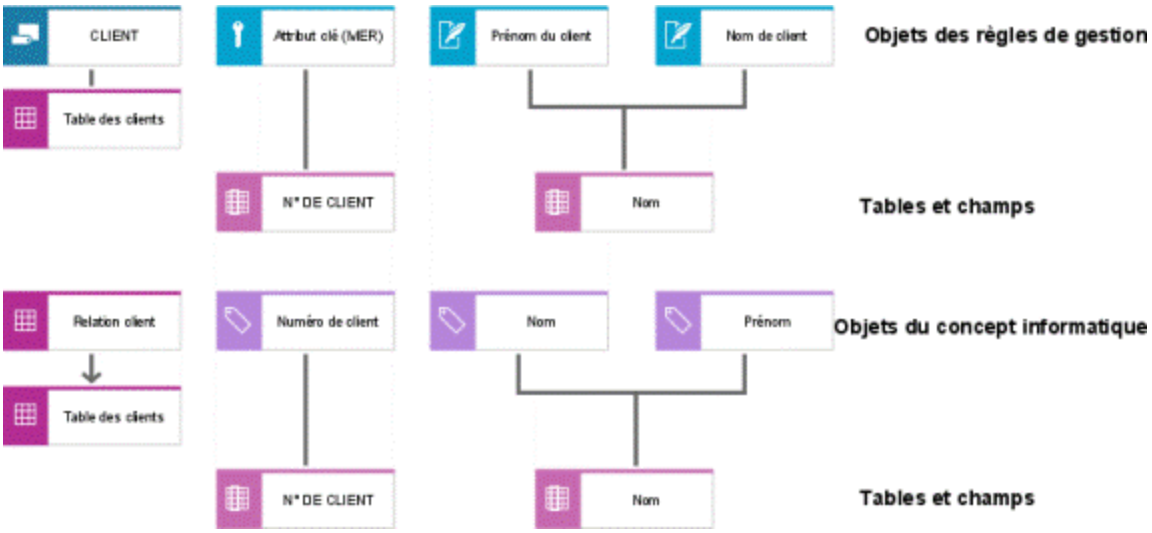


Illustration 52: Affectation des objets des règles de gestion et du concept informatique

Afin de pouvoir définir le site exact de tables et de champs particuliers dans l'entreprise, il doit également être possible de définir chaque exemplaire d'une table. La même chose est valable pour la définition de droits d'accès d'unités organisationnelles à des tables et des champs. L'objet Table définit au **Niveau de type** la structure logique d'une table physique et de ses champs. Il peut exister dans l'entreprise plusieurs exemplaires de ces tables ainsi définies, répartis sur plusieurs sites et enregistrés sur différents supports. Pour pouvoir représenter ceci, le système introduit les types d'objets Tables (exemplaire) et Champ (exemplaire).

Ces objets permettent de définir précisément le nombre d'exemplaires d'une table ou d'un champ. La figure suivante explicite ce rapport.

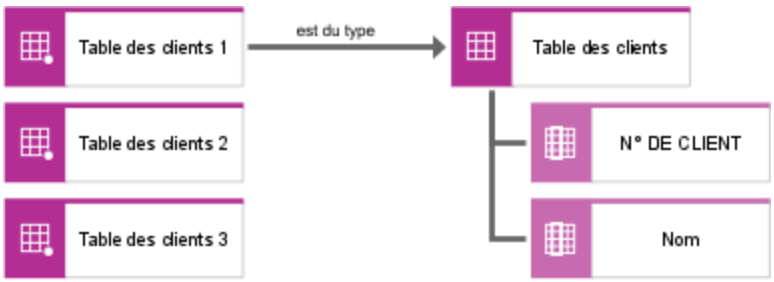


Illustration 53: Exemplaire de table

Vous trouverez une liste des objets et des relations disponibles dans un diagramme de tables dans l'aide **Référence de la Méthode ARIS**.

3.2.3 Diagramme d'affectation de rôles (RAD - Role allocation diagram)

Le modèle de référence SAP® est décrit à l'aide de CPE. Elles représentent les déroulements de l'entreprise à différentes étapes de spécification. Dans les CPE ayant le degré de détail le plus élevé qui sont appelées des **processus** dans la terminologie SAP®, sont montrés les déroulements des actions pendant le traitement des exécutables dans le système SAP®. Des rôles et des exécutables peuvent être affectés à ces processus.

Le modèle correspondant dans ARIS est le diagramme de rattachement de fonctions.La CPE contenant le processus modélisé doit être associée à la définition de fonction correspondante. Il est ainsi possible de déterminer dans le diagramme d'affectation de fonctions les rôles nécessaires pour lancer les exécutables. Comme il n'existe pas de relations directes entre les rôles et les exécutables, il est impossible de décider, lors de l'apparition de plusieurs rôles, de quels exécutables ils sont responsables. L'affectation des rôles à des exécutables s'effectue donc dans le diagramme d'affectation de rôles (RAD). Un rôle est représenté par colonne. Les exécutables sont placés dans les colonnes, ce qui crée des relations implicites.

Les informations peuvent être utilisées pendant l'introduction de R/3 pour créer les profils d'utilisateurs et les concepts d'autorisation nécessaires pour l'exploitation du système SAP®.

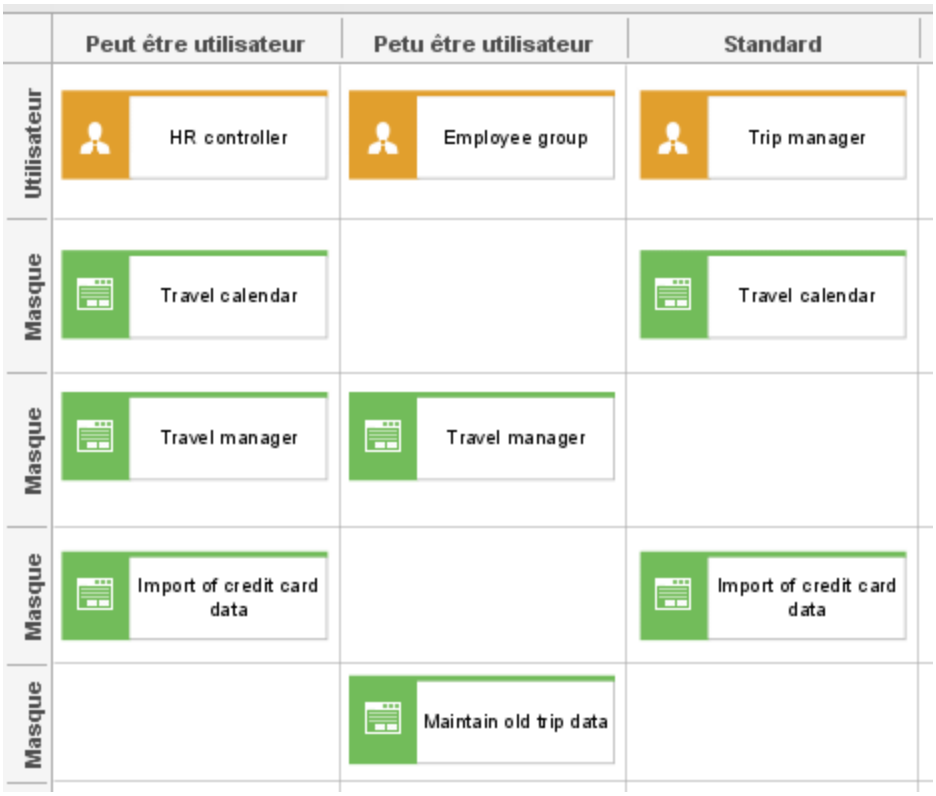


Illustration 54: Diagramme d'affectation de rôles (RAD - Role allocation diagram)

3.3 Vue organisationnelle

3.3.1 Règles de gestion

3.3.1.1 Conceptualisation organisationnelle des entreprises

Les entreprises sont des configurations sociales complexes qui peuvent être divisées en unités synoptiques. Afin de maîtriser leur complexité, des organigrammes sont définis et des règles sont fixées. Le résultat de ce processus de structuration est appelé " organisation ". Par le passé, peu d'études ont été faites pour analyser le rôle des critères organisationnels par rapport à l'évolution des systèmes d'information. Des concepts d'entreprise modernes comme la production ou la gestion en flux tendu ou la production intégrée par ordinateur ont cependant un rapport étroit avec la conceptualisation organisationnelle de l'environnement d'analyse. C'est pourquoi le concept ARIS possède une vue descriptive spécifique pour l'organisation.

Dans la structuration organisationnelle de l'entreprise, on peut établir une distinction entre la structure organisationnelle et l'organisation opérationnelle.

L'organisation structurelle rassemble les règles relatives à la structuration statique de l'entreprise. L'organisation opérationnelle comprend les règles relatives aux tâches à exécuter par l'entreprise. Cette structure relative aux tâches dans l'optique d'une répartition de fonctions sur les responsables des tâches est traitée dans la vue de gestion de la maison ARIS. La vue organisationnelle comprend ainsi essentiellement la représentation de la configuration de l'organisation structurelle.

L'agencement de l'organisation optimale d'une entreprise dans le but de réduire au maximum les dépenses en coordination se définit par rapport aux conditions environnementales et aux objectifs prévus. Ainsi, il est impossible de définir des structures organisationnelles optimales universelles dans le sens de structures de références.

La structure des unités organisationnelles peut s'orienter par rapport à différents critères.

La décomposition fonctionnelle est très répandue. Dans cette structure, tous les produits et domaines relèvent de chacune des fonctions de l'entreprise (approvisionnement, production, finances et comptabilité, ventes). L'avantage de la forte spécialisation des collaborateurs est toutefois contrebalancé par les coûts élevés de communication et de coordination entre les secteurs fonctionnels.

La conception et l'utilisation de systèmes d'information se sont également longtemps orientées selon cette décomposition fonctionnelle des entreprises. En observant des chaînes de processus globales dans l'optique de l'édition jointe d'objets de données de même catégorie, il est toutefois assez difficile de trouver le rapport entre les fonctions distinctes avec une telle structure.

Au sein du débat autour du traitement intégré de données, une banque de données unifiée pour soutenir les différentes fonctions est devenue inéluctable. L'intégration visée des fonctions mène néanmoins au résultat suivant : l'annulation de l'objectif de réduction de la complexité par la décomposition fonctionnelle.

C'est pourquoi d'autres critères de décomposition fonctionnelle sont souvent utilisés lors de la définition de l'objectif de l'intégration de la fonction.

Une décomposition selon des critères comme les zones ou les produits peut par ex. avoir lieu d'être. Une décomposition par produit est représentée schématiquement à la figure suivante (cf. Scheer, Wirtschaftsinformatik 1994, pp. 26 et suiv.).

Lors d'une organisation structurelle par zone, les unités organisationnelles sont définies selon la répartition géographique de l'entreprise ou du secteur d'entreprise. Cette organisation est particulièrement pertinente pour des fonctions de distribution, car les facteurs régionaux, comme les législations différentes, peuvent être pris en compte de manière plus précise.

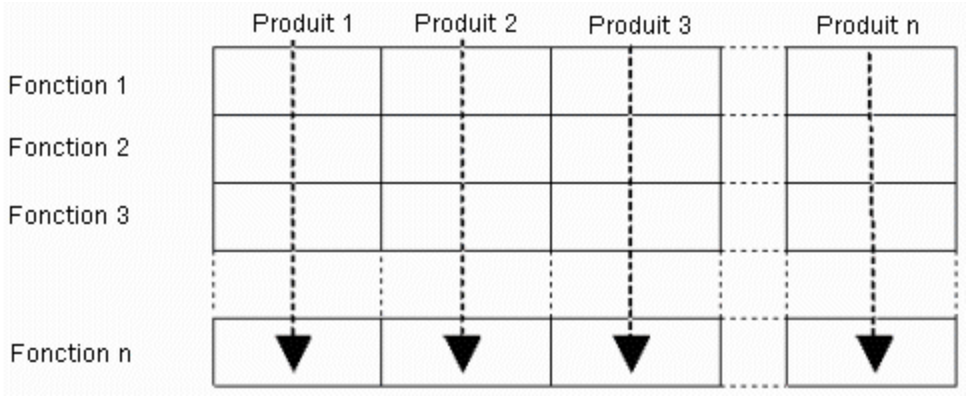


Illustration 55: Décomposition organisationnelle par produit

Une organisation structurelle par produit définit les unités organisationnelles pour les produits ou les groupes de produits. Ainsi, un maximum de fonctions possibles et pertinentes pour le groupe de produits correspondant sera intégré à un groupe de produits. L'objectif étant de réduire les dépenses en communication dues à la décomposition fonctionnelle. En revanche, des dépenses en médiation apparaîtront entre les systèmes partiels orientés groupes de produits.

Des formes organisationnelles mixtes sont souvent formées pour contrer ces effets. Ce type est représenté pour l'exemple Achat à la figure suivante (cf. Scheer, Wirtschaftsinformatik 1994, pp. 26 et suiv.).

	Groupe de produits 1	Groupe de produits 2	Groupe de produits 3
Achat central		Choix des fournisseurs	
		Création du contrât	
Disposition			
Commande			
Contrôle des factures			

Illustration 56: Formes organisationnelles mixtes

Une décomposition uniquement fonctionnelle reviendrait à avoir un achat central responsable de tous les groupes de produits. Ainsi, tous les effets de synergie existant entre les groupes de produits pourraient être mis à profit, mais l'exécution d'un processus d'achat par toutes les fonctions partielles mènerait à de graves problèmes de prise de décision. Une répartition des fonctions d'achat parmi les différents groupes de produits a pour conséquence la création d'un service Achats par groupe de produits effectuant toutes les fonctions d'achat. Les effets de synergie, par ex. lors du choix des fournisseurs et de la négociation des contrats type, ne pourraient avoir lieu qu'avec un effort accru en prise de décision.

Dans la décomposition représentée ci-dessus, les fonctions d'achat, pour lesquelles beaucoup d'effets de synergie sont attendus, sont décomposées de façon fonctionnelle, c-à-d. qu'elles sont prises en compte par un poste d'achat central. Les fonctions devant respecter des besoins et des restrictions spécifiques à certains groupes de produits distincts sont décomposées de façon orientée objet par groupe de produits. Elles peuvent ainsi être impliquées en temps réel dans les déroulements de processus des différents groupes de produits. Le déroulement en soi des processus est donc exécuté dans des unités décentralisées, alors que les relations existantes entre les unités décentralisées sont prises en compte par le niveau de coordination supérieur et central.

Le concept ARIS accorde une attention particulière aux formes organisationnelles flexibles grâce à la perspective nettement orientée processus. La constitution de structures organisationnelles tenant compte simultanément de différents critères de décomposition s'impose tout particulièrement pour les approches orientées comptabilité, comme le concept de centre de profit.

3.3.1.2 Organigramme

L'organigramme est une façon typique de représentation des structures d'organisation. Il permet, selon les critères de structuration choisis, de représenter les unités organisationnelles formées (en tant que responsables des tâches) et leurs interconnexions.

Les unités organisationnelles sont les responsables des tâches à accomplir pour réaliser les objectifs de l'entreprise.

Les liens représentent les relations entre les unités organisationnelles. Vous trouverez un exemple illustrant cela dans la figure suivante.

Pour une spécification plus précise des relations de subordination, différents types de liaisons existent entre les unités organisationnelles. Une liaison peut alors signifier :

- est le supérieur professionnel
- est le supérieur hiérarchique
- est constituant pour

Lorsque les compétences fonctionnelles sont saisies, l'organigramme représente la répartition des tâches d'entreprise.

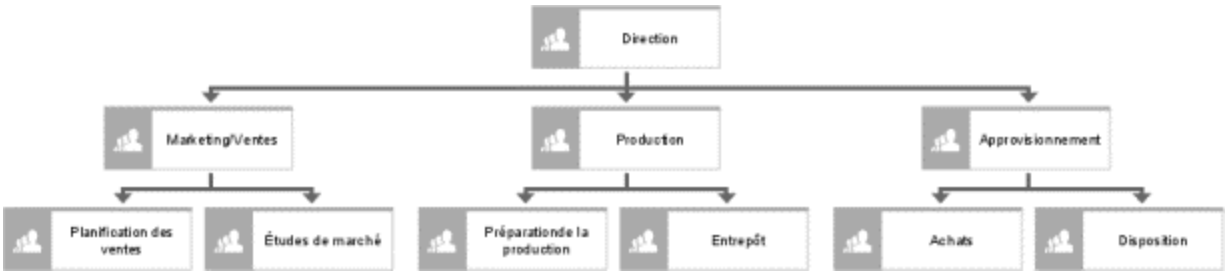


Illustration 57: Organigramme

Pour représenter certains postes de l'entreprise pour lesquels il existe, p. ex., des descriptions de postes, il existe un type d'objet spécial **Poste de travail**. Sa représentation est illustrée dans la figure suivante. Il est possible d'affecter plusieurs postes à une unité organisationnelle. La signification des liaisons est la même que la signification des liaisons entre les unités organisationnelles.

Des personnes, occupant ces postes, peuvent être affectées aux unités organisationnelles et aux postes. ARIS propose des objets propres également pour les personnes, comme illustré par la figure suivante. L'affectation d'une personne à une unité organisationnelle indique que cette personne est affectée comme collaborateur à une unité organisationnelle ; l'affectation à un poste définit l'occupation actuelle des postes dans l'entreprise. La figure en montre un exemple.

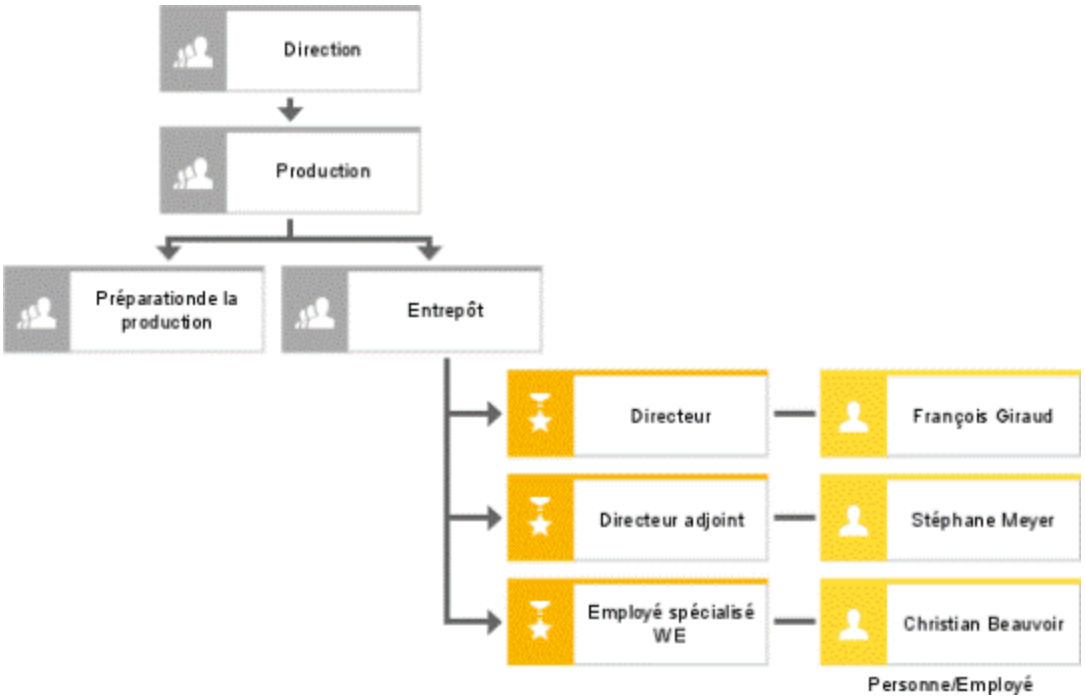


Illustration 58: Organigramme avec affectation de personnes et de postes

Les unités organisationnelles et les personnes peuvent également être caractérisées. Il est donc possible de définir, pour une unité organisationnelle, s'il s'agit p. ex. d'un service, d'un département ou d'un groupe ; les personnes peuvent être affectées p. ex. aux rôles **Chef de service**, **Chef de groupe** ou **Chef de projet**.

Les objets **Type d'unité organisationnelle** et **Rôle** permettent de représenter cette caractérisation. La figure illustre un exemple d'assimilation d'unités organisationnelles et de personnes à un type.

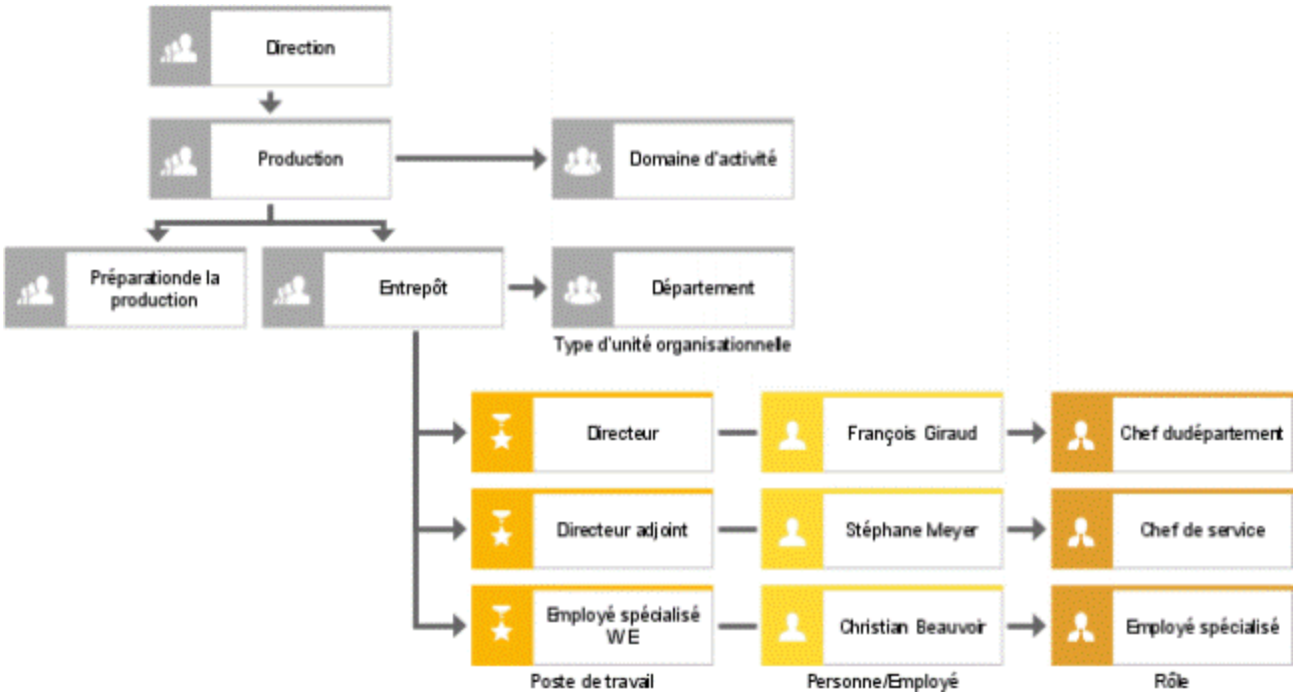


Illustration 59: Types de personnes

L'utilisation de ces types d'objets permet de représenter des règles d'entreprise générales qui sont dérivées des unités organisationnelles concrètes ou des personnes de l'entreprise. Il est donc possible de définir dans les chaînes de processus que seuls certains rôles ont le droit d'exécuter une fonction ou que seuls certains rôles peuvent avoir accès à un objet d'information.

La modélisation de la structure organisationnelle de l'entreprise est le point de départ de la topologie de réseau devant être définie au niveau du concept informatique ; la topologie de réseau doit soutenir le mieux possible la structure organisationnelle. La définition de la topologie de réseau comprend les liaisons de réseau et les nœuds de réseau, se trouvant à des endroits spécifiques au sein de l'entreprise. Le site d'une unité organisationnelle est donc le lien le plus important entre les règles de gestion et le concept informatique de la vue organisationnelle. C'est pour cela qu'il est possible d'affecter à chaque unité organisationnelle le site sur lequel elle se trouve, dès les règles de gestion. La figure en montre un exemple.

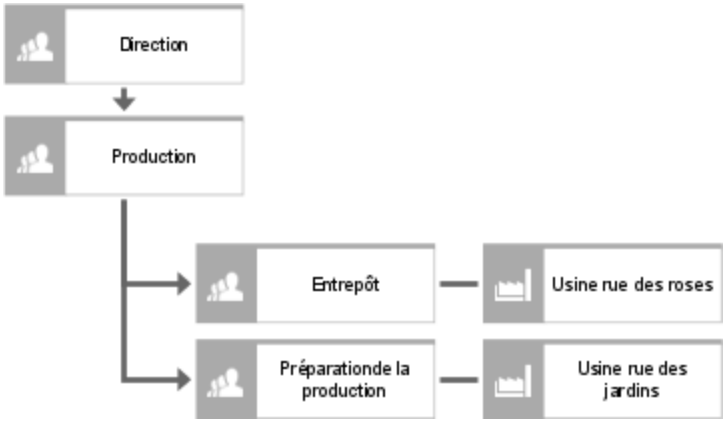


Illustration 60: Affectations de sites

Les sites peuvent être classés arbitrairement de façon hiérarchique. Un site peut être une usine, un bâtiment ou également, dans le cas d'une observation détaillée, un bureau, voire un poste de travail dans une pièce. C'est pourquoi il est possible d'affecter, au sein d'un concept informatique, les nœuds d'un réseau aux différents postes de travail de l'unité organisationnelle, c.-à-d. de définir par ex. que dans un bureau (bureau n°102) il y ait trois noeuds de réseau disponibles au total.

La figure illustre l'exemple d'une hiérarchisation des sites.

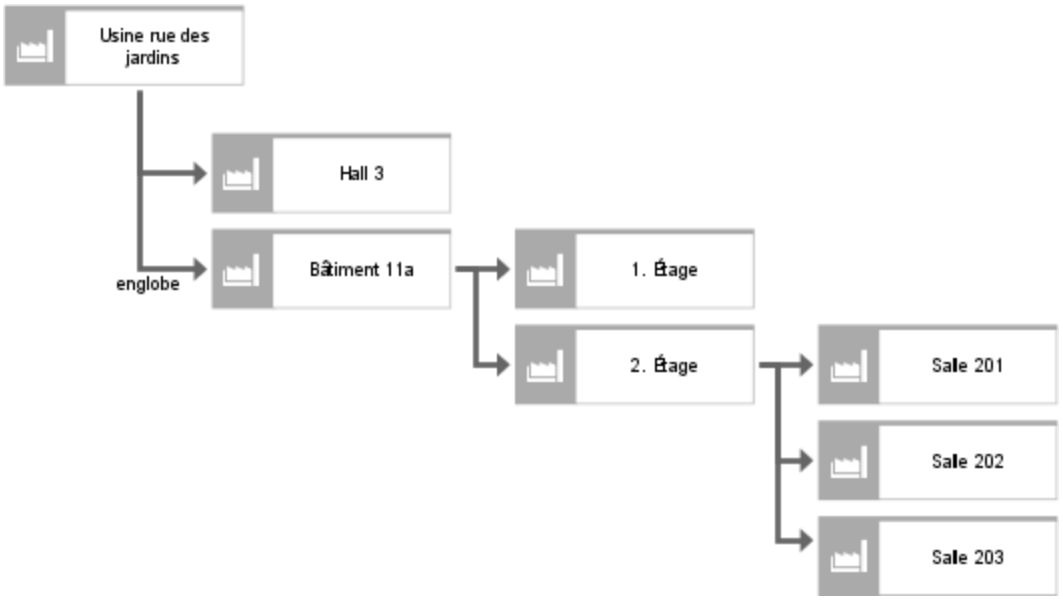


Illustration 61: Hiérarchies des sites

3.3.2 Concept informatique - Topologie de réseau

La structure de l'entreprise représentée dans l'organigramme peut désormais être gérée par des systèmes de communication et d'information. Les exigences posées à l'organisation structurelle de ces systèmes d'information peuvent être définies en premier lieu dans le concept informatique sous forme de topologies de réseau.

Dans un premier temps, il est possible d'intégrer différents types de réseaux dans le modèle **Topologie de réseau**.

Un type de réseau représente l'assimilation à un type de différents réseaux basés sur une technologie strictement identique.

La figure illustre un exemple de type de réseau :

Illustration 62: Représentation graphique d'un type de réseau

Les types de réseaux peuvent être reliés entre eux et peuvent être hiérarchisés en tant que structures logiques.

Il est possible d'affecter d'autres types de nœuds de réseaux et d'autres types de liaisons de réseaux à chaque type de réseau. Ainsi, des restrictions technologiques résultant de la sélection d'un type de réseau déterminé pour une entreprise peuvent immédiatement être reconnues. Pour chaque type de liaison de réseau, il est possible de représenter le type de nœud de réseau dans lequel il peut arriver.

Pour les types de composants matériel, il peut s'agir de matériel réseau pour la réalisation de structures de réseau définies ou de types de composant matériel qui peuvent être connectés à des types de nœuds de réseaux.

Comme pour les types d'applications ou pour les types de réseaux, il ne s'agit pas non plus pour les types de composants matériel d'exemplaires de composants matériel individuels pouvant p. ex. être identifiés par un numéro d'inventaire de l'entreprise, mais d'assimilations sur une même base technologique. Les types de composants matériel peuvent être classés hiérarchiquement arbitrairement. Les types de composants matériel peuvent être classés hiérarchiquement selon vos souhaits.

Un type de composant matériel représente l'affectation à un type de plusieurs exemplaires de composants matériels, basés sur une technologie strictement identique.

Avec les types de liaisons et de nœuds de réseaux, il est possible de construire une sorte de modèles de référence de la topologie de réseau. Il est possible de représenter les types de composants matériel pouvant être utilisés pour la réalisation de certains types de liaisons de réseau ou de certains types de nœuds de réseau. Pour les types de liaisons, il peut s'agir p. ex.

d'un type particulier de câble de transmission. Il est également possible de représenter quel type de composant matériel peut être connecté à quel type de nœud de réseau. Les types de nœuds de réseau peuvent également être reliés à des types de composants matériel qui servent à la réalisation de types de nœuds. La figure suivante en montre un exemple.

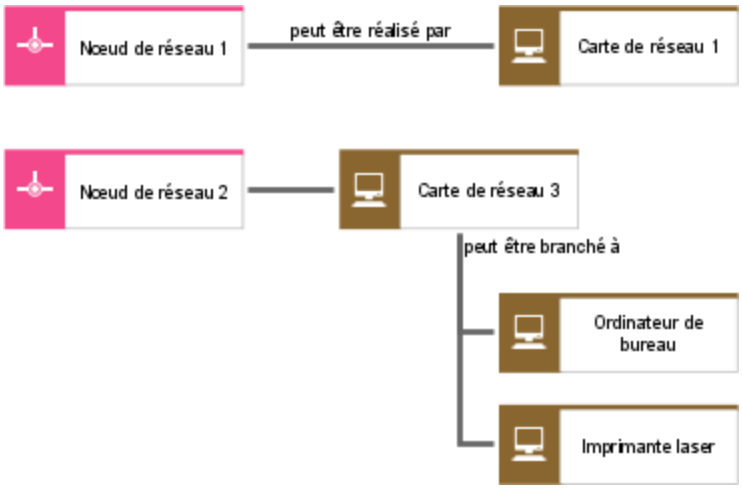


Illustration 63: Topologie de réseau

La relation entre la topologie de réseau et les objets des règles de gestion se fait grâce à deux structures.

D'une part, il est possible de définir, pour chaque type de composant matériel, l'unité organisationnelle ou le poste qui peut en être responsable.

D'autre part, il est possible de définir, pour chaque type de réseau, chaque type de nœud de réseau, chaque type de liaison de réseau et chaque type de composant matériel, le site de l'entreprise où il peut se trouver. Le site est ainsi le lien central entre les règles de gestion et le concept informatique de la vue organisationnelle.

Vous trouverez une liste des objets des types d'objets et de relations disponibles dans un modèle de la topologie du réseau dans l'aide **Référence de la Méthode ARIS**.

3.3.3 Implémentation

3.3.3.1 Diagramme de réseau

La représentation de la réalisation concrète de la topologie de réseau spécifiée dans le concept informatique a lieu dans le diagramme de réseau.

L'objet **Réseau** permet d'intégrer les réseaux existant concrètement dans l'entreprise. Il est possible de spécifier pour chaque réseau les nœuds de réseau et les liaisons de réseau qui le composent.

Chaque réseau, chaque nœud de réseau et chaque liaison de réseau peut être localisée précisément dans l'entreprise. Le site peut être une usine, un bâtiment, un ensemble de bâtiments, un bureau ou un poste de travail individuel.

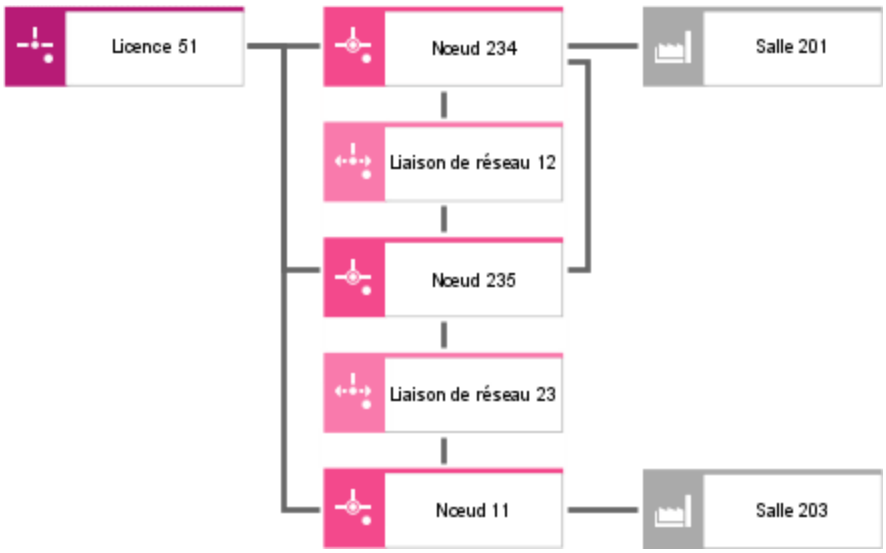


Illustration 64: Diagramme de réseau avec affectation de site

Pour chaque liaison de réseau et pour chaque nœud de réseau il est possible d'intégrer les composants matériels qui ont permis leur réalisation. En outre, l'organisation structurelle de chaque composant matériel peut être représentée. Les composants matériel permettent, d'une part, la réalisation de liaisons de réseau et de nœuds de réseau, et peuvent, d'autre part, être connectés à des nœuds de réseau. Cette relation peut également être représentée dans le diagramme de réseau. Pour chaque objet au niveau exemplaire il est possible de modéliser la relation à l'objet correspondant du niveau de concept informatique. Il est ainsi possible de voir que le réseau de l'usine de la **rue des jardins** est un réseau du type **FDDI ANSI X3.139**.

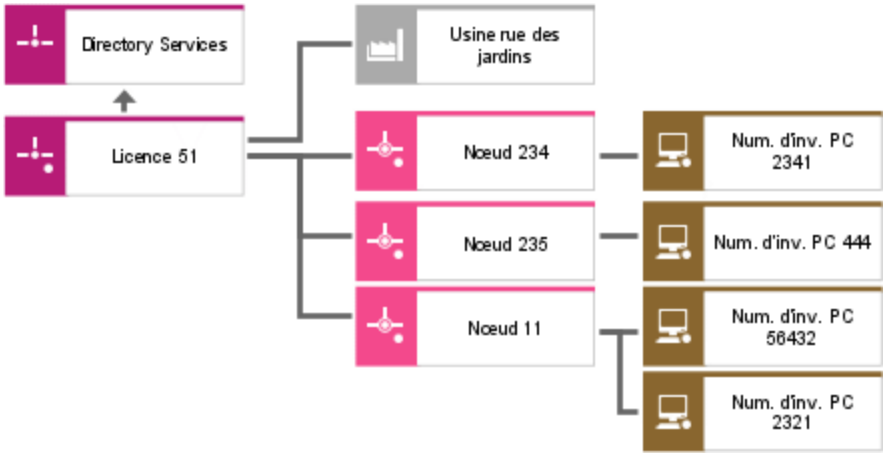


Illustration 65: Diagramme de réseau avec des composants matériels et une affectation de site

Dans les diagrammes de réseau, les affectations de types permettent d'établir les relations au concept informatique et l'affectation de composants de réseau à des sites concrets permet d'établir des relations aux règles de gestion.

Vous trouverez une liste des types d'objets et de relations disponibles dans un diagramme de réseau dans l'aide **Référence de la Méthode ARIS**.

3.3.3.2 Modélisation du flux de matières - Ressources techniques

Pour la représentation du flux de matières dans les modèles de processus (CPE avec flux de matières, DCP avec flux de matières), le système affecte aux fonctions du processus d'entreprise les types de matières comme entrée ou sortie de fonction. Tout comme c'est le cas pour l'affectation d'objets d'information à des fonctions (représentation de la transformation d'informations par des fonctions), cette affectation permet de représenter la transformation de types de matières d'entrée en types de matières de sortie. En outre, il est possible d'enregistrer dans les chaînes de processus les ressources techniques nécessaires à la transformation de la matière. Pour cela, le système distingue les moyens d'exploitation, les installations d'entreposage, les systèmes de transport et les moyens auxiliaires techniques.

Le type de modèle **Ressources techniques** permet de hiérarchiser les ressources, de les assimiler à un type et de les classer. Vous disposez des types d'objets suivants :

MOYENS D'EXPLOITATION

Les moyens d'exploitation sont les exemplaires des différents types de moyens d'exploitation dont une entreprise dispose pour remplir ses tâches. L'attribution de numéros d'inventaire (par ex. le numéro d'une usine) permet souvent d'identifier les moyens d'exploitation.

TYPE DE MOYENS D'EXPLOITATION

Un type de moyens d'exploitation représente l'assimilation à un type de différents moyens d'exploitation basés sur une technologie strictement identique.

CLASSE DE MOYENS D'EXPLOITATION

Il est possible de regrouper des types de moyens d'exploitation similaires dans une classe de moyens d'exploitation. La similarité peut résulter de différents aspects de classification. De ce fait, un type de moyens d'exploitation peut aussi être affecté à plusieurs types de moyens d'exploitation.

INSTALLATIONS D'ENTREPOSAGE

Les installations d'entreposage sont les exemplaires de différents types d'installations d'entreposage, qui sont à la disposition d'une entreprise pour remplir ses fonctions. Les installations d'entreposage sont souvent identifiées à l'aide de numéros d'inventaire.

TYPE D'INSTALLATIONS D'ENTREPOSAGE

Un type d'installations d'entreposage représente l'assimilation à un type de différentes installations d'entreposage, basés sur une technologie strictement identique.

CLASSE D'INSTALLATIONS D'ENTREPOSAGE

Il est possible de regrouper les types d'installations d'entreposage similaires dans une classe d'installations d'entreposage. La similarité peut résulter de différents aspects de classification. De ce fait, un type d'organisation des stocks peut aussi être affecté à plusieurs classes de type d'installations d'entreposage.

MOYENS AUXILIAIRES TECHNIQUES

Un moyen auxiliaire technique est un exemplaire individuel d'un type de moyen auxiliaire technique. En règle générale, il est possible de l'identifier par un numéro d'inventaire.

TYPE DE MOYENS AUXILIAIRES TECHNIQUES

Un type de moyen auxiliaire technique représente l'assimilation de différents moyens auxiliaires techniques, basés sur une technologie strictement identique, à un type.

CLASSE DE MOYENS AUX. TECHN.

Il est possible de regrouper des types de moyens auxiliaires techniques similaires dans une classe de moyens auxiliaires techniques. La similarité peut résulter de différents aspects de classification. De ce fait, un type de moyen auxiliaire technique peut aussi être affecté à plusieurs classes de moyens auxiliaires techniques.

SYSTÈME DE TRANSPORT

Un système de transport est un exemplaire individuel d'un type de système de transport. En règle générale, il peut être identifié par un numéro d'inventaire ou un numéro d'installation.

TYPE DE SYSTÈME DE TRANSPORT

Un type de système de transport représente l'assimilation de différents systèmes de transport, basés sur une technologie strictement identique, à un type.

CLASSE DE SYSTÈME DE TRANSPORT

Il est possible de regrouper des types de systèmes de transport similaires dans une classe de système de transport. La similarité peut résulter de différents aspects de classification. De ce fait, un type de système de transport peut aussi être affecté à plusieurs classes de systèmes de transport.

Les possibilités de hiérarchisation du type de modèle **Ressources techniques** permettent de décrire la structure de systèmes techniques complexes. Ainsi, il est par ex. possible de représenter les composants d'un site de production complexe et leurs relations réciproques.

Outre les possibilités de modélisation décrites ci-dessus, tant les affectations de sites que les responsabilités organisationnelles peuvent être définies pour des ressources techniques. Vous disposez pour cela des types d'objets **Site**, **Unité organisationnelle**, **Groupe**, **Poste**, et

Personne du type de modèle **Organigramme** ; ils peuvent être reliés aux types d'objets **Moyen d'exploitation**, **Installations d'entreposage**, **Moyens auxiliaires techn.** et **Système de transport**.

La figure illustre un exemple pour le type de modèle **Ressources techniques**.

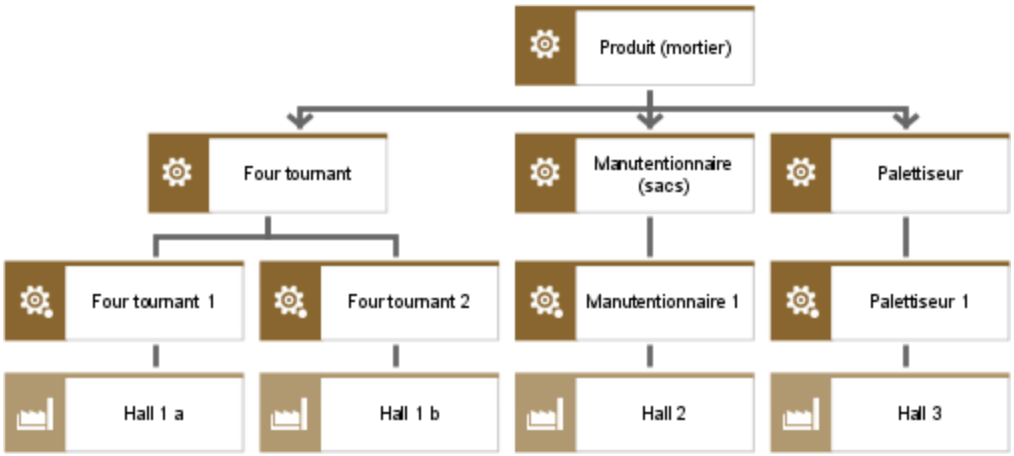


Illustration 66: Exemple d'un modèle "Ressources techniques"

3.4 Vue des processus

3.4.1 Règles de gestion

Les relations entre les objets de la vue des données, de la vue organisationnelle et de la vue des fonctions sont considérées dans la vue de gestion et la vue des processus. Les relations à considérer proviennent des liaisons existant entre les vues.

On observe tout d'abord les relations existant entre deux vues, ensuite des diagrammes sont élaborés pour représenter des relations entre les trois vues.

3.4.1.1 Relier des fonctions avec organisation - CPE

La liaison entre la vue des fonctions et la vue organisationnelle permet d'affecter les fonctions définies dans l'arbre de fonctions aux responsables des tâches (unités organisationnelles) de l'organigramme. Cette affectation définit la responsabilité et la compétence de décision d'une unité organisationnelle pour les fonctions affectées. En observant ces affectations organisationnelles le long d'une chaîne de processus (processus d'entreprise), le degré d'intégration de la fonction, c'est à dire les phases de fonctions d'un processus d'entreprise devant être traitées par une unité organisationnelle, est défini.

La figure suivante illustre un exemple d'affectation d'unités organisationnelles à des fonctions. Ici, l'unité organisationnelle responsable de l'exécution est affectée à la fonction située à gauche. Les relations de subordination et de supériorité des fonctions sont représentées dans la vue des fonctions (arbre de fonctions) et les relations entre les unités organisationnelles sont représentées dans la vue organisationnelle (organigramme). Ainsi, il n'est plus besoin de les définir à ce stade.

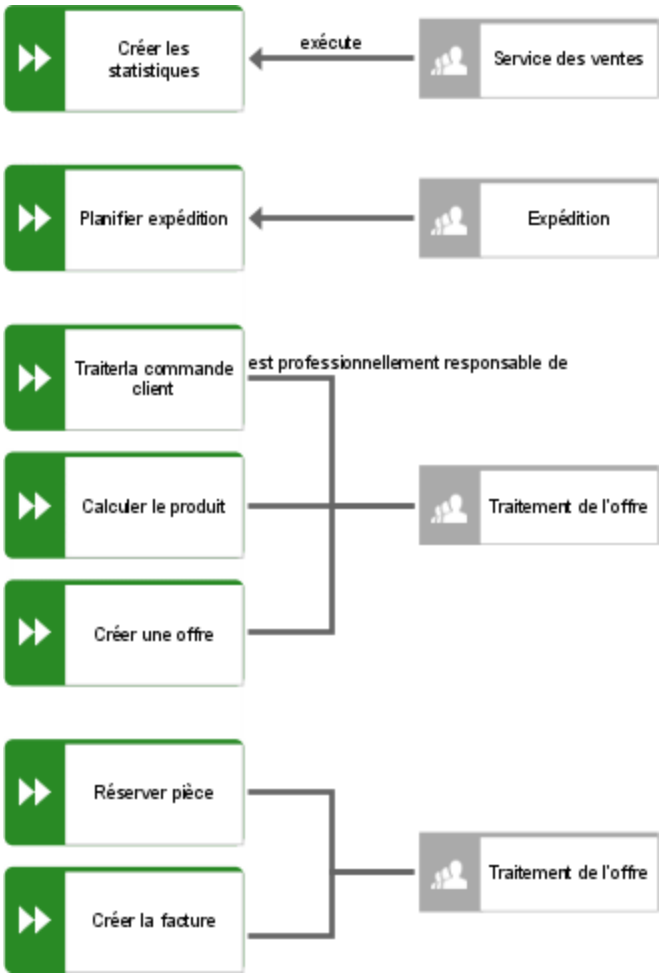


Illustration 67: Affectation des unités organisationnelles à des fonctions

3.4.1.2 Commande événementielle - Chaîne de processus événementielle (CPE)

L'enchaînement de fonctions au sens d'un processus d'entreprise est représenté dans des chaînes de processus. Dans ces chaînes, il est possible d'indiquer les événements de départ et d'arrivée pour chaque fonction. Les événements déclenchent les fonctions et sont le résultat de ces dernières.

Un événement est la manifestation de l'état d'un objet d'information pertinent pour la gestion d'entreprise ; il commande ou influence la suite du déroulement du processus d'entreprise.

Les événements déclenchent les fonctions et sont le résultat de ces dernières. Contrairement à une fonction qui se produit dans le temps, un événement est ponctuel.

Le changement d'état d'un objet d'information peut être lié à la première manifestation de cet objet (p. ex. : **la demande client est arrivée**) ou à un changement de l'état fixé dans une valeur d'attribut (p. ex. : **l'offre est refusée**). Comme les objets d'information et les attributs sont décrits dans la vue des données d'ARIS, la représentation événementielle des chaînes de processus constitue un lien entre la vue des données et la vue des fonctions ; elle est donc rattachée à la vue de gestion d'ARIS.

Le symbole graphique de l'événement est un hexagone. La désignation d'un événement doit comprendre à la fois l'objet d'information (**commande**) et le changement d'état de cet objet (**est arrivée**). La figure suivante montre des événements.

Illustration 68: Événements (représentation graphique)

Les événements déclenchent les fonctions et sont le résultat de ces dernières. L'enchaînement de ces changements événement-fonction donne naissance à des chaînes de processus événementielles. Une chaîne de processus événementielle (CPE) montre le déroulement chronologique d'un processus d'entreprise.

La figure suivante illustre un exemple de CPE. Comme les événements définissent l'état ou la condition qui déclenche une fonction ainsi que l'état qui en marque l'achèvement, les nœuds de départ et d'arrivée d'une telle CPE sont toujours des événements. Un événement peut déclencher plusieurs fonctions simultanément et, inversement, une fonction peut engendrer plusieurs événements. Pour pouvoir représenter ces ramifications et ces boucles de traitement dans une CPE, le système utilise une règle sous forme de cercle. Celle-ci n'est cependant pas une simple relation graphique ; elle définit également les connexions logiques entre les objets qu'elle relie.

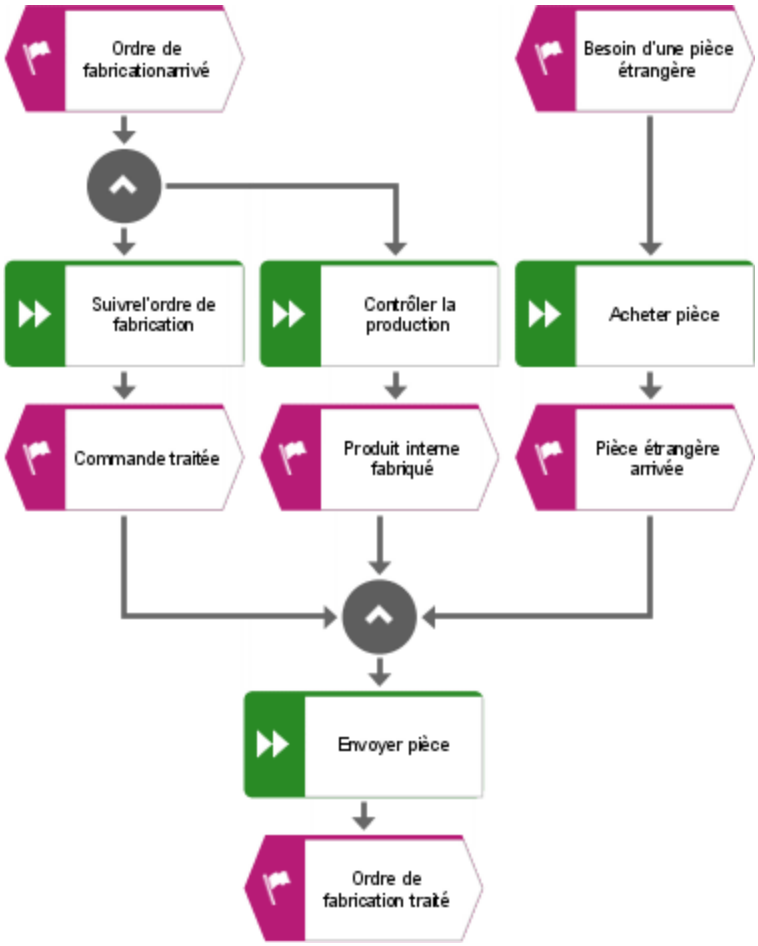


Illustration 69: Exemple de CPE

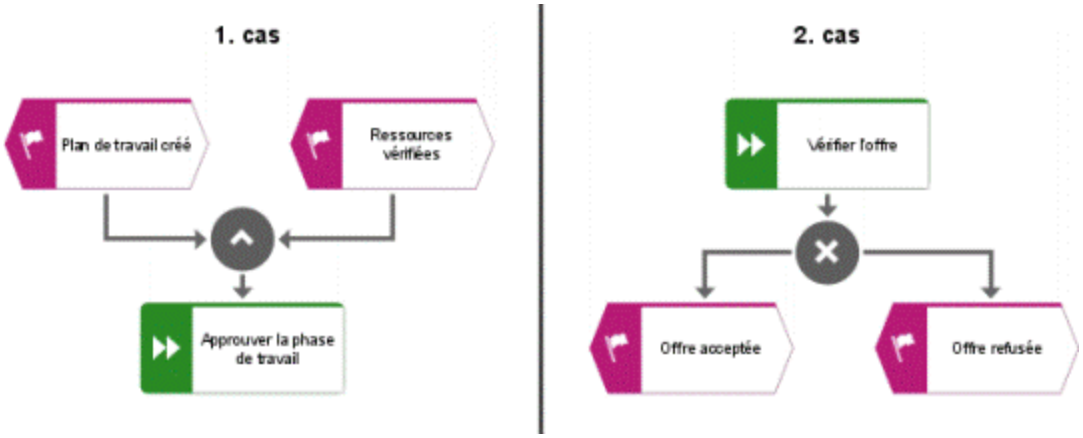


Illustration 70: Exemples de règles

Dans le premier cas de la figure, il existe une connexion ET par une règle ET entre les événements de départ. Cela implique que le processus **activer phase de fabrication** ne peut être lancé que lorsqu'il existe une gamme opératoire et lorsque les ressources nécessaires ont été vérifiées. Pour que le processus soit lancé, il faut que les deux événements aient déjà eu lieu. Le deuxième cas représente une connexion OU exclusive (OU exclusif) à l'aide d'une règle XOR. Le résultat de la fonction **Vérifier l'offre du fournisseur** peut être l'acceptation ou le refus de l'offre. Les deux cas de figure ne peuvent toutefois pas se présenter en même temps. Outre ces deux cas et la connexion au sens d'un 'OU exclusif', il peut aussi exister des relations plus complexes.

De manière générale, il est ainsi possible de différencier deux types de connexions :

1. Connexions événementielles et
2. Connexions de fonctions.

Un aperçu des connexions événementielles et des connexions de fonctions est présenté dans la figure suivante (cf. Hoffmann, Kirsch, Scheer, Modellierung mit Ereignisgesteuerten Prozessketten, 1993. p.13).

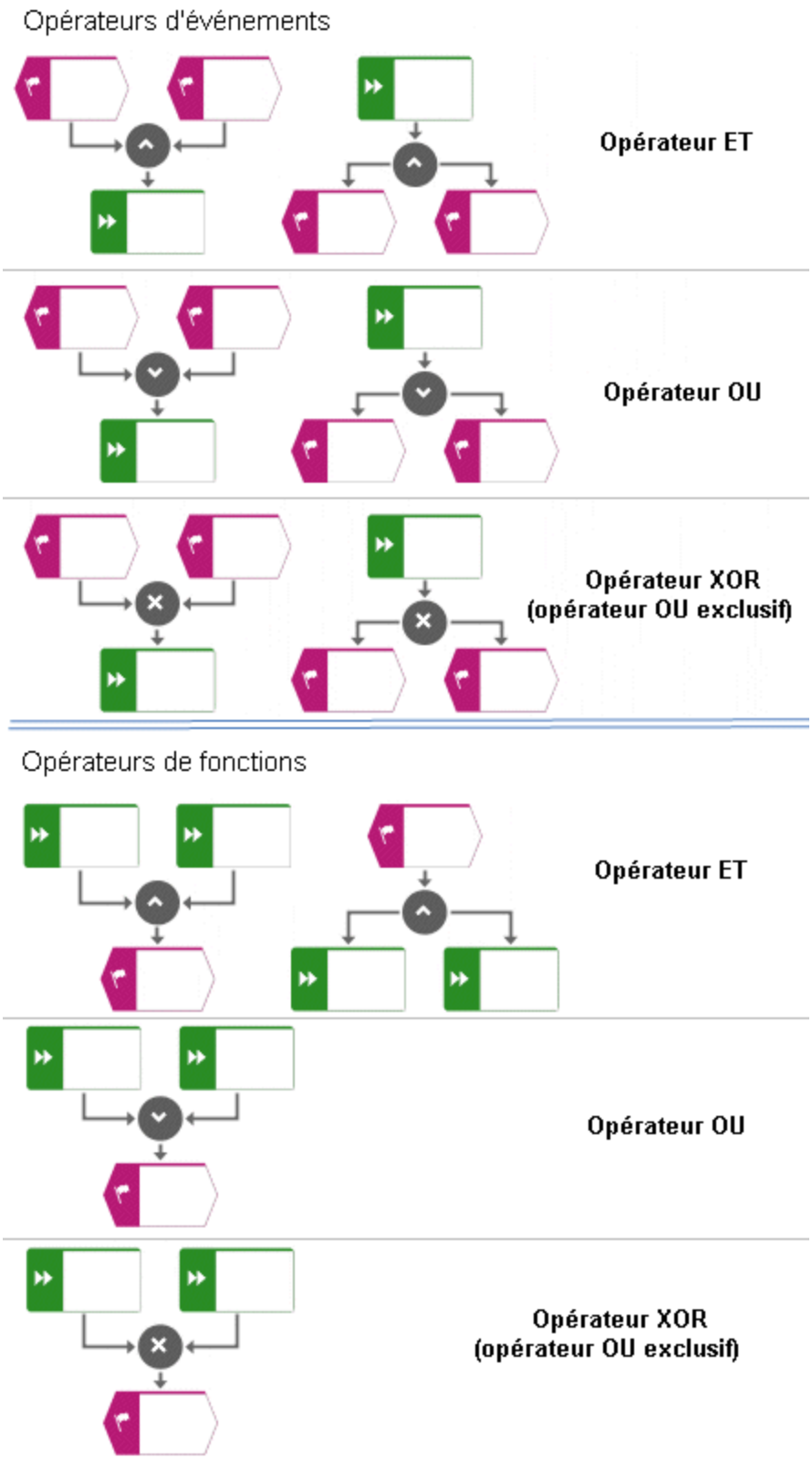


Illustration 71: Opérateurs de connexion (règles)

Il est surtout important d'observer les restrictions concernant les connexions de fonctions. Comme les événements ne peuvent pas prendre de décision (seules les fonctions sont aptes à le faire), la connexion d'un événement déclencheur dotée de connecteurs OU et OU exclusif n'est pas autorisée !

Vous trouverez ci-dessous les connexions possibles illustrées à l'aide d'exemples.

LA CONNEXION D'ÉVÉNEMENTS DÉCLENCHEURS

RÈGLE ET

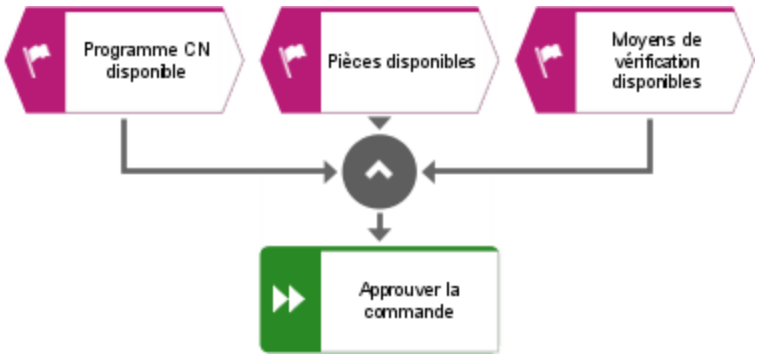


Illustration 72: Connexion ET d'événements déclencheurs

La fonction ne peut être lancée que lorsque tous les événements se sont produits.

RÈGLE OU

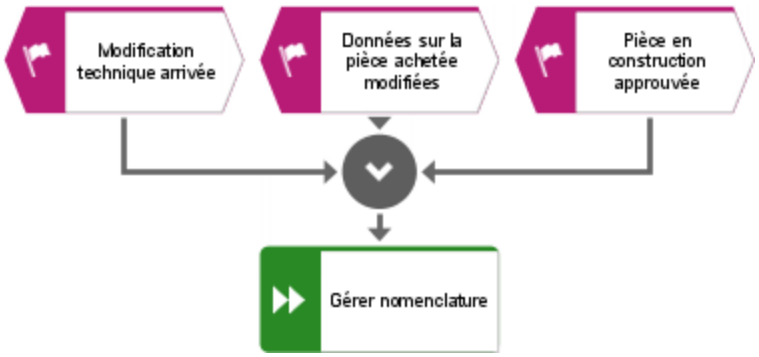


Illustration 73: Connexion OU d'événements déclencheurs

La fonction est exécutée lorsqu'au moins un des événements s'est produit.

OU EXCLUSIF (RÈGLE XOR)

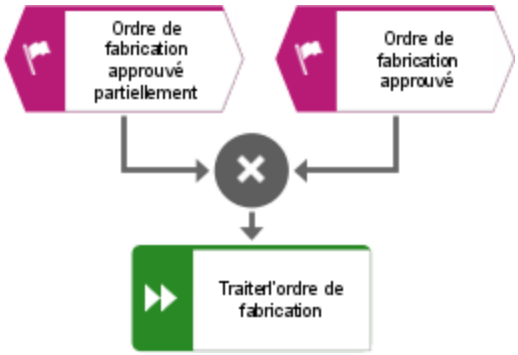


Illustration 74: Connexion OU exclusif d'événements déclencheurs

La fonction est lancée lorsqu'un et seulement un des événements s'est produit.

LA CONNEXION D'ÉVÉNEMENTS GÉNÉRÉS

RÈGLE ET

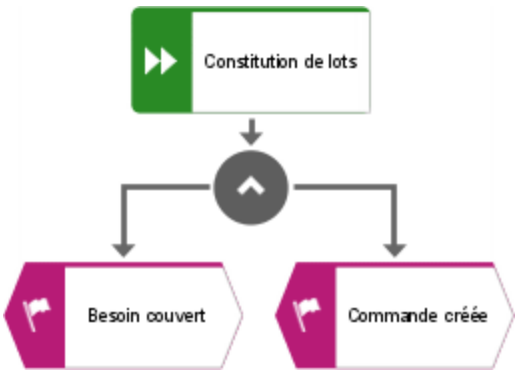


Illustration 75: Connexion ET d'événements générés

La fonction entraîne l'intervention de tous les événements.

RÈGLE OU

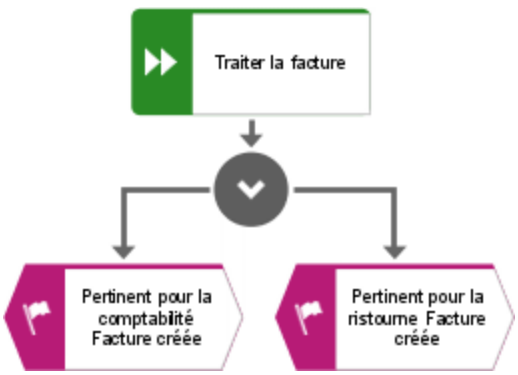


Illustration 76: Connexion OU d'événements générés

L'exécution de la fonction entraîne l'intervention d'au moins un des événements.

OU EXCLUSIF (RÈGLE XOR)

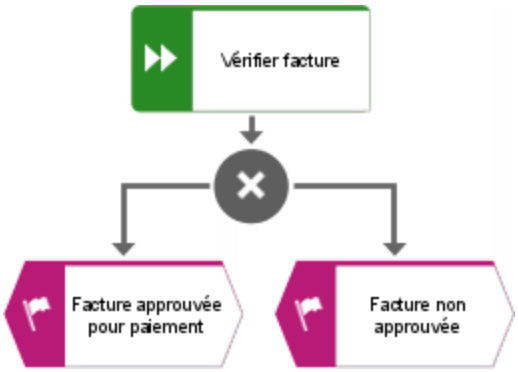


Illustration 77: Connexion OU exclusif d'événements générés

L'exécution de la fonction entraîne l'intervention d'au plus un des événements.

LA CONNEXION DE FONCTIONS AVEC DES ÉVÉNEMENTS GÉNÉRÉS

RÈGLE ET

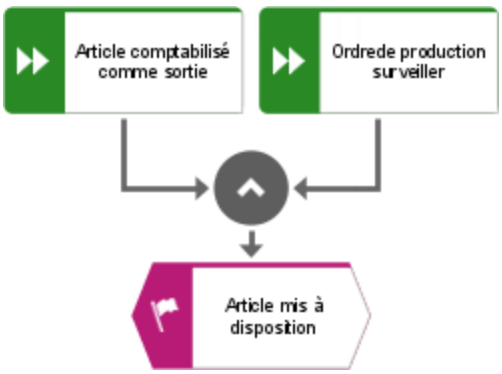


Illustration 78: Connexion ET de fonctions avec des événements générés

L'événement se produit lorsque toutes les fonctions ont été exécutées.

RÈGLE OU

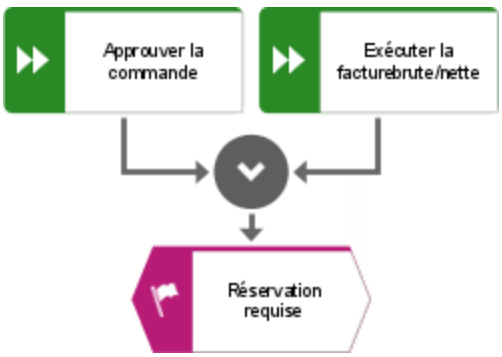


Illustration 79: Connexion OU de fonctions avec des événements générés

L'événement se produit lorsqu'au moins une des fonctions a été exécutée.

OU EXCLUSIF (RÈGLE XOR)

Illustration 80: Connexion OU de fonctions avec des événements générés

L'événement se produit lorsqu'une seule fonction au plus a été exécutée.

LA CONNEXION DE FONCTIONS AVEC DES ÉVÉNEMENTS DÉCLENCHEURS

RÈGLE ET

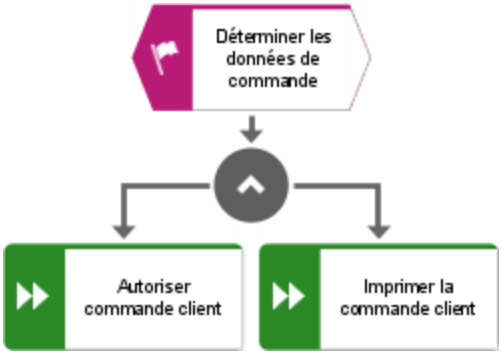


Illustration 81: Connexion ET de fonctions avec des événements déclencheurs

Toutes les fonctions sont déclenchées par l'événement.

RÈGLE OU

Les événements n'ont aucune compétence décisionnelle ! Cette connexion n'est pas possible !

OU EXCLUSIF (RÈGLE XOR)

Les événements n'ont aucune compétence décisionnelle ! Cette connexion n'est pas possible !

En plus de la représentation sous forme de chaînes de processus événementielles, ces ramifications peuvent être également représentées sous forme de tables dans les colonnes événement et fonction d'un diagramme de chaînes de processus. La classification séquentielle des fonctions dans un diagramme de chaînes de processus rend la représentation des ramifications et des boucles de traitement très peu claire.

3.4.1.3 Diagramme de rattachement de fonctions (E/S)

En plus de la représentation de la gestion des événements illustrée au chapitre **Commande événementielle - Chaîne de processus événementielle (CPE)** (page 67), la transformation de données d'entrée en données de sortie et la représentation des flux de données entre les fonctions représentent également une liaison entre la vue de données et la vue des fonctions du concept ARIS. La transformation de données d'entrée en données de sortie peut être représentée dans un diagramme nommé Diagramme de rattachement de fonctions (E/S). La figure ci-dessous illustre un exemple de diagramme de rattachement de fonctions (E/S). La fonction **Fixer les délais de livraison** a les données d'entrée **Données pièces**, **Données stock**, **Données de nomenclature** et **Données expédition**. Les **données de la demande** sont aussi bien des données d'entrée que des données de sortie. Les fonctions de la vue des fonctions et les objets d'information de la vue des données font donc partie d'un diagramme de rattachement de fonctions (E/S). Les flèches définissent si un objet d'information fait office uniquement de date d'entrée, ou uniquement de date de sortie ou de date d'entrée/de sortie. Il est en outre possible d'ajouter des spécifications plus précises, qui indiquent, p. ex., si un objet d'information est créé ou supprimé par la fonction. En fonction du degré de détail, il peut s'agir des objets d'information suivants : cluster/modèle de données, types de relations/ d'entités ou attributs de la vue des données.

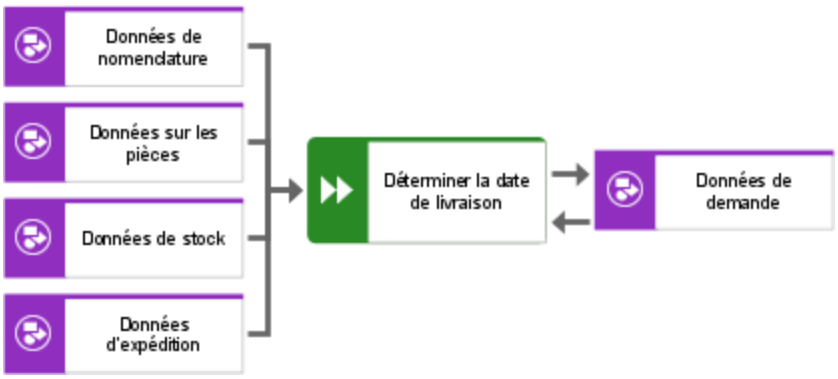


Illustration 82: Exemple de diagramme de rattachement de fonctions (E/S)

L'exemple présenté ici souligne l'objectif réel du diagramme de rattachement de fonctions (E/S), à savoir la représentation des données d'entrée et de sortie d'une fonction.

En plus de l'enregistrement des données d'entrée et de sortie d'une fonction, tous les autres objets pouvant être affectés aux fonctions dans une CPE sont à votre disposition, sauf les événements. Lors de la modélisation de chaînes de processus en diagrammes CPE, l'utilisateur a ainsi la possibilité de n'intégrer que les événements et les fonctions, et de créer derrière chaque fonction un diagramme de rattachement de fonctions (E/S) avec les autres relations de la fonction. Cela permet une représentation bien plus claire des processus d'entreprise et explique également l'utilisation d'une nouvelle désignation pour ce type de

modèle. La figure ci-dessous illustre un exemple d'une telle représentation dans un diagramme de rattachement de fonctions.

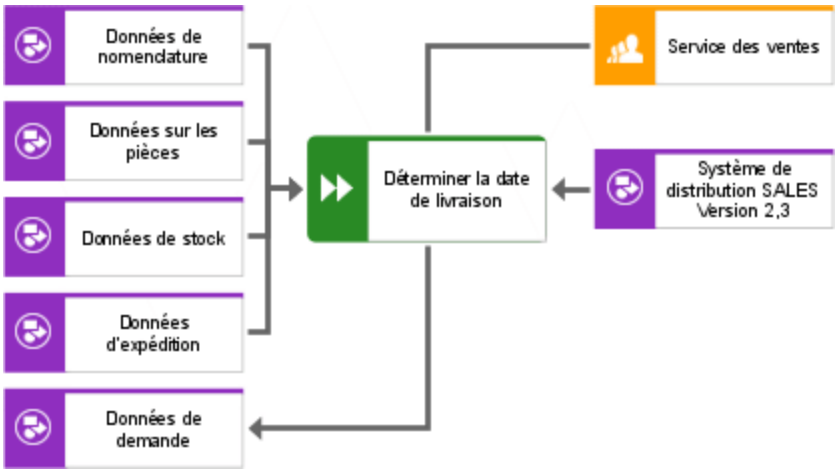


Illustration 83: Représentation étendue du diagramme de rattachement de fonctions

En plus de la représentation séparée de la transformation de données en diagrammes de rattachement des fonctions (E/S), ces informations peuvent également être intégrées dans une CPE. Un exemple est illustré dans la figure ci-dessous. Les liens entre les fonctions et les objets d'information ont la même signification que ceux du diagramme de rattachement de fonctions (E/S) ; leur intégration dans une chaîne de processus fortement ramifiée peut cependant donner des représentations peu claires.

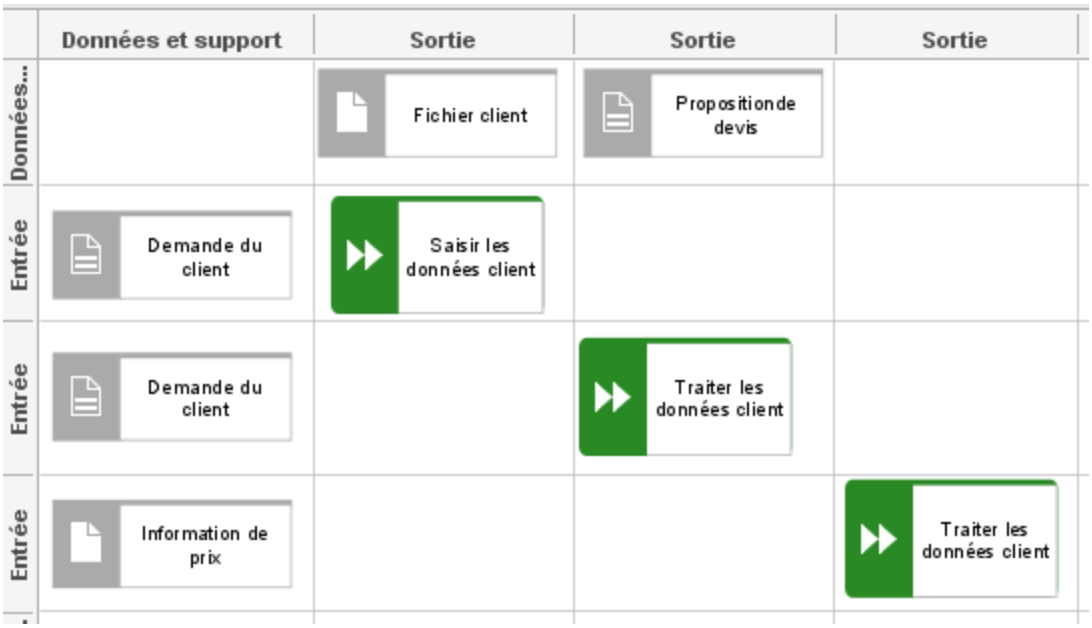


Illustration 84: CPE avec données d'entrée/de sortie

3.4.1.4 Diagramme d'événements

Les événements définissent l'occurrence de modification d'état d'objets d'information. Chaque événement fait référence à des objets d'information précis du modèle de données et définit l'état des objets d'information correspondants à un moment donné.

En premier lieu, les événements sont spécifiés grossièrement de manière descendante (par exemple, **Commande est traitée**). La prochaine étape de spécification de la modélisation du processus permet de définir les événements plus détaillés qui selon des combinaisons précises font que cet événement ait eu lieu au niveau précédent. Ainsi p. ex., l'apparition des événements **Faisabilité vérifiée**, **En-tête de commande saisi** et **Positions de commande saisies** dans la somme peut définir l'état **Commande est traitée**.

Ces relations entre les événements à des niveaux de modélisation différents peuvent être représentées grâce au diagramme d'événements. Un diagramme d'événements peut, à cet effet, être placé derrière un événement au premier niveau (hiérarchisation !!) ; les événements de ce diagramme et leurs relations (aussi celles passant par des connexions de règles) pourront être représentés au niveau plus détaillé. En outre, des objets d'informations du modèle de données peuvent être représentés dans ce type de modèle et liés aux événements. Ceci permet de définir quel événement définit les modifications de quel objet d'information.

Un exemple est illustré dans la figure suivante.

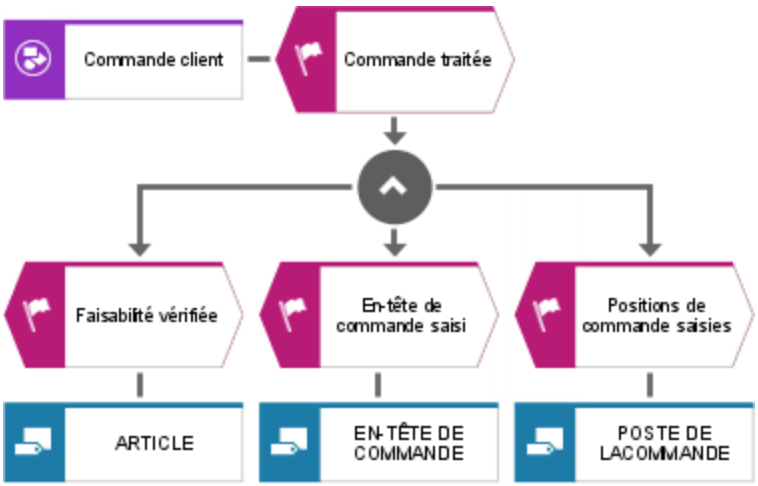


Illustration 85: Exemple de diagramme d'événements

3.4.1.5 Diagramme de chaînes de plus-value

Le diagramme de chaînes de plus-value permet en premier lieu de spécifier les fonctions de l'entreprise qui sont directement impliquées dans la plus-value de celle-ci. Ces fonctions peuvent être reliées entre elles sous forme d'une suite de fonctions et forment ainsi une chaîne de plus-value. La figure illustre un exemple de chaîne de plus-value.

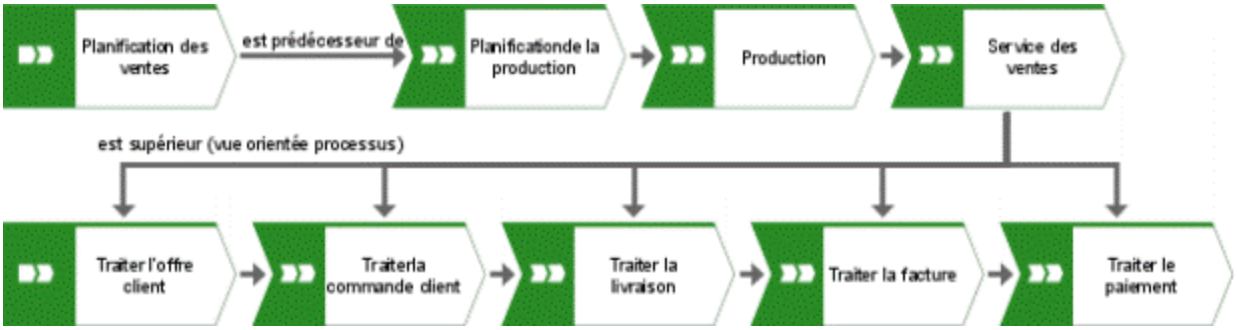


Illustration 86: Chaîne de plus-value

Les fonctions peuvent être hiérarchisées dans le diagramme de chaînes de plus-value, de façon similaire à un arbre de fonctions. C'est toujours la hiérarchie orientée processus qui est représentée.

Outre la hiérarchisation de fonctions, il est possible de représenter, dans un diagramme de chaînes de plus-value, les relations entre les fonctions et les unités organisationnelles et les objets d'information. Lors de l'affectation d'unités organisationnelles à des fonctions, le système distingue, tout comme dans les chaînes de processus, la responsabilité professionnelle, la responsabilité informatique et l'exécution effective d'une fonction.

Vous trouverez une liste des autres relations disponibles dans le diagramme de chaînes de plus-value dans l'aide **Référence de la Méthode ARIS**.

3.4.1.6 Modélisation orientée objet

UML est disponible pour la modélisation dans l'environnement UML. Toutes les informations sur la méthode concernant les diagrammes UML et les éléments UML sont directement proposées dans l'interface d'ARIS UML Designer.

3.4.1.7 Matrice de choix des processus

Différents scénarios de processus sont représentés dans la matrice de choix des processus en affectant des processus principaux à des scénarios individuels.

L'utilisateur peut ensuite déterminer quelles fonctions des processus de scénarios interviennent dans l'entreprise. Une matrice doit toujours contenir, en tant que processus,

toutes les fonctions principales (fonctions de scénario) d'une application ou d'un modèle de référence d'un secteur.

Pour modéliser une matrice de choix des processus, vous pouvez utiliser les types de symboles suivants :

- Scénario
- Processus
- Processus principal

Un scénario représente un processus de scénario dans la matrice de choix regroupant différents processus principaux.

Le processus représente des fonctions du processus de scénario qui seront décrites plus précisément par des modèles de processus dans des modèles de référence.

Le processus principal représente les fonctions principales dans les arbres de fonctions auxquels les processus (fonctions des processus de scénario) sont affectés.

La figure suivante illustre un exemple de matrice de choix des processus.

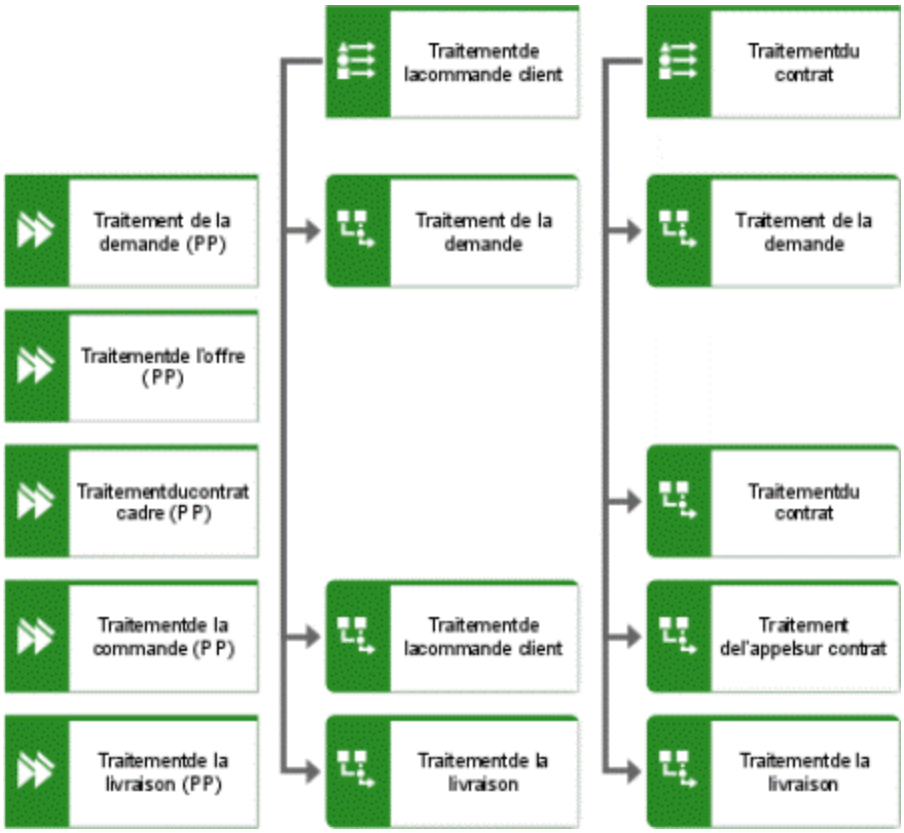


Illustration 87: Matrice de choix des processus (extrait du modèle de référence R/3 de SAP AG)

3.4.1.8 Modélisation du flux de matières

En dehors de la représentation du flux d'informations, les CPE peuvent également être utilisées pour la représentation de transformation de matières. Dans ARIS, un type de modèle propre vous est proposé (CPE avec flux de matières, représentant un élargissement du type de modèle **CPE**), permettant la représentation du flux des matières au sein des processus d'entreprise.

3.4.1.9 CPE (flux de matières)

Outre les types d'objets de la CPE, la CPE avec flux de matières propose également les types d'objets suivants :

- Type de matières
- Type d'emballage
- Type de moyens d'exploitation
- Moyens d'exploitation
- Type de moyens aux. techn.
- Moyens auxiliaires techniques
- Type d'installations d'entreposage
- Installations d'entreposage
- Type de système de transport
- Système de transport

Le type d'objet **Type de matières** peut être relié au type d'objet **Fonction** par une liaison entrante ou une liaison sortante. Dans le cas d'une liaison entrante, le système définit quelles matières sont requises comme entrée pour une fonction. La sélection du type de liaison correspondant permet de définir si la matière est utilisée entièrement, partiellement ou n'est pas utilisée par la fonction. Une liaison sortante spécifie les types de matières créés par la fonction.

La transformation de la matière nécessite des ressources techniques. Celles-ci peuvent également être reliées, dans les chaînes de processus, au type d'objet **Fonction**. En plus du type de liaison **requiert**, le type de liaison **requiert alternativement** vous est proposé afin de spécifier d'éventuelles ressources alternatives.

Si des matières sont empaquetées au sein d'une fonction, cela requiert certains types d'emballages. Pour spécifier les types d'emballages correspondants, il est possible de modéliser une relation entre la fonction et le type d'emballage requis.

La figure suivante représente une CPE (flux de matières) et les types de ressources techniques s'y rapportant, ainsi que les types d'emballages.

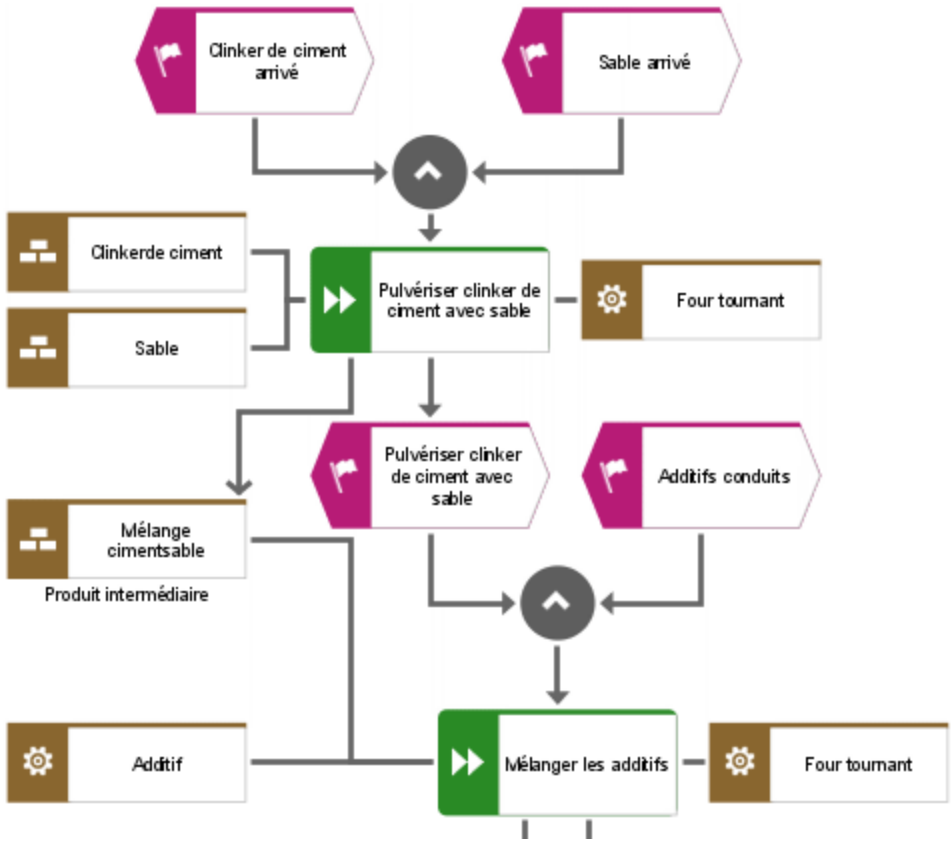


Illustration 88: Extrait d'une CPE (flux de matières)

3.4.1.10 CPE (sous forme de colonnes/lignes)

La description suivante est valable de la même façon pour la CPE (sous forme de lignes).

Pour le type de modèle **CPE (sous forme de colonnes)**, les explications sur la CPE sont valables pour l'essentiel, à l'exception près que dans ce modèle, tous les symboles sont répartis dans les différentes colonnes. Grâce à cette représentation, la CPE gagne en netteté. Les éléments d'application et d'organisation sont placés dans l'en-tête du diagramme. Tous les autres symboles sont placés dans la deuxième ligne des colonnes.

La génération automatique de relations non visibles (implicites) est une caractéristique propre à tous les modèles de corridor, donc à tous les modèles modélisés en colonnes et/ou en lignes. Par exemple, les relations implicites "assiste" sont créées automatiquement dans les colonnes définies par défaut de la CPE (sous forme de colonnes) lors de la modélisation d'applications et de fonctions. Des relations implicites 'exécute' sont créées entres les fonctions et les éléments organisationnels. De plus, l'utilisateur a la possibilité d'ajouter d'autres colonnes à la suite, qui seront intitulées selon les relations implicites.

- Contribue à

- Décide de
- Est le responsable informatique de
- Est professionnellement responsable de
- Doit être informé en cas d'annulation
- Doit informer du résultat de
- Doit être informé de
- Accepte
- Joue un rôle consultatif

La figure suivante illustre l'exemple d'une CPE (sous forme de colonnes).

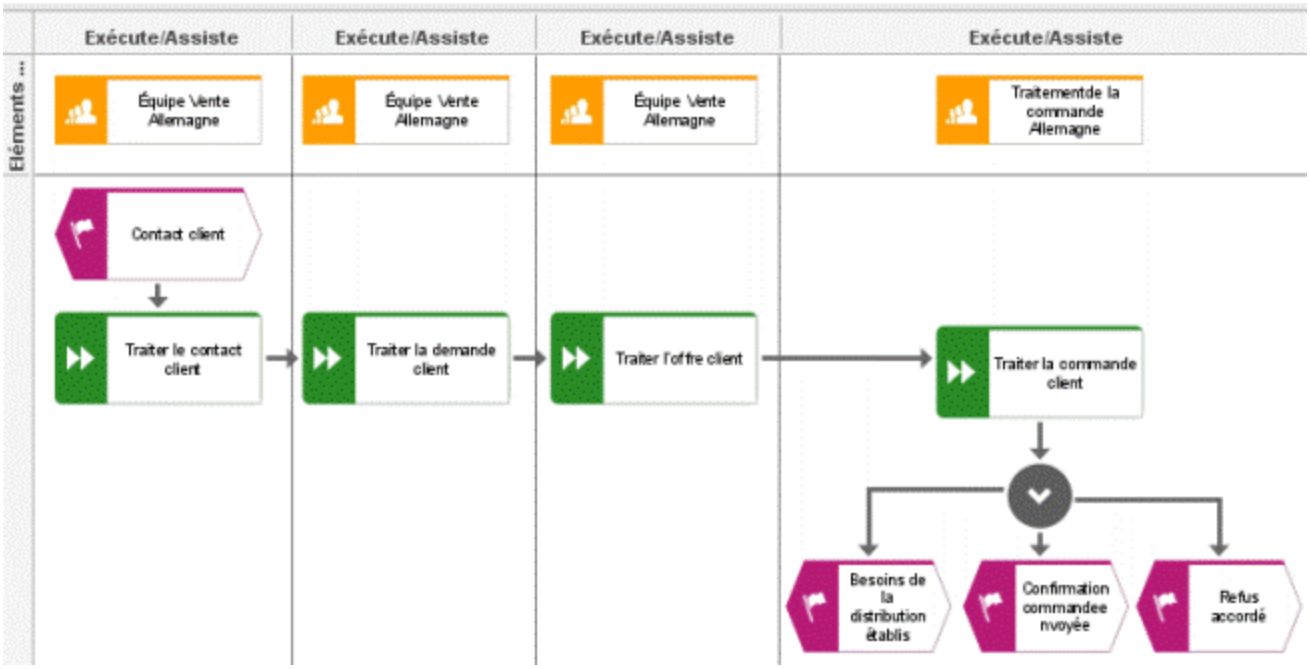


Illustration 89: CPE (sous forme de colonnes)

La seule différence entre la CPE (sous forme de colonnes) et la CPE (sous forme de lignes) est la direction différente de la modélisation. Dans la CPE (sous forme de colonnes), la modélisation s'effectue de haut en bas, alors que dans la CPE (sous forme de lignes), elle s'effectue de gauche à droite.

3.4.1.11 Diagramme de contrôles d'affaires

Dans un diagramme de contrôles d'affaires, les risques éventuels et les possibilités de contrôle de risque sont illustrés pour un processus ou plus exactement pour une fonction.

Un risque (Risk) représente un danger potentiel pour un processus. Il peut compromettre l'objectif du processus désiré.

Le contrôle de risque, ou du moins sa réduction (Control) est l'un des moyens existant pour éliminer ou réduire les risques.

La résolution de risque (Solution) n'est autre que la réalisation de la réduction du risque (Control) pour un risque (Risk) précis.

La mise en page du business controls diagram correspond à une matrice ou à une table. Les risques de processus possibles s'inscrivent sur l'axe des abscisses et les possibilités de contrôle de risque s'inscrivent sur l'axe des ordonnées. Les résolutions de risques sont alors ajoutées en tant que connexions entre un risque et une réduction de risque. En outre, peuvent être ajoutées au modèle d'autres unités organisationnelles (telles que les exigences utilisateurs) ainsi que des documents soutenant également la réalisation du contrôle de risque d'un risque précis.

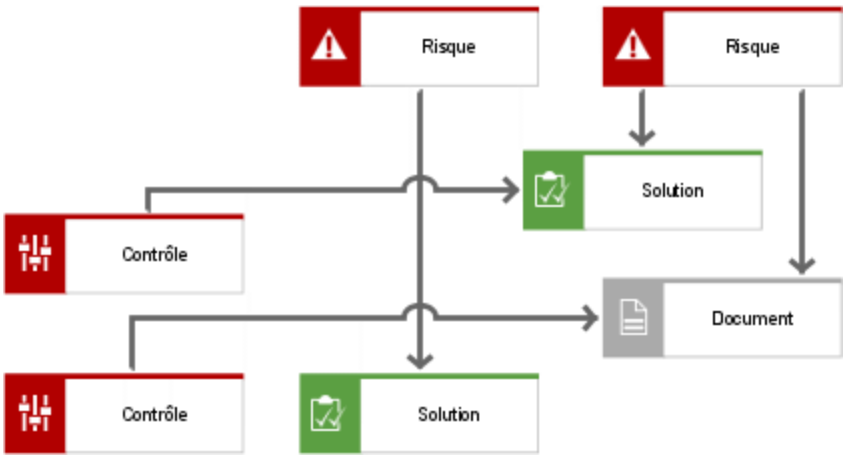


Illustration 90: Exemple de diagramme de contrôles d'affaires

En général, ce type de modèle est utilisé pour la description de processus SAP® par défaut. Ici, le modèle montre les risques et les contrôles du risque qu'une solution SAP® propose quant au processus observé.

3.4.1.12 eBusiness scenario diagram

L'importance d'un déroulement sans heurts du processus d'entreprise au sein de l'entreprise ne cesse de croître. Ici, l'accent est mis sur le déroulement des processus déterminés au niveau des interfaces des entreprises d'une part, et l'interface entre l'entreprise et ses clients d'autre part. Les contacts devraient s'exécuter de manière claire, rapide, uniforme et directe.

Il devient également de plus en plus important de trouver des partenaires commerciaux du point de vue de l'entreprise et des fournisseurs du point de vue du consommateur. La structuration optimale de ces processus signifie un avantage en matière de compétition. La plate-forme appropriée pour assister ces relations bilatérales est l'Internet. Puisque les processus au sein de l'environnement décrit précédemment sont multiples, il est nécessaire de définir ce que vous entendez par **e-business**.

Le terme E-business désigne tous les processus assistés par ordinateur entre deux sujets économiques et la tentative d'obtenir une nouvelle plus-value grâce à de nouveaux supports.

Cela peut tout simplement être l'achat d'un article sur Internet, un projet très complexe de deux entreprises ou la création d'un site Web pour la présentation d'une entreprise.

Les relations Entreprise - Entreprise sont appelées Business-To-Business (B2B), les relations Client - Entreprise sont appelées Business-To-Consumer (B2C).

Le 'e-Business scenario diagram' a été développé pour assister l'e-Business.

Le fait de pouvoir observer globalement une chaîne de plus-value, c'est-à-dire du client final à toutes les entreprises participant au processus, doit permettre de développer un potentiel de développement. L'objectif est par exemple l'amélioration de la chaîne d'approvisionnement, la réduction des coûts d'approvisionnement et de distribution ou le perfectionnement de la structure des systèmes d'information. Les contenus représentés par les objectifs sont rendus modélisables par cette méthode.

Les objets économiques sont insérés dans la première ligne du diagramme et nommés **Business participants**. Les entreprises participantes peuvent être associées en utilisant un organigramme. Ici, l'accent est mis sur les processus uniques participant au processus global des sujets économiques et sur les interfaces qui les relient entre eux. Un processus singulier (Business process) est un processus d'entreprise participant de manière décisive à des coopérations en allant au-delà de l'entreprise qui peut être associée au modèle de processus. Le Business process est assisté par des applications (Business components), p. ex. par le système R/3.

Les rôles des collaborateurs participant au processus peuvent également être définis. Ils sont désignés par **Employee role** dans le modèle.

Les interfaces sont caractérisées par la transmission des informations spécifiques au processus. Elles sont regroupées dans Business documents et peuvent adopter la forme d'un document XML ou HTML. Le Business document peut aussi être associé comme modèle de

données. Au lieu de cet objet, vous pouvez utiliser les objets **Money transaction** (pour la représentation du flux monétaire), **Goods shipment** (pour la représentation du flux des marchandises), **E-mail**, **Internet**, **Intranet**, **Extranet** et **Téléphone portable** (pour la spécification de l'aspect technologique de la transmission de données).

Dans le modèle, tous les déroulements concernant une entreprise sont modélisés dans la ligne située sous le Business participant mais dans la même colonne.

Il en résulte que les marges des colonnes forment les interfaces abstraites. Celles-ci sont très importantes car elles contiennent le potentiel de perfectionnement principal et sont donc extrêmement intéressantes à des fins de modélisation.

Explication de terme : dans l'exemple de modèle suivant, OEM signifie **Original Equipment Manufacturer** et MRP, **Material Ressource Planning Controller**.

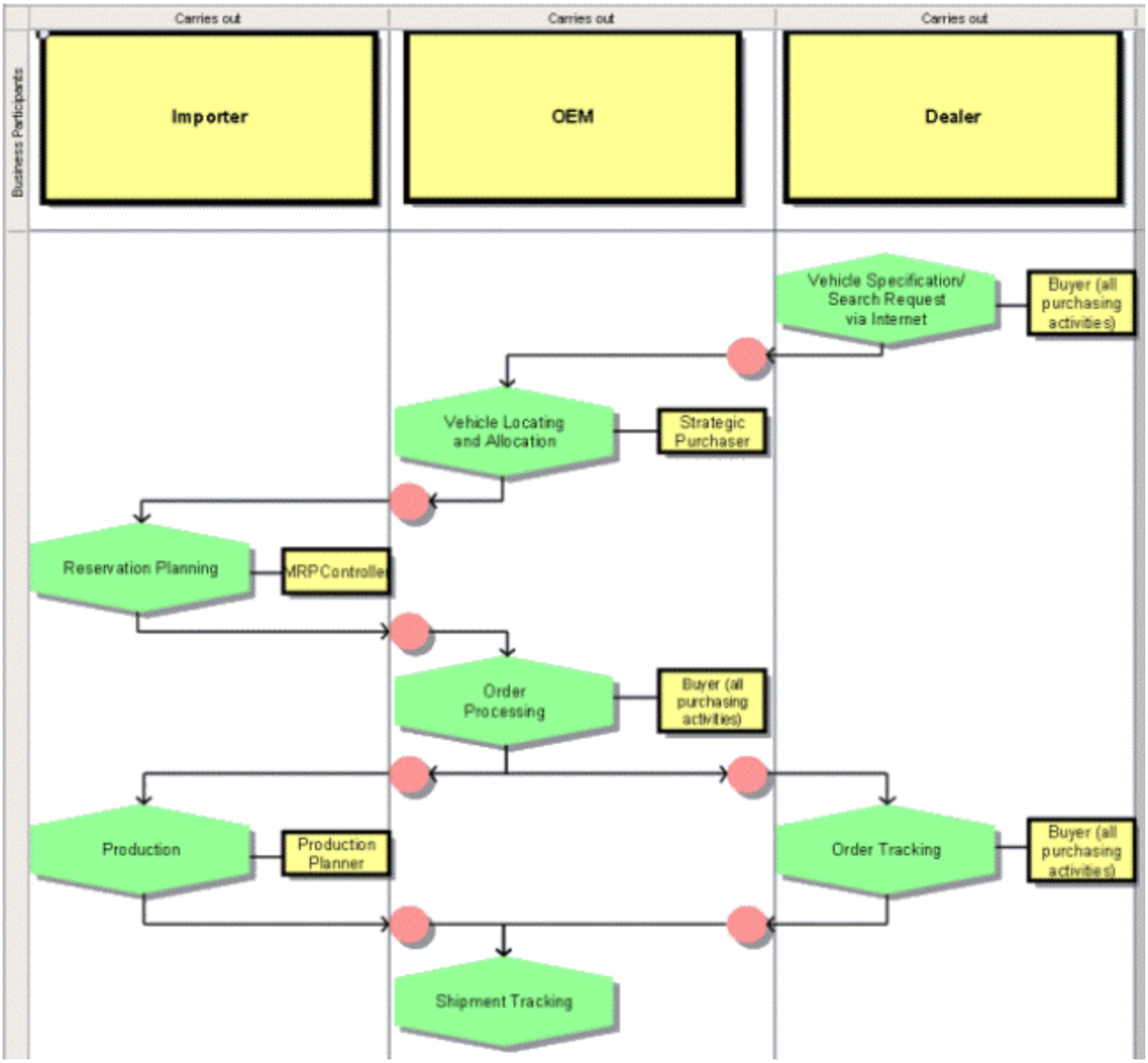


Illustration 91: Exemple d'un eBusiness scenario diagram : secteur automobile

Dans l'exemple de modèle, la coopération d'un fabricant, d'un importateur et d'un commerçant est représentée. Dans la structure globale, chacun a ses propres processus spécifiques qui échangent des informations avec les processus d'autres partenaires commerciaux via des Business documents au niveau des interfaces. Les personnes participant aux processus d'entreprise sont saisies et affectées avec leurs rôles.

3.4.1.13 Modèle de composition

Le modèle de composition sert en général à la hiérarchisation ou à la systématisation d'un fait (spécialisation ou généralisation de faits).

Un élément de composition représente un fait (dans le sens de la systématisation souhaitée).

Des modèles que l'on affecte aux faits peuvent être associés aux différents éléments de composition de la hiérarchie de faits.

Le domaine d'application du modèle de composition le plus utilisé est la gestion de qualité, plus spécialement la certification. Là, par l'intermédiaire d'un modèle de composition, une norme est subdivisée en différents composants, et les modèles contribuant à l'exécution des critères de qualité sont associés à tous les éléments de composition.

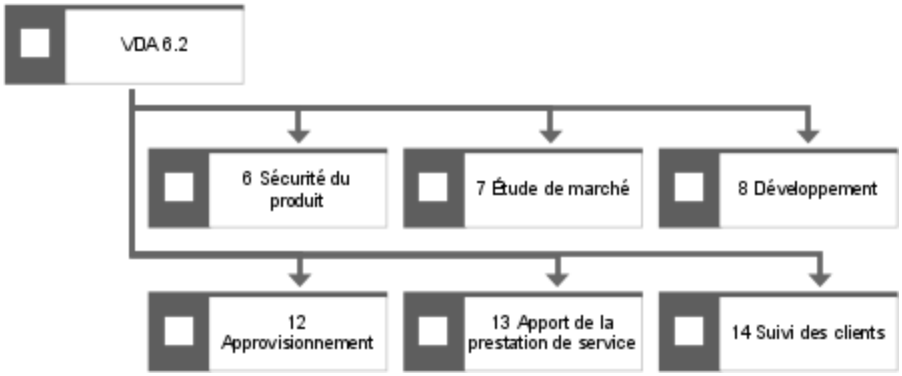


Illustration 92: Exemple d'un modèle de composition (extrait de la norme VDA 6.2)

Ces faits peuvent être aisément évalués grâce à un rapport ou peuvent être utilisés à des fins de documentation.

3.4.1.14 Diagramme de rôles

Le diagramme de rôles sert généralement à la description plus précise des processus. Les unités organisationnelles qui participent aux processus et à leurs rôles sont primordiales. Les propriétés suivantes permettent de caractériser les objets et leurs relations :

Un rôle participe à des processus en tenant compte des autorisations. Non seulement la possibilité d'exécution est importante mais aussi la spécification de la catégorie d'autorisation dans le processus (un rôle à une participation active). Pendant l'exécution d'un processus avec une certaine autorisation, la chaîne de relations rôle - participation - processus (avec participation - condition d'autorisation ainsi que participation - valeur d'autorisation) est donc construite.

Un rôle peut être occupé par des personnes, des postes ou des systèmes d'informations. Le rôle forme le lien entre les processus et les ressources qui y participent. Il est défini par une agrégation d'attentes envers les ressources participant aux processus.

L'exécution d'un processus requiert des aptitudes (skills) que le rôle ou la ressource concerné(e) doit posséder. Pour être en mesure de définir des rôles de manière orientée processus, les processus doivent être évalués et les exigences des processus envers les personnes/systèmes participants doivent être définies. Les exigences envers les personnes/systèmes signifient plus précisément que les exigences sont le savoir et les aptitudes (skills) des personnes/systèmes. L'évaluation d'une aptitude est normalisée par une échelle d'évaluation affectée.

Ainsi, le diagramme de rôles vous permet de représenter des processus et des processus élémentaires, les ressources y participant, leurs aptitudes ou leurs aptitudes requises, ainsi que leurs autorisations.

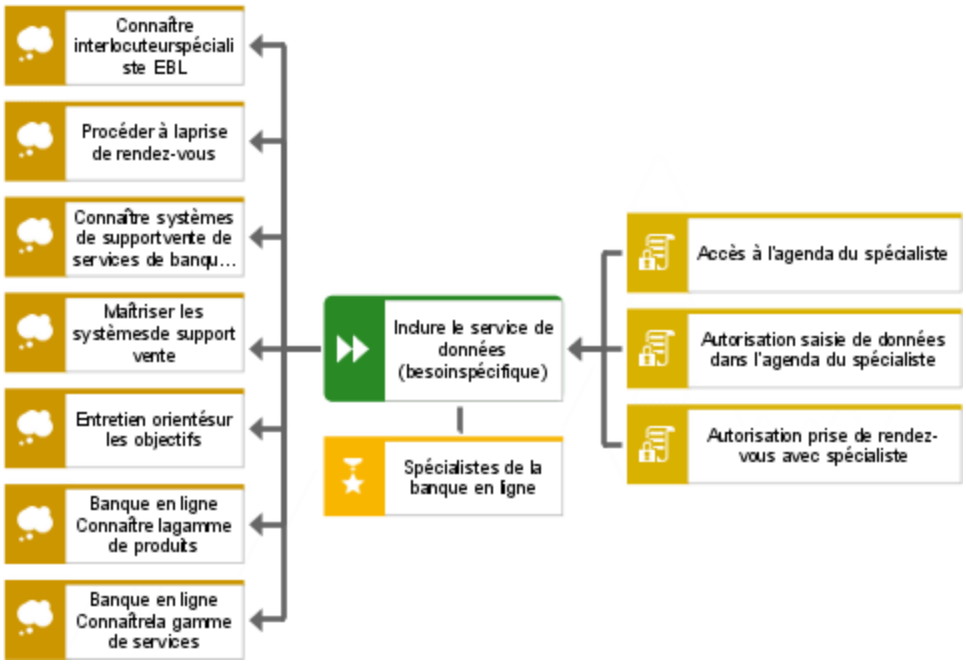


Illustration 93: Diagramme de rôles

Dans l'exemple de modèle sont montrés les exigences du processus élémentaire envers le rôle (aptitudes et autorisations) et parallèlement les exigences du rôle élémentaire envers la ressource quant aux aptitudes et autorisations.

Le diagramme est associé au processus élémentaire et au rôle élémentaire correspondants. L'association du diagramme au processus élémentaire permet de voir les exigences de la CPE de processus correspondante (correspond au modèle de référence de processus). Par l'association au rôle élémentaire il est possible de voir l'exigence du rôle élémentaire à la ressource concernant son aptitude et autorisations du diagramme de structure de rôle.

3.4.1.15 Modèle express

Le modèle du type **Modèle express** vous offre la possibilité de modéliser sans limites dues à la méthode. Dans le Modèle express, le type d'objet **Objet express** est disponible, avec plus de 30 symboles différents. Entre les objets express, des relations du type **est en relation avec** peuvent être créées. Plusieurs liaisons de ce type sont autorisées entre deux objets.

Les attributs par défaut peuvent être gérés pour le modèle, l'objet et la liaison.

Plusieurs modèles express peuvent être associés à chaque objet d'un type d'objet quelconque de la Méthode ARIS. Vous pouvez également associer autant de modèles de la Méthode ARIS que vous le souhaitez à un objet express, indépendamment du type de modèle.

3.4.1.16 Dessin de masque

Dans un dessin de masque, vous pouvez spécifier, dans le cadre de la conception du logiciel avec ARIS, les exigences spécialisées émises pour un dialogue ou un formulaire Web.

Dans la colonne **Mise en page**, vous déterminez la structure du dialogue ou du formulaire de page Web. La procédure lors de la conception par ex. d'un dialogue ressemble au travail avec l'éditeur de ressource dans un environnement de développement.

Les composants graphiques pouvant être placés dans la colonne **Mise en page** sont, entre autres, les zones de textes, les compteurs, les cases d'options et cases, les zones de liste modifiable, les boutons de commande, les contrôles d'arborescence et de liste, ainsi que les bitmaps et le texte statique. A l'aide du type d'attribut **Tabindex**, vous pouvez déterminer l'ordre dans lequel les différents éléments de masque sont appelés lorsque vous utilisez la touche de tabulation.

Différents éléments de données ou objets de fonctions peuvent être placés dans les colonnes **Données** et **Fonctions**. Grâce à une liaison du type **représente**, les objets conçus peuvent être mis en relation avec les éléments de données et les fonctions qu'ils éditent.

Chaque dessin de masque peut être associé à l'objet de masque correspondant qui est par ex. utilisé dans une CPE ou un modèle du type **Navigation de masque**. En outre, un dessin de masque peut également être associé au type d'entité, au cluster, au type d'objet complexe, à la classe ou à la fonction/fonction informatique qui est traité(e) avec le masque.

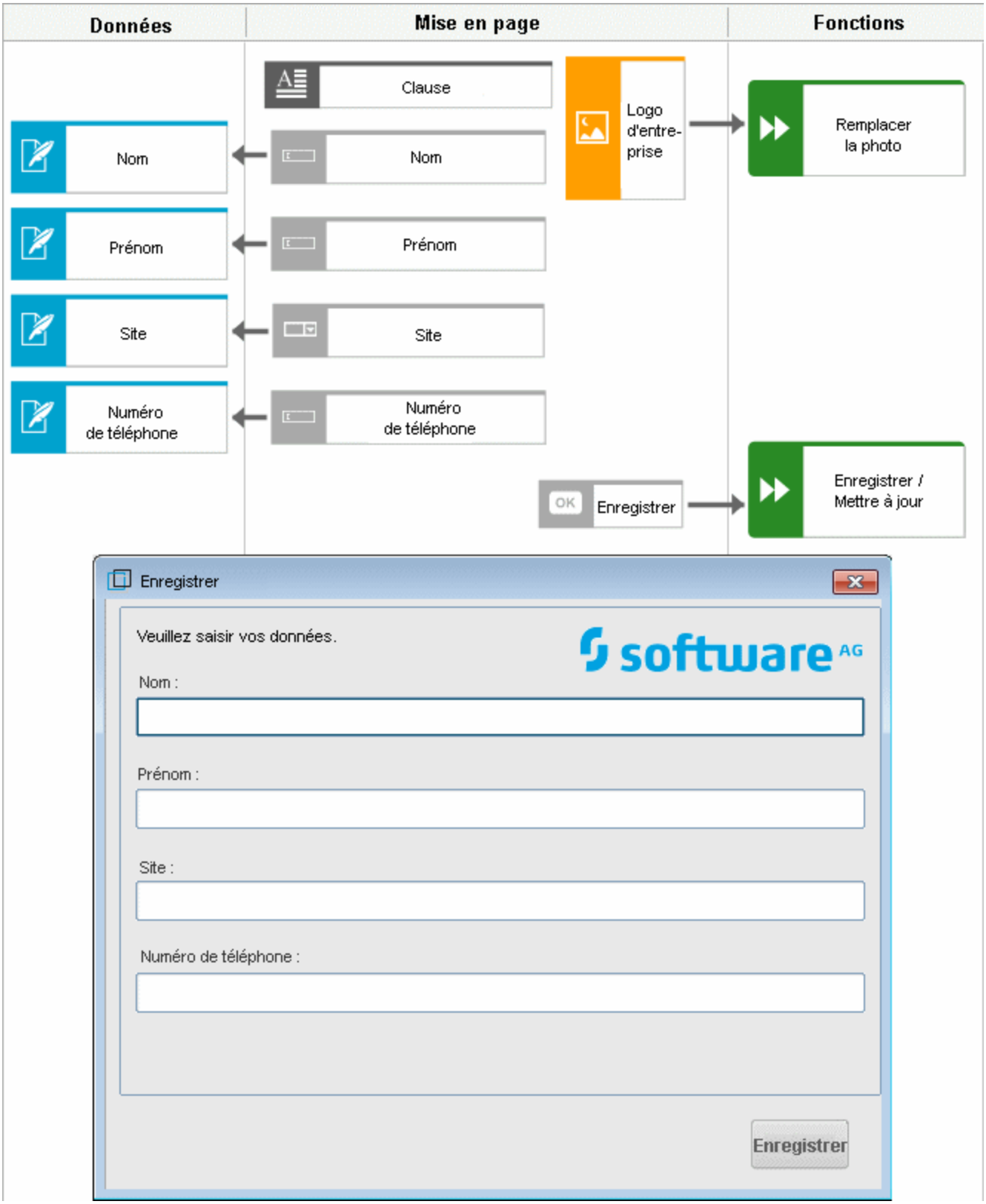


Illustration 94: Exemple de dessin de masque d'une boîte de dialogue d'enregistrement et la conversion en C++

3.4.1.17 Navigation de masque

Dans un modèle du type **Navigation de masque**, vous pouvez décrire la composition d'un masque contenant plusieurs masques partiels (par exemple, un site Web composé de plusieurs zones de formulaire ou cadres), ou décrire les transitions entre différents masques. Les transitions entre les masques peuvent être détaillées.

Exemple

Vous souhaitez expliciter qu'un élément de masque doit être activé avant que l'utilisateur puisse passer dans un autre masque. Affectez l'élément de masque déclenchant (du modèle **Conception de masque**) au masque à l'aide de la liaison **contient**. Tirez une liaison du type **appelle** de l'élément de masque vers le masque suivant.

Il est également possible de représenter la dépendance de la navigation d'événements. Différents événements peuvent survenir lorsque l'utilisateur quitte un masque. Lorsqu'un utilisateur a par exemple rempli la page d'enregistrement d'un magasin en ligne, l'enregistrement peut réussir ou échouer. Selon les cas, il bascule vers la page de contenu du catalogue ou revient dans la page d'enregistrement.

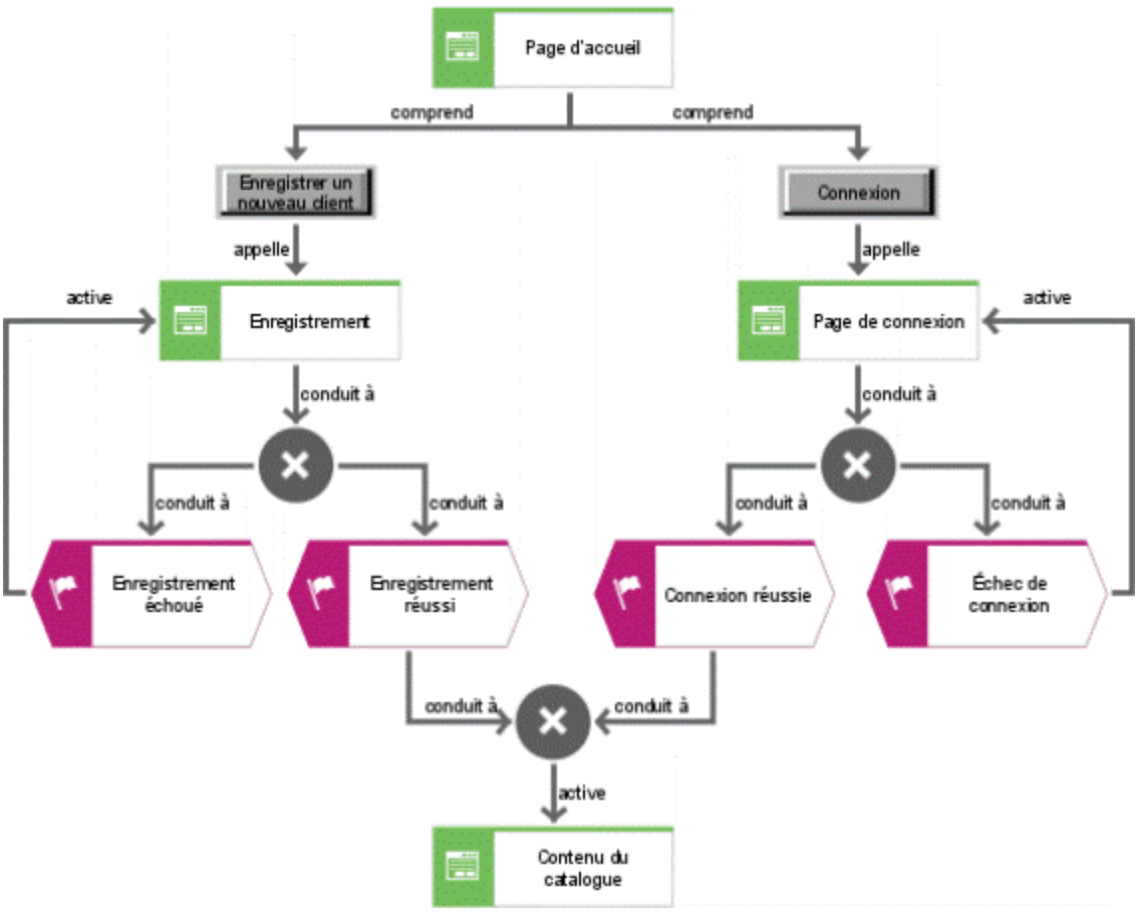


Illustration 95: Exemple pour la navigation de masque avec des événements

3.4.1.18 Matrice des champs d'activités

La matrice des champs d'activités représente les différents marchés dans lesquels une entreprise est implantée et permet de visualiser leur importance pour la réussite de l'entreprise.

Chaque marché est décrit :

- par le produit ou le service proposé ;
- par le groupe de clients à laquelle l'offre s'adresse.

Les produits et les services (objets de type **Prestation**) sont placés dans les cellules de la première colonne de la matrice des champs d'activités. Le groupe cible (différents éléments organisationnels) est placé dans les cellules de la première ligne. Vous définissez le marché en plaçant un objet de champ d'activité dans la cellule à l'intersection de la ligne de produit et de la colonne de groupe cible. Des relations implicites de type **appartient au champ d'activité** sont alors construites entre le produit/service et l'élément organisationnel.

Pour mettre en évidence l'importance d'un champ d'activité par rapport aux autres, cinq symboles sont disponibles ; ils permettent de définir l'importance sur des valeurs allant d'**insignifiant** à **très important**.

Lors de la modélisation, tenez compte du fait que les champs d'activités ne peuvent être placés qu'une fois dans la matrice.

Vous pouvez indiquer la stratégie d'entreprise pour laquelle chaque champ d'activité est important. Une stratégie décrit les procédures à long terme avec lesquels l'entreprise tente d'atteindre ses objectifs par rapport à son environnement.

La figure suivante montre une matrice de champs d'activités du secteur pharmaceutique.

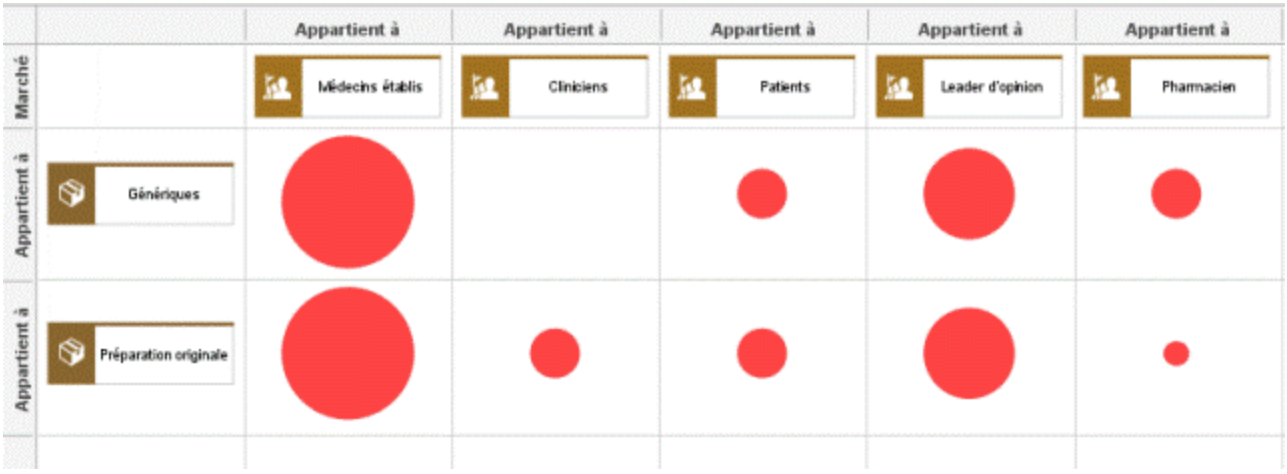


Illustration 96: Exemple de matrice des champs d'activités

Un diagramme cible peut également être associé à des champs d'activités. Le diagramme cible contient les objectifs définis pour le champ d'activité ainsi que les processus et les facteurs de succès qui permettent d'atteindre les objectifs fixés.

Les facteurs de succès du diagramme cible peuvent servir de base à une analyse de facteurs de succès si les attributs **Succès - réel**, **Succès - requis** et **Succès - concurrent** sont gérés dans le groupe d'attributs du même nom. Le succès peut être défini de **minimal** à **très important** à l'aide d'une échelle à 5 niveaux.

Procédure

Pour effectuer une analyse de facteurs de succès :

1. lancez le rapport ARIS à l'aide du menu contextuel du champ d'activité (**Evaluer > Rapport**),
2. Dans l'assistant de rapport, sélectionnez le script de rapport **Analyse FPS (Objet).rso** du groupe **BPM** dans le chemin par défaut.

Le rapport est généré au format HTML.

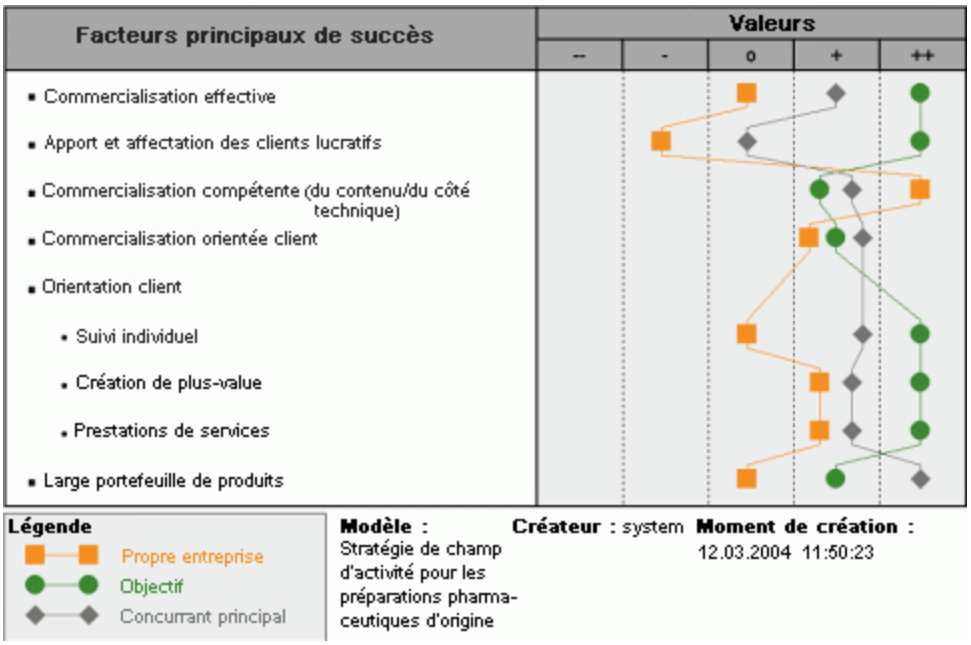


Illustration 97: Rapport

Vous pouvez également lancer l'analyse des facteurs de succès à l'aide du menu contextuel du diagramme cible. Sélectionnez ensuite le script de rapport **Analyse FPS (modele).rsm**.

3.4.2 Concept informatique

3.4.2.1 Diagramme d'accès

Les relations représentées par la suite entre les objets et qui ont été expliquées dans les descriptions du concept informatique des autres vues peuvent être admises dans le diagramme d'accès de la vue de gestion. Les différentes relations doubles sont traitées séparément pour obtenir une plus grande lisibilité de la représentation.

3.4.2.2 Fonctions avec données

Vous pouvez définir à cet endroit les flux d'informations entre les types d'applications, les types de modules ou les fonctions informatiques. A cet effet, un objet de flux d'informations est créé entre les types d'applications ou types de modules correspondants. Pour spécifier plus précisément le flux d'informations entre les types de systèmes, l'objet de flux d'informations est lié à un diagramme MERe ou à un diagramme de table. Les objets du flux d'information peuvent ainsi se trouver au niveau des règles de gestion, du concept informatique ou de l'implémentation.

La figure suivante en montre un exemple.

Illustration 98: Flux de données entre types d'applications

Outre les flux d'information, les données d'entrée et de sortie peuvent être représentées sous forme d'objets de données des règles de gestion ou du concept informatique pour chaque type d'application, type de module et type de fonction informatique. La direction de la flèche vous permet de déterminer si le flux de données est entrant (Input) ou sortant (Output).

La figure suivante en montre un exemple.

Illustration 99: Données E/S au niveau du concept informatique

3.4.2.3 Organisation avec fonctions

Les relations entre les aspects organisationnels et les aspects fonctionnels définis au niveau du concept informatique permettent de répondre aux questions suivantes :

- Qui (quelles unités organisationnelles, quels postes, quelles personnes,...) est responsable des types d'applications et de modules de la vue des fonctions au niveau du concept informatique ou qui est l'utilisateur de ces systèmes ?
- Dans quels sites (vue organisationnelle) de l'entreprise les types d'applications ou de modules sont-ils utilisés ?
- Sur quelles plates-formes disponibles de l'entreprise (type de composant matériel (vue organisationnelle)) les types d'applications peuvent-ils fonctionner ?

Pour répondre à la première question, vous pouvez tirer des liaisons entre les unités organisationnelles de l'organigramme (unités organisationnelles, postes et personnes) et les objets du diagramme de type d'application (type d'application, type de module, fonction informatique,...) dans le diagramme d'accès. A cet endroit, vous pouvez encore définir quelle signification cette relation doit avoir. Il est fait la distinction suivante :

- Une unité organisationnelle peut être **professionnellement responsable** d'un type d'application.
- Une unité organisationnelle peut être **responsable du développement** d'un type d'application.
- Une unité organisationnelle peut être un **utilisateur** d'un type d'application.

Afin de répondre à la question du site, des sites de la vue organisationnelle peuvent être affectés aux types d'applications, de modèles et de fonctions informatiques.

Dans ce concept informatique, il ne s'agit pas de traiter avec des types d'applications individuels au sens de licences individuelles mais bien avec des types d'applications. Ainsi, aucun emplacement de type d'application n'est défini au moyen de cette relation (cela est réalisé au niveau d'implémentation) mais les emplacements possibles d'un type d'application particulier sont soulignés.

Vous pouvez définir dans le concept informatique de la vue organisationnelle les types de composants matériels étant disponibles dans l'entreprise. Vous pouvez mettre en relation ces types de composants matériels avec les types d'applications dans la vue de gestion. Ce qui permet de déterminer les plates-formes matérielles sur lesquelles les types d'applications, de modules ou de fonctions informatiques peuvent fonctionner. Vous pouvez aussi affecter à cet endroit les types d'interfaces utilisateur repris dans la vue des fonctions, les types de systèmes d'exploitation et les types du SGBD aux types de composants matériels.

Vous trouverez une liste des relations possibles dans un diagramme d'accès dans l'aide **Référence de la Méthode ARIS**.

Vous trouverez des exemples de relations dans la figure suivante.

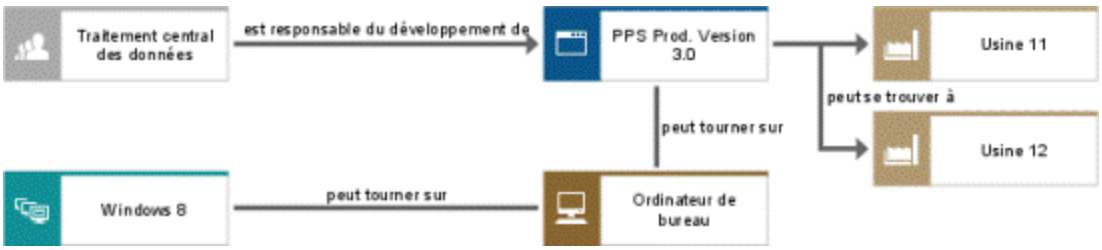


Illustration 100: Diagramme d'accès (extrait)

3.4.2.4 Diagramme de déroulement de programme

Dans le diagramme d'accès, il est possible de créer les liaisons aux types d'objets de la vue organisationnelle et de la vue des données pour les types d'applications, les types de modules et les types de fonctions informatiques spécifiés dans le diagramme de type d'application (cf. chapitre **Diagramme d'accès** (page 97)). Dans ce type de modèle, il n'est cependant pas possible de représenter directement l'affectation de fonctions des règles de gestion. Cette affectation a lieu dans le diagramme de type d'application. Il n'est également pas possible de représenter directement les déroulements chronologiques possibles des types d'applications, types de modules et types de fonctions informatiques. En suivant strictement l'architecture ARIS, il n'est possible de comprendre ces connexions qu'à l'aide de la navigation à travers différents types de modèles.

Dans le cadre de l'agencement du système, les types de diagrammes qui permettent une vue intégrée de tous les aspects de l'agencement du système se sont cependant imposés (par ex. Plan de déroulement du programme (PD) (page 100)).

C'est pour cela qu'**ARIS** propose le type de modèle **Diagramme de déroulement de programme**. Ce modèle permet de modéliser toutes les relations avec les types d'applications, types de modules et types de fonctions informatiques qui sont mis à votre disposition par **ARIS** dans les autres types de modèles, indépendamment de la répartition des vues d'**ARIS**. Il est également possible de représenter les déroulements chronologiques des types d'objets cités. Pour ce faire, ce type de modèle propose également des événements. De façon similaire à l'affectation de fonctions et d'événements dans la CPE, il est possible de définir des suites de modules dans le diagramme de déroulement du programme. L'événement doit ici être compris dans le sens d'un trigger, qui lance des types de modules et des types d'applications. Les ramifications peuvent être représentées avec les règles que l'on trouve également dans la CPE. Cependant, dans le diagramme de déroulement de programme il est possible, contrairement à la CPE, de définir les déroulements sans devoir faire intervenir des événements.

3.4.2.5 Plan de déroulement de programme (PD)

Le plan de déroulement de programme (PD) sert à représenter des suites de traitement dans un programme. Les liaisons entre les objets démontrent la suite du traitement. Les données ne sont pas représentées par ce diagramme.

La figure suivante illustre un exemple simplifié de suites de traitement d'un guichet électronique. Dans la représentation des suites de traitement, une orientation vers l'implémentation apparaît nettement.

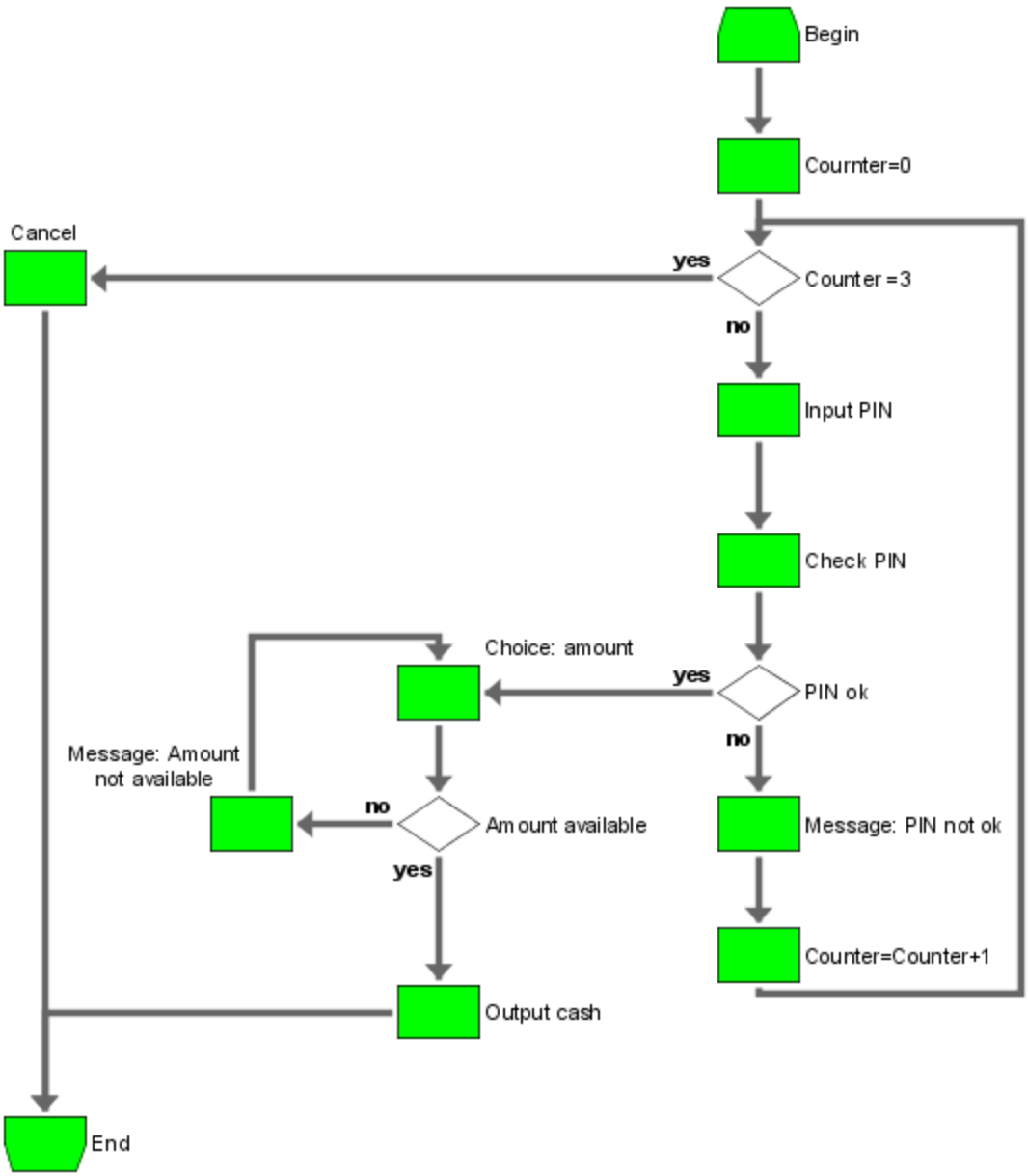


Illustration 101: Exemple d'un plan de déroulement de programme (PD)

3.4.2.6 Diagramme de masque

Le champ d'application du diagramme de masque est la description de masques dans le développement du logiciel. L'objectif étant de déduire automatiquement les masques à partir du diagramme de masques.

Le diagramme de masque forme par conséquent la structure et, jusqu'à un certain point, la fonctionnalité des masques d'écrans. La règle suivante s'applique : la structure du diagramme de masque allant de gauche à droite et du haut vers le bas correspond à la géométrie de l'interface décrite.

Le symbole central est le masque ; il représente une fenêtre, conformément à la terminologie de Windows. Cette fenêtre peut contenir plusieurs onglets (symbole **Page**). De manière générale, la surface peut être divisée géographiquement en différents domaines au moyen d'une table (symbole **Section** pour une ligne et symbole **Colonne** pour une colonne). Les symboles **Section** et **Colonne** peuvent s'imbriquer à volonté, et peuvent ainsi former des surfaces complexes. Des tables (symbole **Table des masques**), des zones de saisie de texte (symbole **Attribut type d'objet complexe**) ainsi que des graphiques (symbole **Bitmap**) et des textes explicatifs (symbole **Texte**) peuvent être placés sur l'interface. Le symbole **Mise en page** vous permet d'affecter des propriétés de représentation aux objets **Masque**, **Page**, **Section**, **Colonne**, **Table des masques**, **Attribut type d'objet complexe** et **Texte**.

D'autres symboles peuvent être utilisés pour la description de la surface d'écran.

La figure suivante montre un exemple de diagramme de masque. La deuxième figure illustre le masque qui en provient.

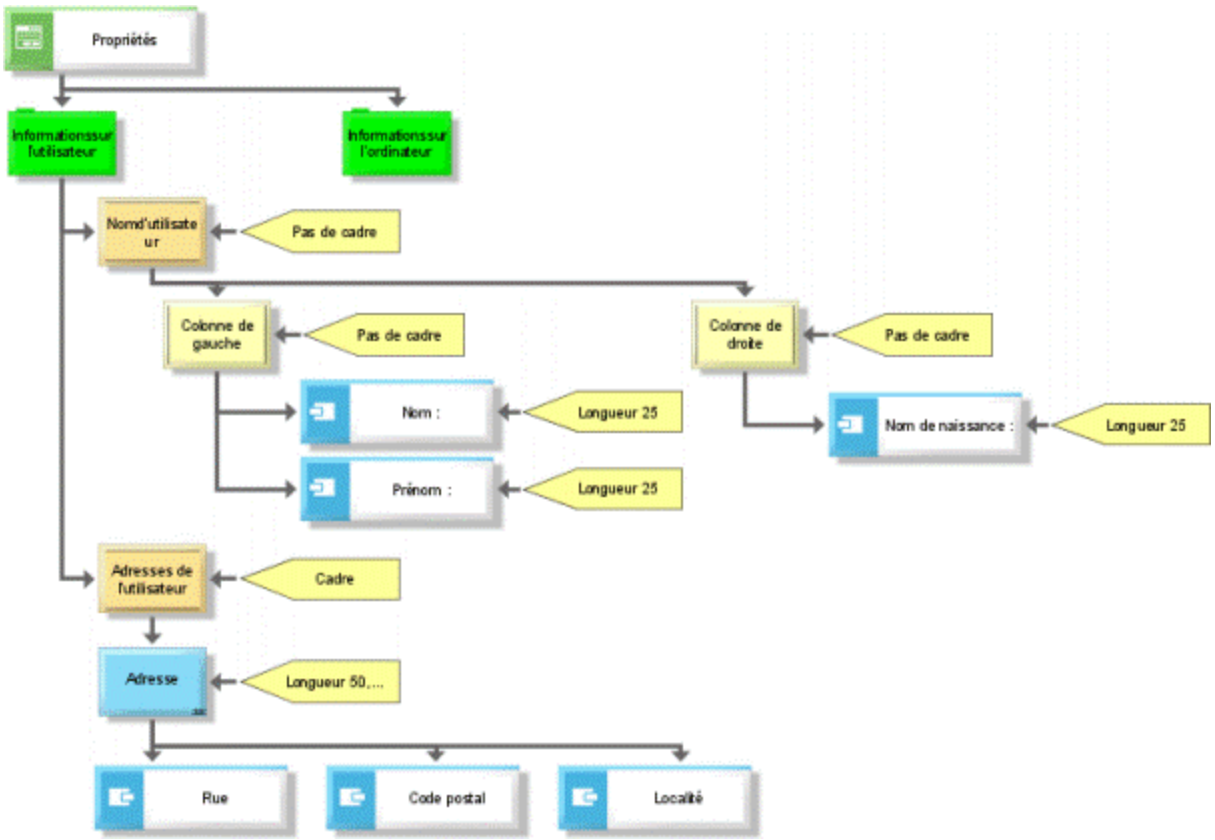


Illustration 102: Exemple d'un diagramme de masque

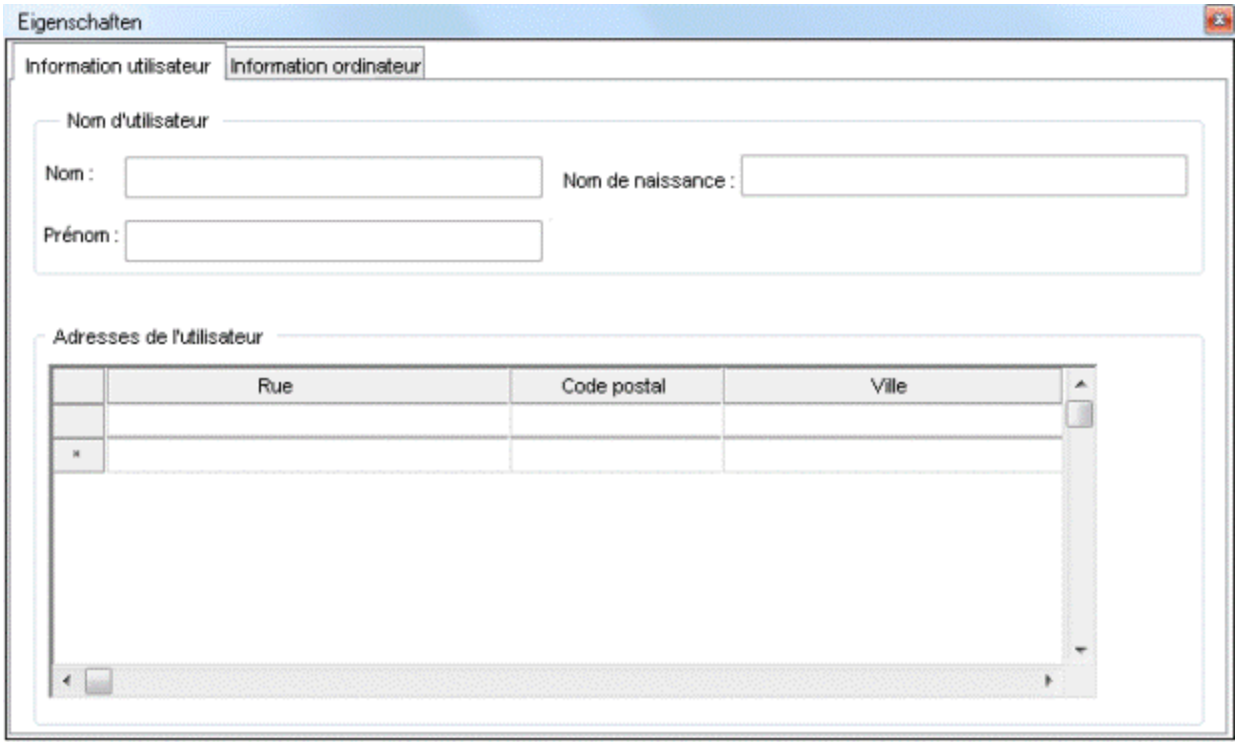


Illustration 103: Masque provenant du diagramme de masque, illustré dans la figure précédente.

3.4.3 Implémentation - Diagramme d'accès (physique)

Les questions traitées dans le concept informatique de la vue de gestion sont également objet du niveau d'implémentation. Contrairement au niveau du concept informatique, on observe les exemplaires concrets des différents objets, et non pas les types d'objets. Ainsi, les relations entre les applications concrètes et les unités organisationnelles sont observées, et pas les relations entre les types d'applications et les unités organisationnelles

Les relations représentées comme suit sont modélisées dans le diagramme d'accès (physique).

3.4.3.1 Fonctions avec données

Pour répondre à la question "quelles données passent entre les applications", des objets de flux d'informations peuvent être créés entre les objets d'application de la vue de fonction. Contrairement au niveau du concept informatique, il ne s'agit toutefois pour ces objets d'application pas de type d'application mais d'exemplaires concrets (différentes licences). Il est donc possible de relier des applications, des modules et des types de sous-programmes entre eux au moyen de liaisons de flux de données. Si, au niveau du concept informatique, il a été défini que le type de module **Système de distribution SD Version 2.1** peut échanger des données avec le type de module **Système de gestion des matières MM version 1.2**, il figure alors au niveau d'implémentation que le module **SD numéro de licence 1234**, installé concrètement, échange des données avec le module **MM, numéro de licence 2352** et le module **MM, numéro de licence 34234**. Les deux modules MM sont des modules du type **Système de gestion des matières MM version 1.2**. Cette situation est illustrée par la figure suivante.

Illustration 104: Flux de données

Afin de pouvoir spécifier plus précisément les objets de données échangés entre les systèmes, les types de modèles de la vue de données correspondants sont associés aux objets de flux d'information.

Outre les flux de données entre les applications, les données d'entrée et de sortie peuvent également être spécifiées pour chaque application. Il existe deux raisons pour que les relations soient représentées dans le diagramme d'accès (physique). Dans le premier cas de

figure, il s'agit, pour les objets de données, d'objets du diagramme de tables (tables, champ, vue (physique)), se trouvant dans la vue de données au niveau d'implémentation. Ces objets de données peuvent être reliés aux objets d'application du niveau de concept informatique ou du niveau d'implémentation au moyen des relations d'entrée/de sortie. Dans le deuxième cas de figure, il s'agit, pour les objets d'application, d'applications concrètes ou de modules du niveau d'implémentation, reliés aux objets de la vue de données.

Une règle générale peut donc être définie :

Si un des types d'objets des relations d'entrée/de sortie est issu du niveau d'implémentation de la vue correspondante, les relations de la vue de processus seront également représentées au niveau d'implémentation (diagramme d'accès (phys.)).

La figure suivante en montre un exemple.

Illustration 105: Relations d'entrée/de sortie

3.4.3.2 Organisation avec données

Les mêmes questions que dans le concept informatique sont mises en avant :

- Quelles unités organisationnelles sont responsables des objets de données ?
- Qui a le droit d'accès à quels objets de données ?
- Dans quels composants matériels les objets de données sont-ils enregistrés ?

Contrairement au concept informatique, les relations sont ici créées avec les objets de données représentés au niveau d'implémentation de la vue de données.

La responsabilité des objets de données est donc définie pour des relations et des attributs pour des structures physiques, à savoir, des tables, des champs et leurs exemplaires [table (exemplaire), champ (exemplaire)].

Cette dépendance est représentée par des liaisons générées dans le diagramme d'accès (phys.) entre les objets de la vue organisationnelle (unité organisationnelle, poste, personne...) et les objets du diagramme de table (table, champ, vue (phys.),...) déjà évoqués antérieurement.

Lors de la création de liaisons entre les unités organisationnelles, les tables et les champs, la signification de la relation doit être définie à chaque fois. **Est responsable de** signifie que l'unité organisationnelle est responsable du contenu de chaque table ou de chaque champ ; **accède** signifie que le poste ou la personne a droit d'accès aux objets de données représentés.

Outre la définition des droits d'accès et des responsabilités, il est désormais possible de définir avec l'objet Composants matériels (vue organisationnelle/ implémentation), dans quels composants matériels existants les objets d'informations de l'entreprise se trouvent concrètement.Ces composants matériels étant par exemple déterminés de façon univoque au moyen de numéros d'inventaire de l'entreprise. Pour ce faire, des objets d'information du niveau d'implémentation (tables, champs, etc.), de concept informatique (relations, attributs) ou de règles de gestion (types d'entités, cluster/modèle de données, etc.) peuvent être reliés dans le diagramme d'accès (physique) avec l'objet **Composants matériels**.

Un exemple est illustré dans la figure ci-dessous.

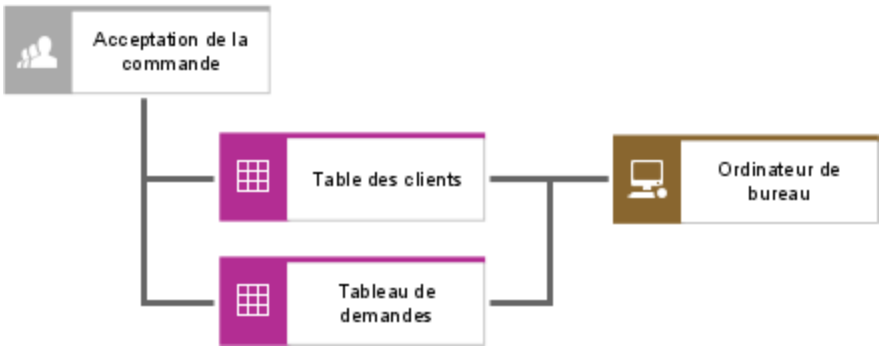


Illustration 106: Affectations aux composants matériels

3.4.3.3 Organisation avec fonctions

Les relations définies dans le diagramme d'accès (phys.) entre les objets de la vue organisationnelle et la vue de fonction permettent de répondre aux questions suivantes :

DANS QUELS COMPOSANTS MATÉRIELS SE DÉROULENT QUELLES APPLICATIONS, ET QUELS TYPES D'APPLICATIONS PEUVENT S'Y DÉROULER ?

Pour représenter ces dépendances, les relations **est la plateforme de** et **peut être la plateforme de** peuvent être modélisées entre les objets d'application du niveau d'implémentation (application, module, sous-programme, etc.) ou du niveau du concept informatique (type d'application, type de module, etc.) et le type d'objet **Composants matériels** de la vue organisationnelle.

La figure suivante en montre un exemple.

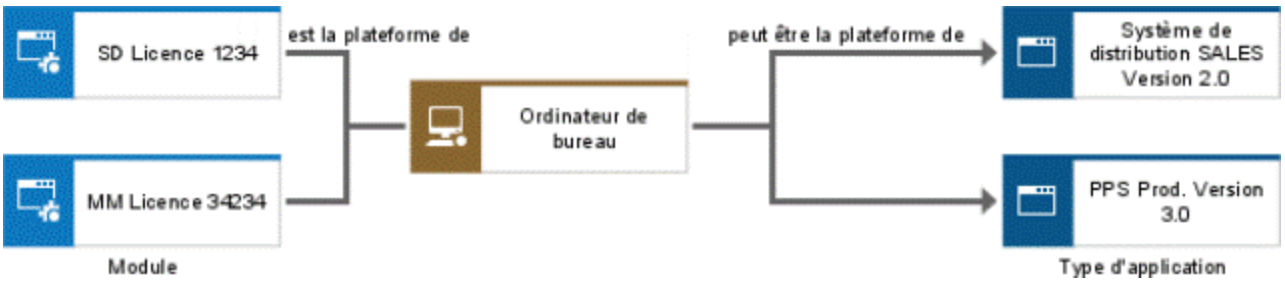


Illustration 107: Composants matériels sous forme de plate-forme

QUELLE UNITÉ ORGANISATIONNELLE EST L'UTILISATEUR D'UNE APPLICATION CONCRÈTE ?

Si les utilisateurs accédant à des types d'application sont définis au niveau du concept informatique, cette relation peut maintenant être définie pour des applications concrètes (licences individuelles) au niveau d'implémentation. Ainsi, plusieurs licences d'une version du

type d'application **ARIS Architect** peuvent par exemple exister dans différentes configurations au sein de l'entreprise. Dans le diagramme d'accès (phys.), il est maintenant possible de représenter quel utilisateur travaille avec quelle licence. Pour ce faire, les types d'objets **Unité organisationnelle**, **Poste** et **Personne** peuvent être reliés avec les types d'objets **Application** et **Module** par la liaison **est utilisateur**. Un exemple est illustré dans la figure suivante.

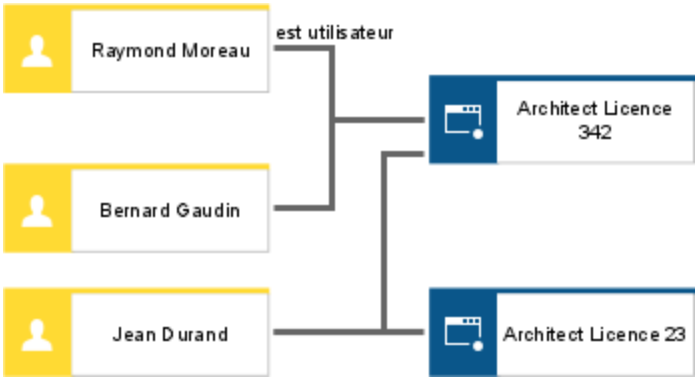


Illustration 108: Utilisateur et application

SUR QUELS SITES DE L'ENTREPRISE SONT INSTALLÉES DES APPLICATIONS ?

Dans le concept informatique, la relation **Type d'application - Site** a permis de déterminer les types d'applications qui peuvent se trouver sur des sites déterminés de l'entreprise. Pour déterminer ensuite les sites de l'entreprise où sont utilisées les licences acquises pour un type d'application, le diagramme d'accès (physique) permet de relier les sites également aux types d'objets **Système d'application**, **Module** et **Fonction informatique**.

La figure suivante en montre un exemple.

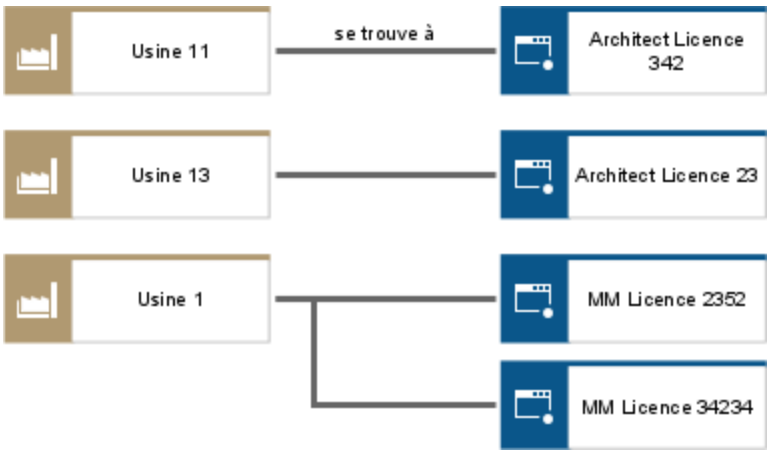


Illustration 109: Affectations de sites

L'aide **Référence de la Méthode ARIS** regroupe toutes les relations à disposition dans le diagramme d'accès (physique).

3.5 Modélisation de prestation/produit

ARIS met à votre disposition plusieurs types de modèles pour décrire les prestations dans une entreprise.

Une prestation est réalisée pendant le processus de plus-value. Elle est le résultat d'une action humaine ou d'une procédure technique. Une prestation/produit peut être soit une prestation de service, soit une prestation en nature.

Les prestations en nature peuvent correspondre à un produit de consommation, un type de matières, un type de moyens d'exploitation, un type de moyens auxiliaires techniques ou un type d'emballage. Déclencheur pour réaliser une prestation/produit est toujours un besoin d'une unité organisationnelle ou d'un client. Une prestation en nature est proposée au client sous forme d'un bien existant sous forme matérielle.

Une prestation de service est un bien immatériel, pour lequel la production et la consommation sont simultanées.

Les fournisseurs de prestations de services sont généralement les banques, les assurances ou l'administration publique.

Plus l'orientation client est forte dans la branche d'un fournisseur de prestations en nature, plus il est important pour lui d'observer avec attention les prestations de service pouvant être proposées autour du produit et de les développer.

Les différents types de modèles d'ARIS se prêtent tant à la description de prestations en nature ou de services pures qu'à la description des prestations panachées.

Les types de modèles suivants sont à votre disposition pour la modélisation de prestations :

- Diagramme d'échange de prestations/produits
- Arbre de prestations/produits
- Diagramme de rattachement des produits
- Arbre de produits
- Matrice de choix des produits

3.5.1 Diagramme d'échange de prestations/produits

Le diagramme d'échange de prestations/produits permet d'illustrer la production ainsi que l'échange de prestations/produits dans l'entreprise. Une prestation peut être soit une prestation de service, soit un produit et est représentée par des symboles correspondants. Les produits peuvent être des types de matières, des types de moyens d'exploitation, des types de moyens auxiliaires techniques et/ou des types d'emballages. Vous les avez déjà rencontrés dans la CPE (avec flux de matières). Les prestations/produits en tant qu'entrée et/ou sortie de fonction peuvent être connectés avec les événements de début et/ou de fin de ces fonctions.

Cet échange de prestations/produits entre les fonctions de gestion d'entreprise peut être utilisé de façon intéressante à partir d'un niveau d'abstraction se trouvant entre le diagramme de chaînes de plus-value et la CPE. Les relations d'échange des prestations/produits de la vue organisationnelle sont à illustrer à côté des relations d'échange d'une vue fonctionnelle. Vous trouverez des possibilités de modélisation dans le diagramme d'échange de prestations/produits.

La figure suivante illustre un exemple de diagramme d'échange de prestations/produits.

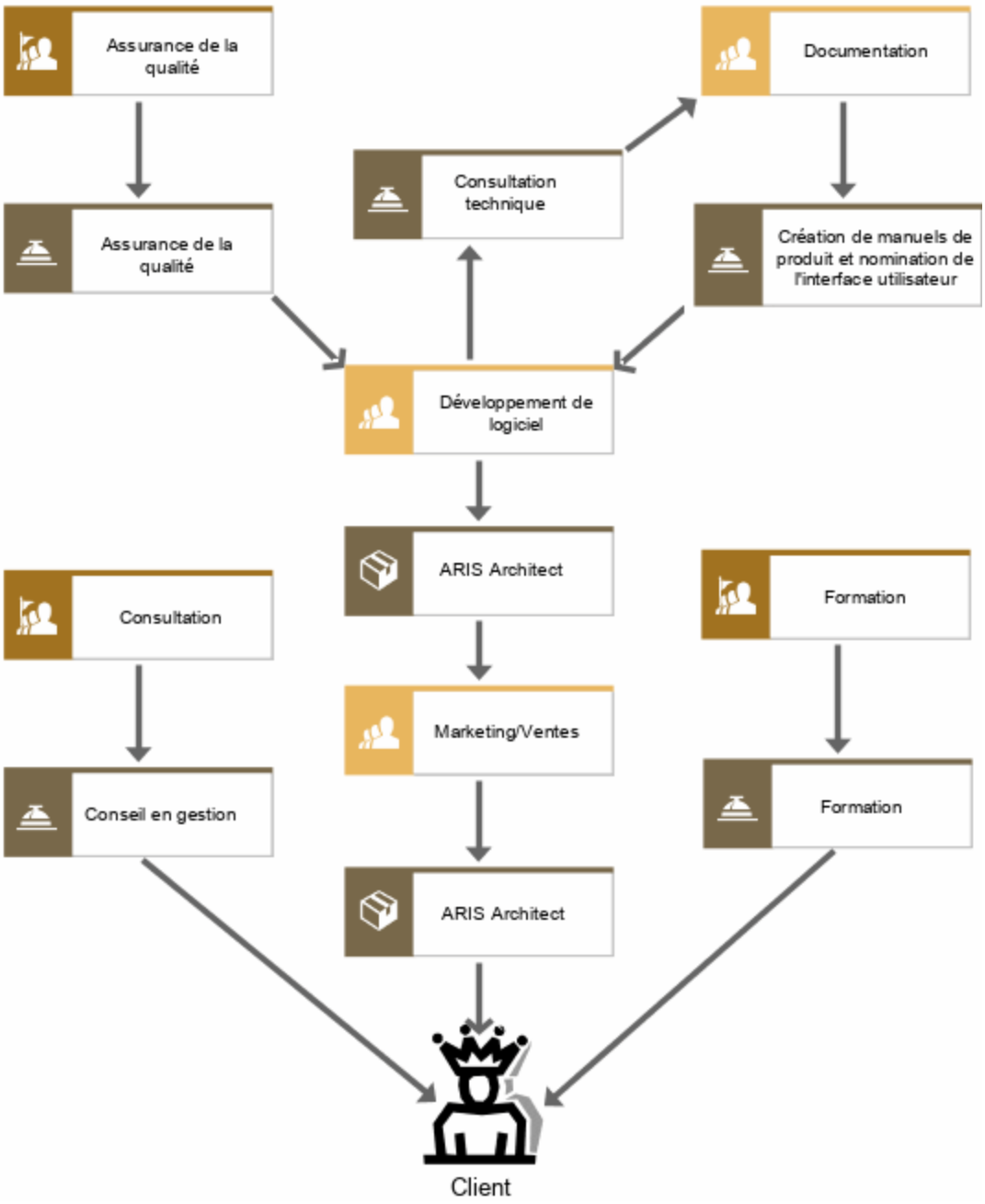


Illustration 110: Exemple d'échange de prestations/produits chez un fabricant de logiciels

3.5.2 Arbre de prestations/produits

Les prestations/produits peuvent être contemplées à différents niveaux d'abstraction. Il est donc pertinent d'enregistrer ces relations dans un modèle dans lequel les différentes parties de prestations dont se compose une prestation/produit sont représentées. Cet aspect statique est représenté dans l'arbre de prestations. Un produit complexe contient par

exemple souvent différents modules qui se décomposent eux-mêmes en diverses pièces détachées. Chacun de ces éléments peut être considéré comme une prestation/produit.

A l'aide de la liaison **est en relation avec** qui est également autorisée entre les prestations et l'arbre de prestations/produits, les dépendances de toutes sortes peuvent être décrites. Les relations entre le crédit à la consommation et le compte courant via lequel les paiements doivent être effectués en font partie.

Des relations de substitution vers d'autres prestations/produits, comme des produits ou des prestations de service (potentiels) de remplacement peuvent être illustrées.

Les relations des prestations/produits aux objectifs (de l'entreprise) sont aussi représentées dans ce modèle statique.

La figure suivante illustre un exemple d'arbre de prestations/produits.

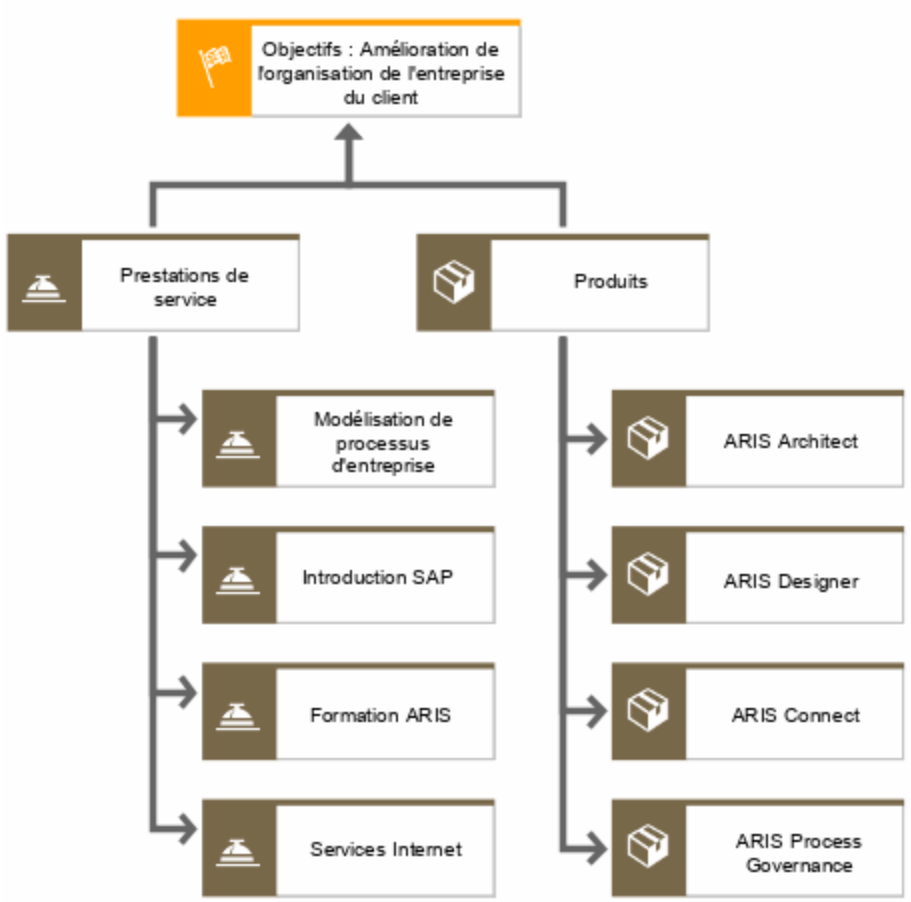


Illustration 111: Arbre de prestations/produits

3.5.3 Diagramme de rattachement des produits

Outre les diagrammes de prestations/produits qui appartiennent au groupe des modèles graphiques, il existe également des modèles de produits permettant une représentation plus abstraite. Le diagramme de rattachement des produits sert en premier lieu à analyser la fabrication de produits dans l'administration publique. Ce modèle permet, comme le

diagramme d'échange de prestations/produits, d'illustrer les unités organisationnelles qui fabriquent ou utilisent des produits, les fonctions nécessaires pour fabriquer ces derniers ou encore les fonctions pour lesquelles les produits livrent une entrée. Les bases (légales) de la commande des différents produits sont montrées en supplément. Les objectifs devant être atteints avec les produits peuvent également être représentés.

La figure suivante illustre un extrait de diagramme de rattachement des produits de l'administration publique.

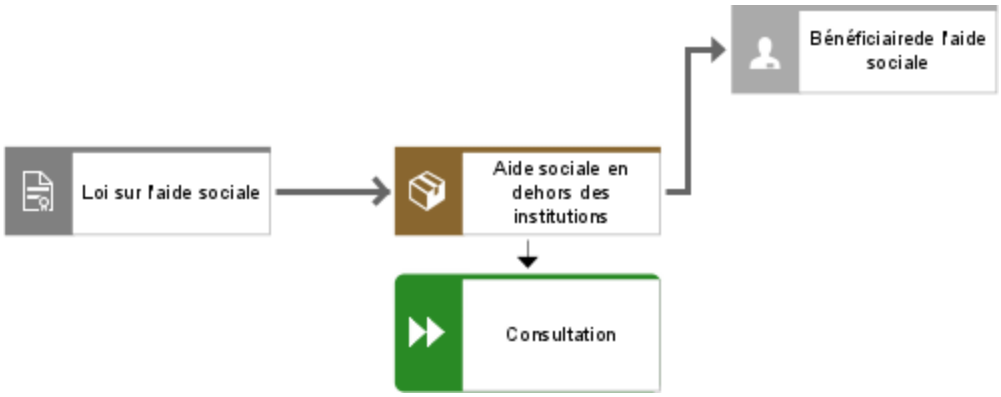


Illustration 112: Exemple de diagramme de rattachement des produits

Le type de modèle offre également la possibilité de décrire des aspects qui sont en relation avec la commercialisation du produit.

Ces aspects sont explicités lors de l'exemple simplifiés de produits bancaires.

Avec l'expansion d'Internet et le nombre croissant de connexions dans les foyers au cours des 10 dernières années, les services bancaires en ligne ont également connu un essor important. Simultanément, le pouvoir d'achat des jeunes a augmenté, et leur importance en tant que groupe cible s'est affirmée.

Cela a entraîné une diversité d'offres pour une prestation/produit qui à l'origine se limitait au **compte courant**.

Il peut par ex. être proposé comme compte Vermeil, dont les titulaires sont pris en charge par les collaborateurs d'une filiale. Ce produit est par ex. proposé à des clients plus âgés, moins familiarisés avec les nouvelles technologies, et plus très mobiles pour des raisons de santé.Ces clients trouvent important d'avoir affaire à des personnes qu'ils connaissent et qui peuvent leur apporter un conseil personnalisé. Les frais de gestion pour un tel compte peuvent être légèrement supérieurs à la moyenne.

Toutefois, dans le cadre de la banque en ligne, le compte courant peut également être proposé sous forme de compte Jeunes peu onéreux. Ce produit s'adresse à des jeunes de 12 à 20 ans, familiarisés avec Internet, mais dont les moyens financiers sont limités. Les frais doivent être réduits au maximum.

Les graphiques suivants montrent des diagrammes de rattachement des produits pour ces deux variantes.

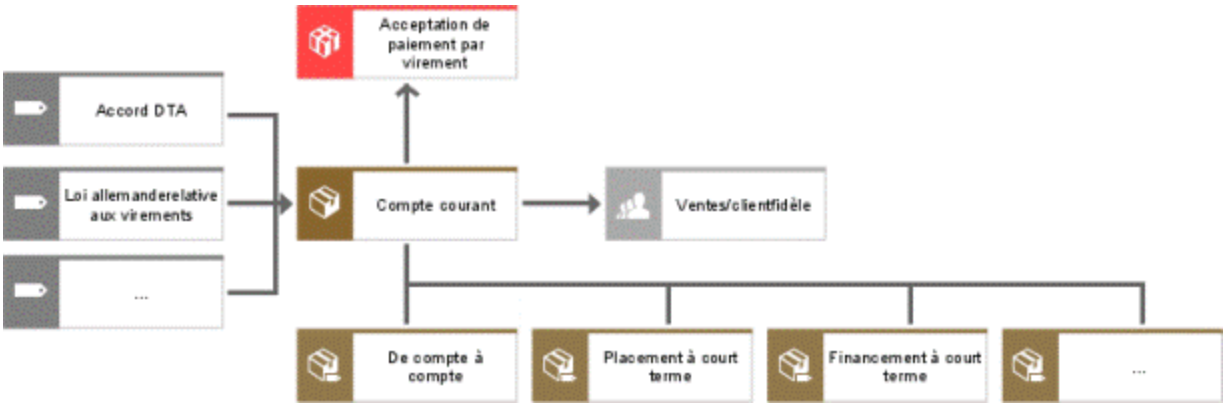


Illustration 113: Diagramme de rattachement des produits - compte courant

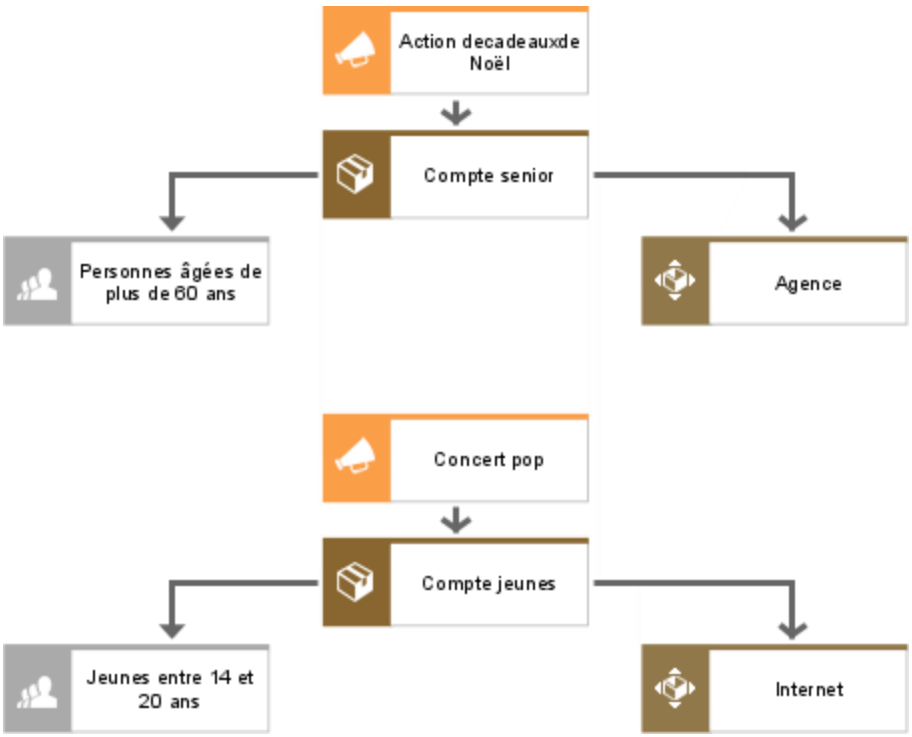


Illustration 114: Diagramme de rattachement des produits - produits à vendre

Les prestations/produits **Compte Jeunes** et **Compte Vermeil** ont été créées comme variantes d'objets du compte courant et caractérisées par l'attribut **Produits à vendre**. Un produit à vendre est une prestation/produit créée par l'entreprise. Il est proposé sous différents noms dans différentes branches. Différents instruments de politique marketing sont utilisés en général pour les différents produits à vendre.

À l'aide du composant ARIS, vous pouvez créer autant de produits à vendre que possible à partir d'un produit.

3.5.4 Arbre de produits

L'objectif de l'arbre de produits est d'analyser la composition de produits dans l'administration publique. Ce modèle correspond en gros à l'arbre de prestations/produits, si ce n'est que la possibilité de la modélisation de produits de remplacement n'existe pas. L'arbre de produits appartient au niveau des règles de gestion de la vue de prestations/produits.

Un arbre de produits est représenté dans la figure suivante.

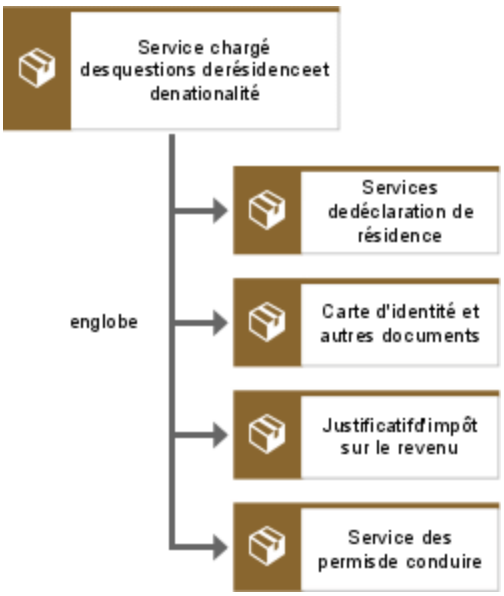


Illustration 115: Classification du groupe de produits "Affaires habitants et nationalités" à l'aide d'un arbre de produits.

3.5.5 Matrice de choix des produits

La matrice de choix des produits met en avant une unité organisationnelle et les produits dont elle est responsable. Ces produits peuvent être affectés aux fonctions nécessaires à leur création. Ce modèle fonctionne comme un point de départ à partir duquel vous pouvez naviguer vers des organigrammes, des arbres de produits et les processus importants pour la création des prestations/produits. La figure illustre un exemple de matrice de choix des produits.

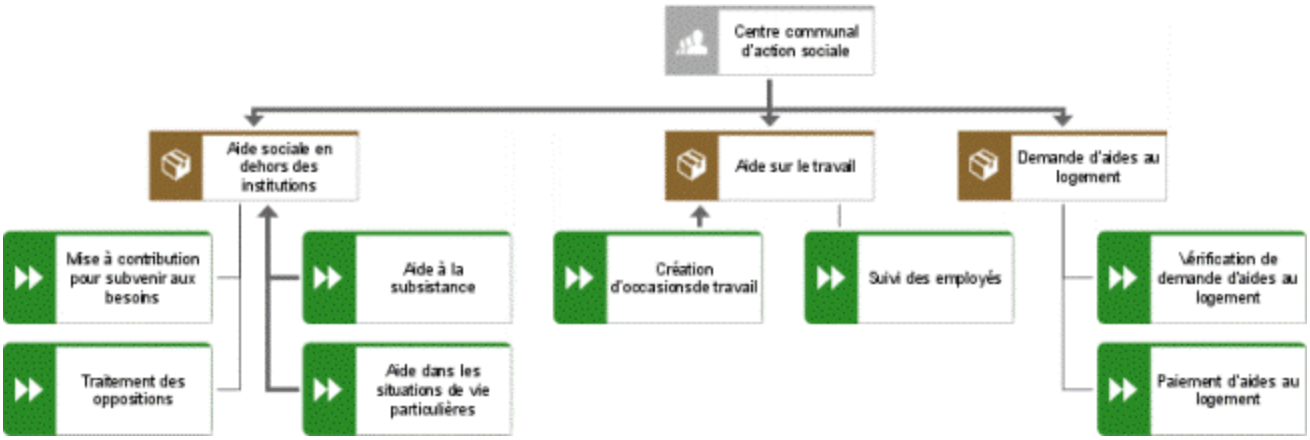


Illustration 116: Matrice de choix des produits de l'Assistance sociale

4 Unified Modeling Language (UML) dans ARIS

4.1 Introduction

UML (Unified Modeling Language) est un langage de modélisation orienté objet dont les structures linguistiques sont standardisées par un groupe de travail de l'OMG (Object Management Group). UML se fonde sur les approches orientées objets d'OMT, Booch et OOSE.

4.2 ARIS UML Designer - Standard UML pris en charge

ARIS UML Designer 9.x prend en charge totalement le standard UML 2,5.

La spécification UML est disponible
sous **https://www.omg.org/spec/UML/2.5/Beta2/PDF/**.

5 Méthodes de gestion des connaissances

5.1 Introduction

La gestion des connaissances permet de gérer systématiquement les connaissances qui sont devenues une ressource de plus en plus importante au sein de l'entreprise. Cette gestion englobe le développement, le contrôle, le soutien et le développement de stratégies, de processus, de structures organisationnelles et de technologies pour un traitement efficace des connaissances au sein de l'entreprise. Toutes les activités concernant l'acquisition, la préparation, la transmission et l'utilisation des connaissances sont regroupées dans la gestion des connaissances. En règle générale, le traitement des connaissances n'est pas un processus isolé ; cette activité a principalement lieu dans les processus d'entreprise opérationnels et ceux liés à l'ordonnancement. Il est donc nécessaire d'avoir une vue intégrée des processus d'entreprise, du traitement des connaissances, des structures organisationnelles, de systèmes d'information, etc.

Une grande partie de ces aspects peut être illustrée par des méthodes ARIS existant depuis longtemps, comme p. ex. les CPE, les organigrammes, les diagrammes de rattachement de fonctions, les MERe, etc. Si vous souhaitez une représentation, une analyse et une amélioration précises du traitement des connaissances, d'autres moyens de représentation sont nécessaires, p. ex. pour l'identification et la structure du contenu de catégories des connaissances pertinentes, pour la description de la répartition des connaissances dans l'organisation ou encore pour la modélisation de l'acquisition et de l'utilisation des connaissances dans les processus d'entreprise.

Nous avons donc ajouté à cet effet deux nouveaux types d'objets, **Catégorie des connaissances** et **Connaissances documentées**, ainsi que deux nouveaux types de modèles, **Diagramme de structure des connaissances** et **Carte des connaissances**. De plus, nous avons ajouté aux types de modèles pour la représentation des processus d'entreprise (CPE, etc.) des structures pour l'acquisition et l'utilisation des connaissances. Les nouveaux types d'objets et de modèles ont été intégrés entièrement et méthodiquement dans les autres types de modèles importants des règles de gestion (MERe, organigramme, arbre de fonctions) pour garantir une vue approfondie. Il est ainsi possible d'intégrer des modèles d'un projet d'optimisation des processus d'entreprise pour l'analyse et l'amélioration du traitement des connaissances.

Le diagramme de structure des connaissances se trouve dans la vue des données des règles de gestion. La carte des connaissances, ainsi que les types de modèles étendus pour la modélisation de processus d'entreprise, se trouvent dans la vue Gestion des règles de gestion.

5.2 Types d'objets pour la modélisation du traitement des connaissances

5.2.1 Catégorie des connaissances

Le type d'objet **Catégorie des connaissances**, illustré par une bulle ovale (voir la figure **Diagramme de structure des connaissances** (page 124), représente un contenu auquel se rapportent certaines connaissances. Des exemples de **catégories de connaissances** sont les suivants : connaissances sur la gestion de projets, connaissances sur un secteur en particulier, connaissances sur une technique particulière, connaissances sur la clientèle et sur la concurrence, etc. De telles catégories permettent de répartir les connaissances existantes ou nécessaires dans l'entreprise.

Les connaissances réparties dans une catégorie peuvent être des connaissances implicites, c'est-à-dire des connaissances que l'on ne peut pas documenter entièrement ou des connaissances existantes concernant les capacités des collaborateurs ou des groupes ; il peut également s'agir de connaissances explicites documentées sous forme de description ou de dessin technique. En règle générale, les catégories des connaissances contiennent les deux. Ainsi, les connaissances sur la gestion de projet peuvent englober aussi bien les expériences des chefs de projet que des remarques d'un manuel sur la gestion de projet.

Les attributs spécifiques ci-dessous permettent une description plus précise des catégories des connaissances (outre les attributs généraux tels que Description, Remarque, Source, etc.) :

Nom d'attribut	Plage de valeurs	Description/Exemple
Fréquence de mises à jour	Type d'énumération : horaire, quotidienne, hebdomadaire, mensuelle, annuelle, rare, jamais	La fréquence de mises à jour indique la fréquence requise de renouvellement des connaissances d'une certaine catégorie, afin de disposer de connaissances actuelles. Ainsi, les connaissances de base en trigonométrie ne devront être renouvelées que rarement (à des fins pratiques) ou jamais ; en revanche, les connaissances sur les cours de la bourse devront être renouvelées chaque jour ou même chaque heure.
Signification	Chiffre en pourcentage : 0..100	L'importance de la catégorie des connaissances dans une entreprise peut être indiquée en pourcentage (0% pour aucune importance, 100% pour une extrême importance).

Nom d'attribut	Plage de valeurs	Description/Exemple
Degré de couvrement	Chiffre en pourcentage : 0..100	Vous pouvez indiquer un degré de couvrement pour les connaissances de votre entreprise situé entre 0% (pas couvert) et 100% (taux de couvrement maximal). Pour représenter le taux de couvrement de la catégorie des connaissances par une unité organisationnelle ou une personne, vous pouvez ajouter un attribut de la liaison du type **dispose de** dans une carte des connaissances.
Avance des connaissances	Chiffre en pourcentage : 0..100	L'avance relative des connaissances par rapport à la concurrence peut être indiquée entre 0% (la concurrence dispose d'une avance maximale des connaissances) et 100% (votre entreprise dispose de l'avance maximale des connaissances).
Utilisation des connaissances	Chiffre en pourcentage : 0..100	Vous pouvez indiquer un degré d'utilisation des connaissances d'une catégorie particulière de connaissances situé entre 0% (ces connaissances ne sont absolument pas utilisées) et 100% (ces connaissances sont utilisées au maximum).
Degré de couvrement souhaité	Chiffre en pourcentage : 0..100	Le degré de couvrement souhaité pour les connaissances de votre entreprise peut varier entre 0% (pas couvert) et 100% (taux de couvrement maximal).
Importance future	Type d'énumération : fortement décroissant, décroissant, restant constant, croissant, fortement croissant	L'**importance future** représente la tendance de modification prévue pour la signification d'une catégorie des connaissances pour l'entreprise.
Structurel Rapidité de changement	Chiffre en pourcentage : 0..100	La rapidité de changement structurel indique la rapidité à laquelle les méthodes appliquées à l'acquisition des connaissances doivent changer (0% : aucun changement ; 100% : rapidité de changement maximale).

Ces attributs permettent d'évaluer une **catégorie des connaissances** par rapport à votre entreprise. Vous pouvez donc les utiliser comme base pour l'identification des mesures urgentes et importantes afin d'améliorer la gestion des connaissances de votre entreprise. Il est souvent utile de représenter ces types de valeurs sous forme de graphique. Vous pouvez tout simplement utiliser la fonction Copier-coller de la fenêtre **Attributs** et les insérer dans un tableur pour y créer les modèles correspondants. Vous pouvez ainsi par exemple comparer le **degré de couvrement** actuel et souhaité sous forme de graphique en barres pour les **catégories des connaissances** souhaitées.

5.2.2 Connaissances documentées

Contrairement au type d'objet **Catégorie des connaissances** qui peut englober des connaissances implicites et explicites, le type d'objet **Connaissances documentées** n'englobe que les **catégories des connaissances** documentées explicitement ou les catégories que l'on pourrait en principe documenter. C'est le cas, par exemple, de connaissances documentées sur l'utilisation d'un logiciel dans un manuel. La distinction faite entre les catégories des connaissances générales et documentées permet de répartir les connaissances en catégories ; en effet, cette distinction permet d'identifier les possibilités et les limites d'un support de système d'information du traitement des connaissances, puisque seules les connaissances documentées peuvent être enregistrées, transmises et traitées sous forme électronique.

Le type d'objet **Connaissances documentées** est représenté par une bulle rectangulaire. Il contient les mêmes types d'attributs spécifiques que le type d'objet **Catégorie des connaissances** (page 120).

5.3 Types de modèles pour la modélisation du traitement des connaissances

5.3.1 Diagramme de structure des connaissances

Un diagramme de structure des connaissances permet de subdiviser le contenu des catégories des connaissances. La figure suivante illustre cette possibilité. Une catégorie des connaissances peut englober aussi bien d'autres catégories de connaissances que des connaissances documentées. Les connaissances documentées peuvent être subdivisées en plusieurs sous-catégories de connaissances documentées. Elles ne peuvent cependant pas englober de catégories de connaissances générales.

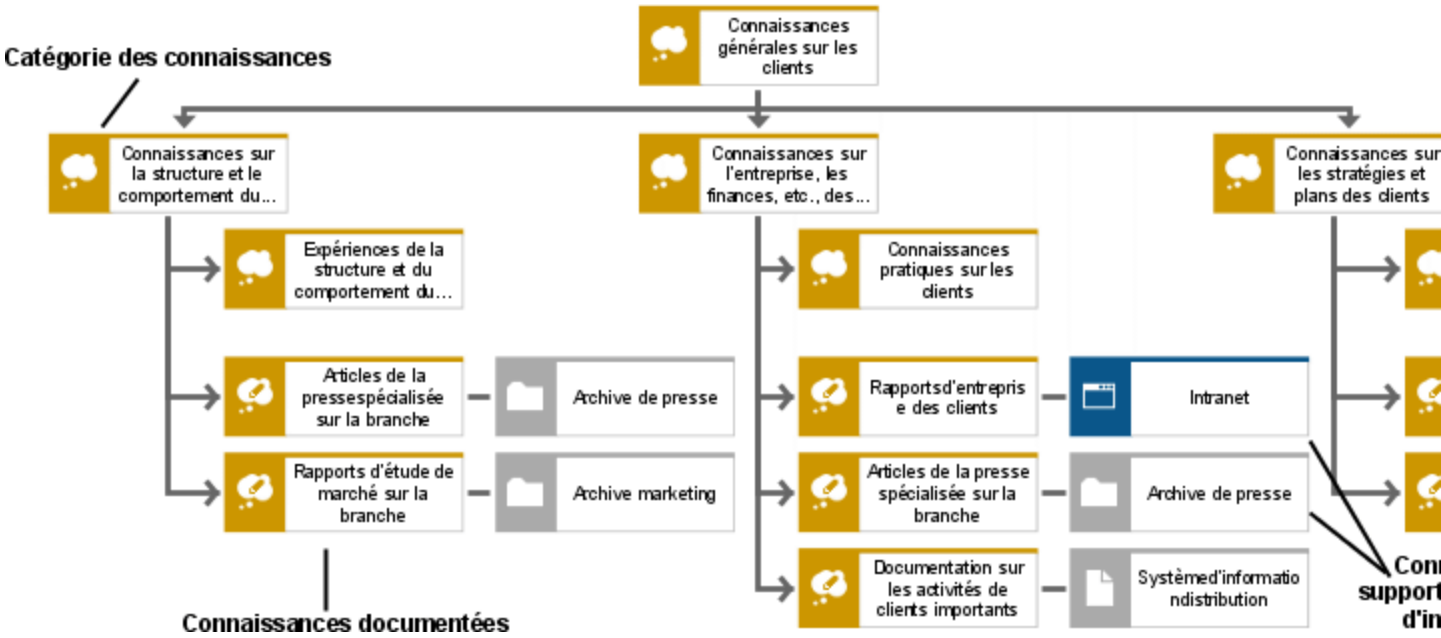


Illustration 117: Diagramme de structure des connaissances

Pour les connaissances documentées, le diagramme de structure des connaissances permet de représenter les supports d'informations sur lesquels les connaissances sont documentées, ou encore de représenter les objets d'information d'un modèle de données ou les classes d'un système orienté objet permettant la documentation de ces connaissances. Vous pouvez enfin sélectionner les types ou les classes d'applications qui permettront la gestion des connaissances documentées.

5.3.2 Carte des connaissances

Une carte des connaissances permet de représenter la répartition organisationnelle des différentes catégories des connaissances. A cet effet, vous pouvez utiliser des liaisons du type **dispose de** pour relier différents types d'objets de la vue organisationnelle (p. ex. unité organisationnelle, poste, personne, site, groupe) à des catégories des connaissances. Une personne ou une unité organisationnelle dispose des connaissances d'une catégorie ; il est possible d'indiquer pour cela un degré de couvrement. La liaison **dispose de** contient l'attribut **Degré de couvrement** qui permet d'indiquer en pourcentage le degré de couvrement des

connaissances de la catégorie sélectionnée par l'unité organisationnelle. Une valeur de 100% indique un couvrement maximal et une valeur de 0% indique qu'il n'existe aucune connaissance pour la catégorie. Cela revient à indiquer que la liaison est manquante. Outre cette évaluation quantitative, évaluation qualitative pouvant être effectuée et affichée sous forme de graphique. Vous pouvez utiliser à cet effet la liaison **Couvrement qualitatif**, qui accepte les valeurs **faible**, **moyen**, **élevé** et **maximal**. Ces indications peuvent être visualisées par des symboles graphiques sur les liaisons, comme l'illustre la figure suivante. Il n'existe aucun rapport direct entre les valeurs des attributs **Degré de couvrement** et **Couvrement qualitatif**. Si vous souhaitez les deux attributs ensemble, nous vous conseillons d'utiliser la qualification **faible** pour un degré de couvrement allant jusqu'à 25%, **moyen** de 26% à 50%, **élévé** de 51 à 75% et **maximal** de 76% à 100%.

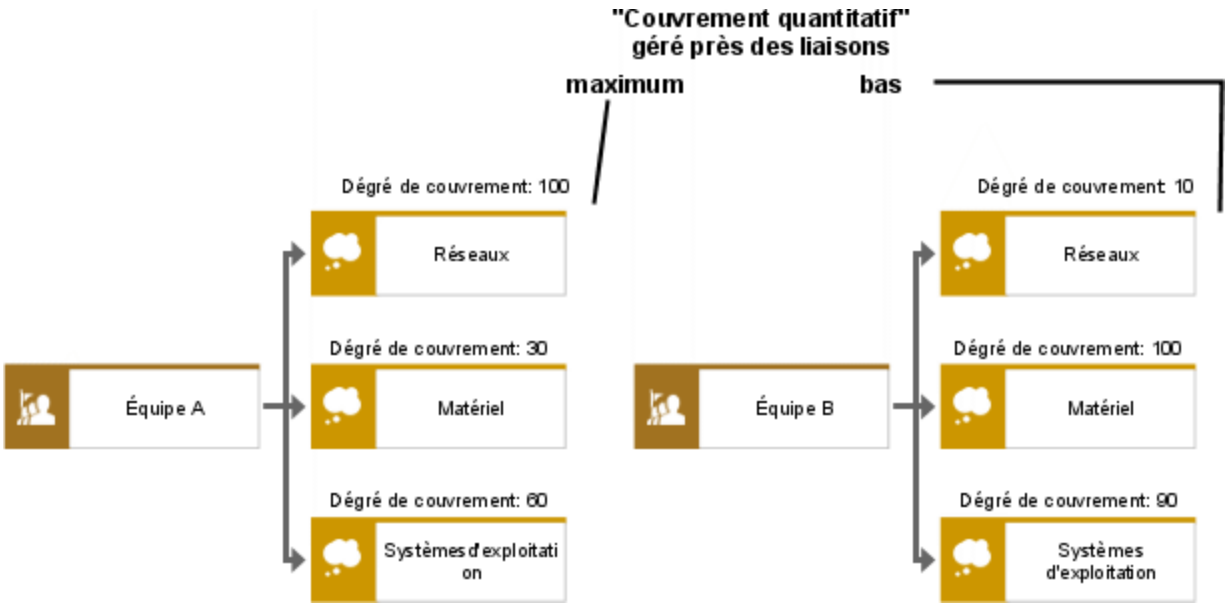


Illustration 118: Carte des connaissances - orientée unité organisationnelle

La carte des connaissances de cette figure est organisée de manière orientée unité organisationnelle, c'est-à-dire que toutes les **catégories des connaissances** pertinentes sont indiquées pour chaque unité organisationnelle. Vous pouvez également sélectionner les **unités des connaissances** comme objets centraux puis y ajouter toutes les unités organisationnelles pertinentes. Les possibilités de navigation d'ARIS (onglet **Relations** du dialogue **Propriétés - Objet**) permettent, dans les deux cas, de trouver simplement les autres liaisons d'une unité organisationnelle ou d'une catégorie des connaissances. Une représentation matrice est souvent utilisée pour les cartes des connaissances. Vous pouvez l'obtenir en triant plusieurs valeurs d'une catégorie des connaissances sous forme de colonnes, comme l'illustre la figure suivante. Dans cet exemple, le nom n'est affiché que pour les **catégories des connaissances** indiquées en haut (comme dans l'en-tête d'une table). Pour les autres valeurs, les noms ont été supprimés à l'aide de la fonction Placement des attributs. En outre, un type de représentation visuelle est proposé alternativement pour les différents degrés de couvrement : les **catégories des connaissances** sont de différentes tailles.

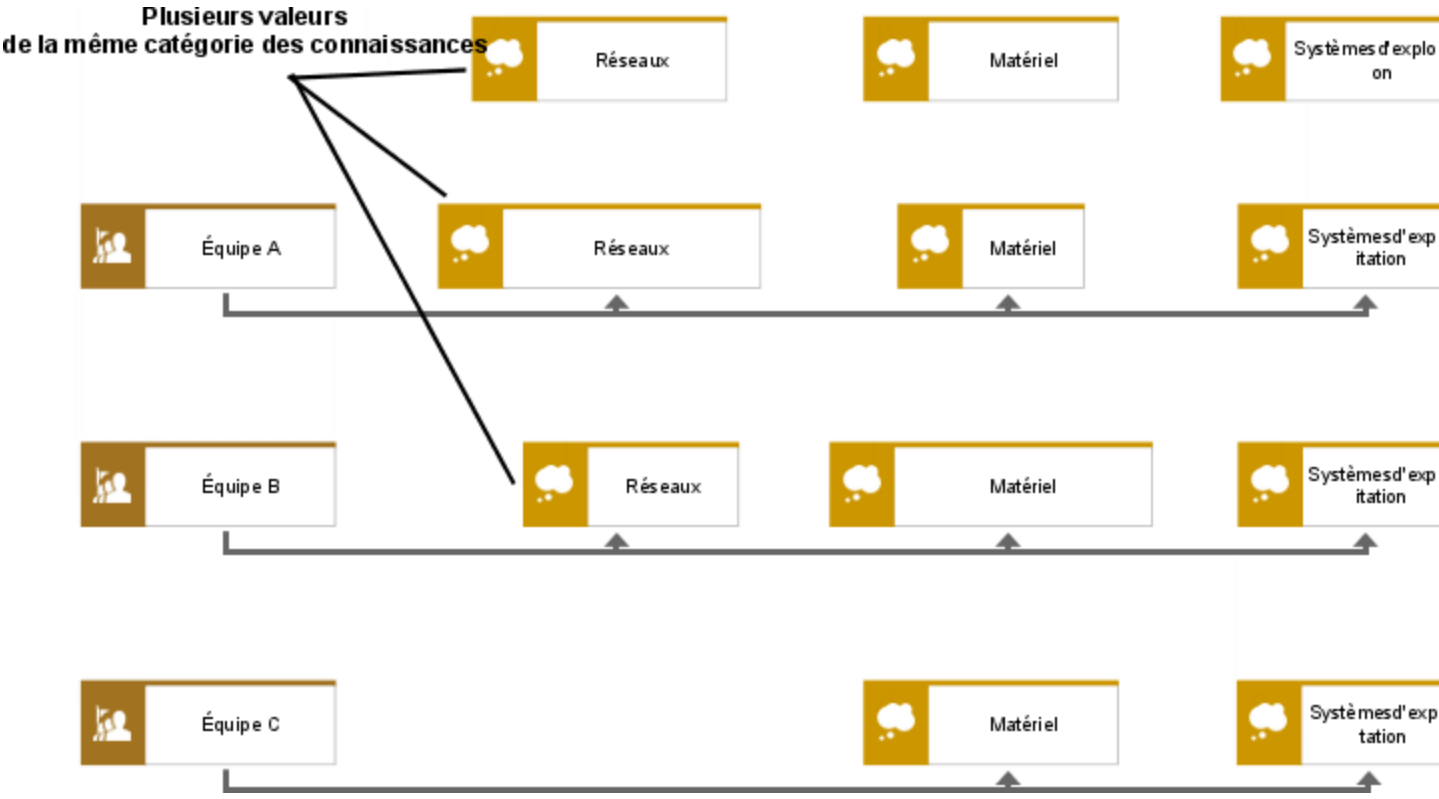


Illustration 119: Carte des connaissances - représentation matrice

5.3.3 Représentation du traitement des connaissances dans les processus d'entreprise

L'utilisation et la création de connaissances dans les processus de l'entreprise sont modélisées dans les types de modèles pour la représentation de processus d'entreprise (CPE,

CPE (avec flux de matières)). Ces types de modèles contiennent également les types d'objets **Catégorie des connaissances** et **Connaissances documentées**. Pour une fonction, vous pouvez indiquer d'une part les connaissances (générales ou documentées) requises pour votre exécution, d'autre part les connaissances créées ou documentées lors de l'exécution. Une telle représentation permet d'analyser les processus d'entreprise à travers le traitement des connaissances. Vous pouvez ainsi, par exemple, déceler les besoins en connaissances non couverts. De même, vous pouvez p. ex. déterminer le profil de qualification requis pour l'exécution d'une fonction.

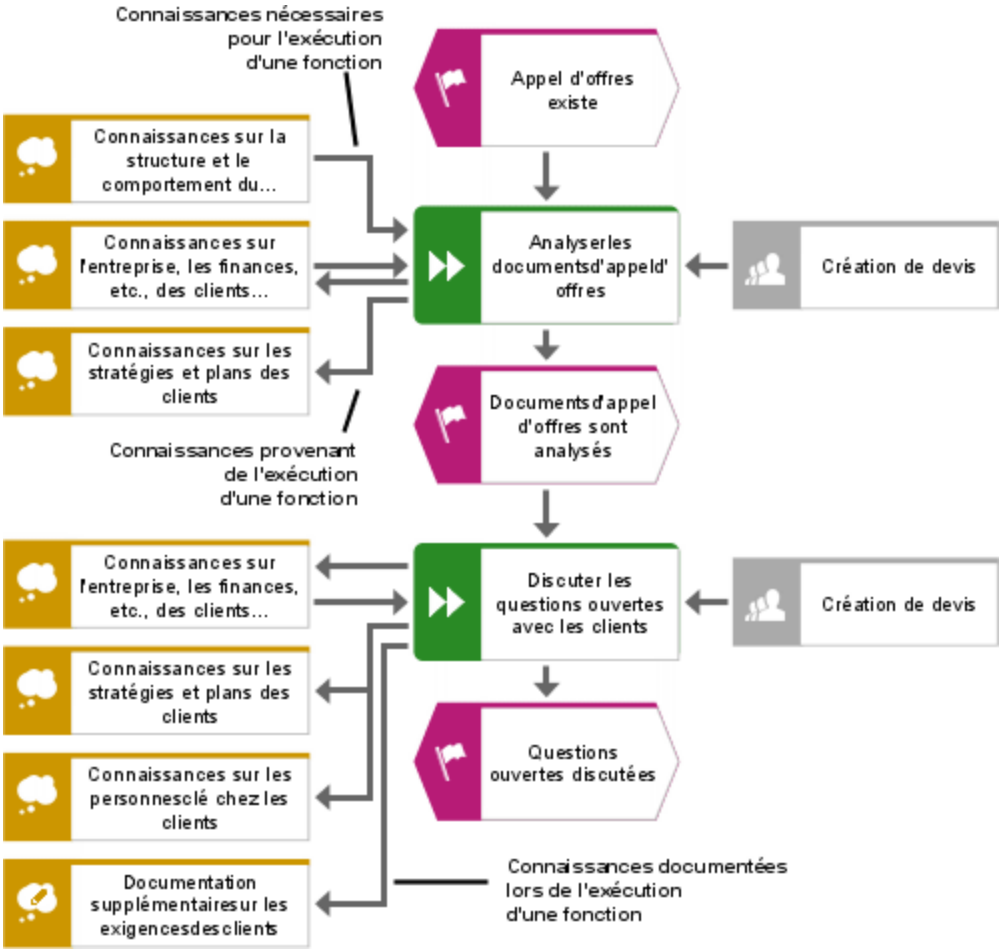


Illustration 120: Traitement des connaissances dans une CPE

6 Méthode du tableau de bord prospectif

6.1 Introduction

Pour pouvoir réussir dans l'environnement turbulent de l'entreprise et dans la concurrence mondiale des marchés actuels, les entreprises doivent disposer de processus optimisés. Leur capacité à réagir rapidement et par rapport à leurs objectifs aux nouveaux développements dans l'environnement de l'entreprise est également primordiale. Pour ce faire, il est indispensable de recourir à des processus de gestion efficaces, permettant une mise en oeuvre cohérente des stratégies de l'entreprise et de ses objectifs. Ces processus de gestion permettent également la réalisation au quotidien des objectifs grâce à des initiatives opérationnelles.

De nombreuses approches classiques de la gestion ne créent pas de relation entre la formulation des stratégies d'entreprise et leur conversion par des initiatives orientées stratégie et par un contrôle conséquent de l'atteinte de l'objectif des objectifs stratégiques.

En outre, de nombreuses entreprises sont encore gérées avec de purs indicateurs financiers dont la signification est relative, car ils sont encore en majorité orientés vers le passé et ne peuvent donc livrer que peu d'informations de gestion significatives pour l'avenir. Seule la prise en compte des causes de la réussite financière (client, processus, innovations, etc.) permet de reconnaître précocement les divergences par rapport aux objectifs stratégiques fixés.

L'approche du tableau de bord prospectif met à disposition une méthodique structurée clairement et simple à implémenter avec laquelle ces faiblesses peuvent être compensées.

6.2 Concept du tableau de bord prospectif

6.2.1 Bases de l'approche TBP

L'approche du tableau de bord prospectif est un système de gestion stratégique présenté pour la première fois en 1992 par Robert Kaplan et David Norton (The Balanced Scorecard-Measures that drive Performance, Harvard Business Review janvier/février 1992). Il a été créé grâce au développement des résultats d'une étude sur le sujet **Approches de la mesure du rendement**. Dans le cadre de cette étude, il a été découvert que les systèmes de mesure du rendement uniquement orientés indicateurs monétaires de nombreuses entreprises empêchent plutôt les activités créatrices de valeurs. A partir de là, Kaplan et Norton ont essayé d'élaborer, en collaboration avec des entreprises innovatrices, un système d'indicateurs qui devrait permettre de mesurer de manière optimale la conversion des visions et stratégies des entreprises.

Pour l'approche du tableau de bord prospectif, des indicateurs provenant de différents angles de vue sont affectés à l'entreprise (également nommés perspectives). Ces dernières sont composées aussi bien de perspectives de rendement internes (par ex. les perspectives d'apprentissage et de développement, les perspectives de processus), qu'externes (par ex. les perspectives de clients, perspectives financière). Cette disposition des indicateurs permet d'atteindre un certain équilibre quant aux objectifs à court ou à long terme, aux indicateurs monétaires et non-monétaires, aux indicateurs précoces et retardés, ainsi qu'aux angles de vue internes et externes. Avec l'utilisation des indicateurs spécifiques à la branche, un composant d'évaluation des performances est également intégré au processus.

A partir de la pure approche de mesure du rendement se développe un système de gestion complet, contenant une gestion d'entreprise orientée objectif, partant de la vision de l'entreprise et allant jusqu'à la formulation et au contrôle des initiatives grâce à des indicateurs équilibrés, en passant par les différentes stratégies de concurrence. L'approche du tableau de bord prospectif est donc plus qu'un simple système d'indicateurs pour la mesure du rendement. Elle assiste les entreprises lors de la communication et la conversion de leur stratégie et du processus d'apprentissage stratégique en découlant (apprentissage en double boucle).

6.2.2 Processus de gestion stratégique et tableau de bord prospectif

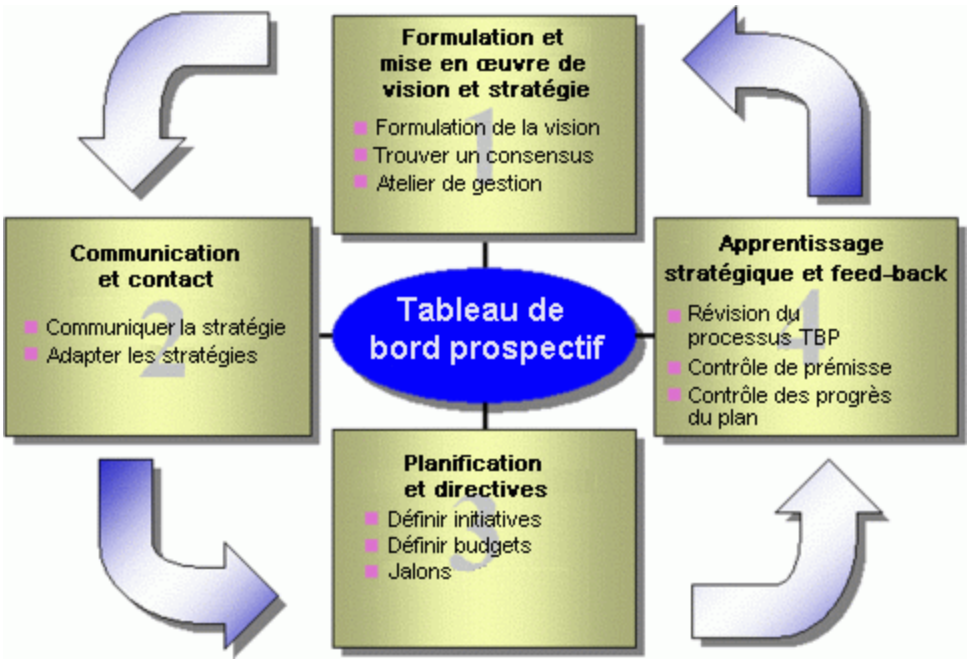


Illustration 121: Le TBP comme cadre d'action pour la gestion stratégique

Le processus de gestion stratégique dans le cadre de l'approche du tableau de bord prospectif se compose de deux phases :

1. Dans la première phase, la stratégie de l'entreprise doit être découverte dans le cadre d'une analyse stratégique. Dans le cadre de l'analyse stratégique, toutes les informations importantes pour la concurrence sont saisies. L'objectif de l'analyse est d'identifier et d'évaluer les tendances, chances et risques du développement de l'environnement de l'entreprise et de saisir des compétences centrales propres à l'entreprise. A la fin de cette phase, la stratégie individuelle de l'entreprise est définie.
2. La conversion de la stratégie de l'entreprise a lieu lors de la seconde phase. Pour cette phase, la méthode TBP offre un potentiel de support. La stratégie de l'entreprise peut être affinée grâce à des stratégies partielles (par ex. des stratégies spécifiques au domaine commercial). D'autres objectifs stratégiques en découlent. Les objectifs stratégiques sont concrétisés grâce aux directives pour des tailles de mesures déterminées. Un programme d'action sous forme d'instructions permet de déterminer comment atteindre les objectifs. Le programme d'action est appliqué dans la planification de l'entreprise à différents domaines ou services de l'entreprise. Une budgétisation permet de concrétiser davantage les actions. A l'aide de différents tableaux de bord stratégiques, l'atteinte de l'objectif est mesurée à l'aide des indicateurs générés. A partir des tableaux de bord définis, un processus de révision est exécuté, dans lequel d'autres initiatives sont déterminées sur la base des divergences possibles ou dans lequel la stratégie est "revue".

6.2.2.1 Formuler et traduire la vision et la stratégie

La stratégie d'une entreprise découle de la vision de l'entreprise.

La vision représente une conception directrice pour l'entreprise observée. Elle n'est pas très concrète en tant que formulation marquante et globale, mais est valable comme maxime d'action supérieure et représente donc une sorte de résumé de la philosophie de l'entreprise.

Pour les différentes unités commerciales stratégiques d'une entreprise, les stratégies correspondantes sont déterminées. Ces stratégies doivent s'orienter à l'objectif de réussite de l'entreprise. Il est donc nécessaire, au début de l'introduction d'un TBP, que les cadres supérieurs de l'entreprise déterminent dans un groupe de travail la vision et les stratégies qui en découlent pour les unités commerciales stratégiques. Dans la plupart des cas, le résultat sera une définition d'objectif financier (perspective financière). Différentes options peuvent être accentuées : les objectifs possibles sont des valeurs déterminées pour les rentabilités du capital, ROI, Shareholder value, produits du chiffre d'affaires ou flux de liquidités. Les objectifs financiers doivent être réalisés grâce à un comportement spécifique sur le marché à traiter (perspective de clientèle). C'est la raison pour laquelle les segments clientèle et marché correspondants sont choisis après la définition de l'objectif. Des objectifs stratégiques sont également définis pour cette perspective de clients et les indicateurs pertinents sont saisis. Vous pouvez par ex. vous référer à des parts de marché ou des taux de croissance de certains segments clients.

Pour réaliser des stratégies orientées marché spécifiques, les ressources d'entreprise correspondantes sont nécessaires. Lors de la définition d'un tableau de bord prospectif, les ressources sont saisies de deux manières :

1. Après la définition des objectifs financiers et clients, les processus d'entreprise (perspective de processus) importants sont considérés, des définitions d'objectifs déterminées et des initiatives et des indicateurs générés. Au centre de l'observation, on retrouve en général les temps et les coûts de processus.
2. Dans le cadre de la perspective d'apprentissage et de développement, les objectifs stratégiques pour le développement du personnel, les techniques d'informations et les innovations découlent des directives d'objectifs stratégiques de la perspective financière, de clients et de processus.

Lors de la définition d'un tableau de bord prospectif, tous les objectifs stratégiques définis se retrouvent dans une structure d'effet réciproque qui est décrite comme chaîne de cause à effet.

Pour cette première étape, il est important qu'un accord soit trouvé au niveau des cadres supérieurs quant à la vision et aux stratégies et objectifs stratégiques en découlant.

6.2.2.2 Perspectives standard d'un tableau de bord prospectif

Kaplan et Norton proposent quatre perspectives standard pour l'élaboration d'un tableau de bord prospectif :

1. Perspective financière - Attentes des actionnaires : "Quel est l'impact de la stratégie sur les finances ?"
2. Perspective de clients - Attentes des clients : "Comment nous positionnons-nous sur les marchés cibles ?"
3. Perspective de processus - Exigences envers les processus : "Quels processus sont importants d'un point de vue stratégique ?"

4. Perspective d'apprentissage - Exigences envers l'apprentissage organisateur et les innovations : "Comment devenons-nous une organisation apprenante ? "Comment développons-nous la croissance ?"

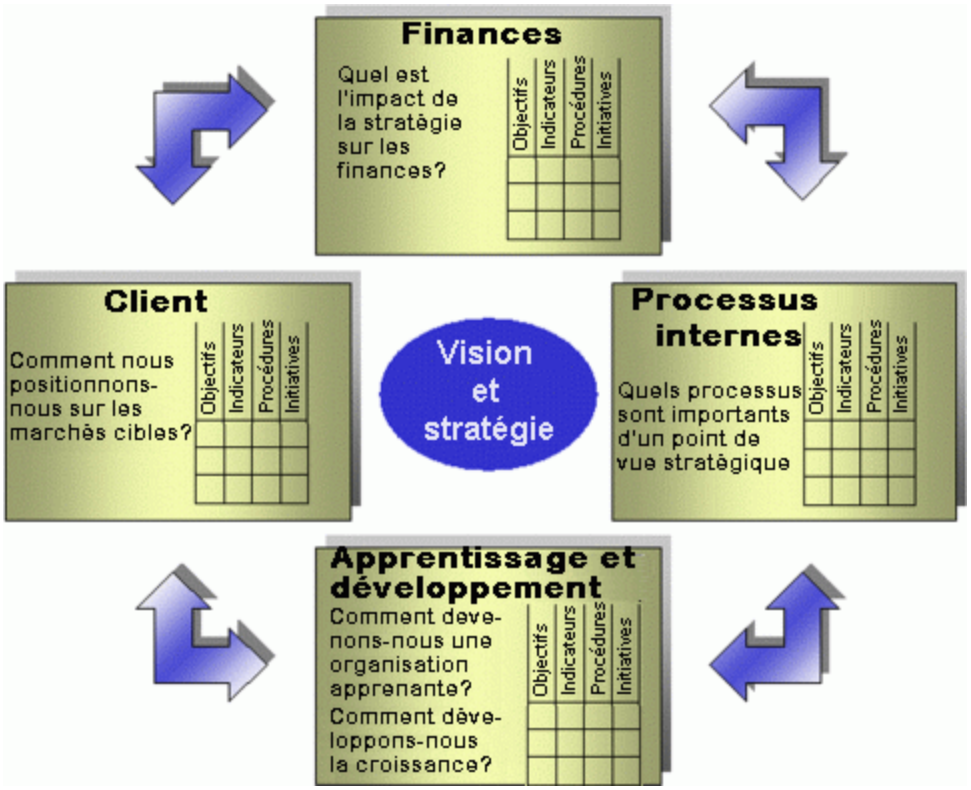


Illustration 122: Les perspectives du TBP

Les perspectives standard constituent une logique implicite, utile lors de la conversion des stratégies et de la formulation de relations de cause à effet concrètes. En outre, il existe la possibilité de définir pour une entreprise d'autres perspectives importantes (par ex. perspective environnement), pertinentes pour la stratégie et dont les objectifs et les indicateurs peuvent être en relation avec les autres perspectives.

6.2.2.3 Chaîne de cause à effet

La chaîne de cause à effet explicite les relations de cause à effet entre les différents objectifs partiels d'une stratégie, au-delà des perspectives définies des indicateurs. Cela signifie qu'elle décrit la manière dont les objectifs des perspectives peuvent être atteints. Les perspectives elles-mêmes constituent dans ce cas les conditions cadre pour la chaîne de cause à effet ("Champ de force de l'entreprise"). On part du principe que les améliorations dans l'environnement des perspectives d'apprentissage et de développement possèdent un effet positif sur les objectifs et les indicateurs des perspectives de processus internes. En outre, les développements de la perspective de processus ont aussi un effet positif sur les objectifs et

les indicateurs de la perspective de clients, ce qui entraîne également une amélioration des objectifs financiers.

Pendant l'élaboration d'un système de tableau de bord prospectif à l'échelle de l'entreprise, les objectifs et les indicateurs sont définis pour chaque stratégie et leur relations d'effets sont représentées dans les différentes perspectives et au-delà des perspectives. Les objectifs et les indicateurs de la perspective financière servent de point de départ et des "responsables positifs" (moteurs de productivité) correspondants sont recherchés dans la perspective de clients. Ceux-ci sont ensuite adaptés de la perspective de processus jusqu'à la perspective d'apprentissage et de développement. Cette procédure permet de diviser les stratégies d'une entreprise ou d'une division en objectifs partiels et indicateurs et également d'en faire découler des initiatives opérationnelles.

Un équilibre entre les indicateurs précoces et retardés découle implicitement de la définition de la chaîne de cause à effet et de la définition des indicateurs.

6.2.2.4 Définition des indicateurs précoces et retardés

L'élaboration de chaînes de cause à effet permet également de déterminer un rapport temporel entre les différents objectifs et indicateurs. Les perspectives **Perspective d'apprentissage et de développement** et **Perspective de processus** appartiennent en général aux moteurs de productivité (indicateurs précoces) qui précèdent temporellement les effets de la perspective de clients et de la perspective financière. Dans la perspective de clients et financière, la majorité des indicateurs sont des indicateurs de résultats qui mesurent la réussite commerciale. Ils sont orientés vers le passé et sont nommés des indicateurs retardés.

Dans chaque perspective du tableau de bord de processus, des indicateurs aussi bien précoces que retardés doivent être définis afin que le rapport entre les initiatives et l'atteinte des objectifs puisse être représenté. Les indicateurs précoces permettent de reconnaître suffisamment tôt des divergences par rapport à l'objectif.

6.2.2.5 Communication et déduction d'autres tableaux de bord prospectifs

L'efficacité d'une stratégie d'entreprise et sa conversion grâce au concept du tableau de bord prospectif dépend, entre autres, de la notoriété et de l'acceptation du tableau de bord au sein de l'entreprise. Pour cette raison, il est extrêmement important que les stratégies et les tableaux de bord de l'entreprise soient rendus publics par un programme de communication étendu à tous ses niveaux hiérarchiques.

A partir de la stratégie d'entreprise supérieure (tableau prospectif de l'entreprise), des directives d'objectifs sont déduites à l'aide d'une approche du haut vers le bas pour les niveaux hiérarchiques inférieurs de l'entreprise et les objectifs stratégiques supérieurs sont adaptés aux objectifs spécifiques aux domaines. Pour ces derniers, des indicateurs mesurant le degré d'atteinte de l'objectif sont également générés. Afin d'atteindre les objectifs, différentes initiatives sont nécessaires dans les domaines de fonction d'une entreprise. Ces initiatives sont également saisies dans les tableaux de bord des niveaux hiérarchiques inférieurs. Toutes les directives d'objectifs doivent être énoncées sur la base de réflexions stratégiques à long terme. Elles ne doivent pas s'orienter aux résultats à court terme qui ne sont pas en harmonie avec la stratégie supérieure. Une réduction de coûts basée sur une courte augmentation de l'intensité de la production constituerait un exemple de résultat à court terme.

6.2.2.6 Planification et directives

Le TBP peut être utilisé pour connecter le processus de conversion de la stratégie au processus de budgétisation sur les différents niveaux hiérarchiques de l'entreprise. Le but du couplage est d'orienter toutes les ressources sur la stratégie de l'entreprise.

Le TBP est intégré en quatre étapes dans le processus de planification et de budgétisation stratégique à long terme.

1. Des directives d'objectifs sont énoncées. Elles représentent un défi, sont communiquées au sein de l'entreprise et acceptées par les collaborateurs. Les identificateurs pertinents se rapportant aux valeurs des résultats de la perspective de clients et financière sont identifiés grâce aux relations de cause à effet entre les indicateurs.
2. Les initiatives stratégiques doivent commencer exactement à l'indicateur dont la valeur actuelle se différencie le plus clairement des valeurs des directives. Cette procédure permet de lier de manière conséquente et à long terme les investissements de capitaux ainsi que les programmes d'action à des valeurs d'objectif importantes d'un point de vue stratégique.

3. Les initiatives essentielles au niveau de l'entreprise sont identifiées afin que des effets synergiques puissent être réalisés avec les objectifs stratégiques.
4. Les valeurs prévues pour les indicateurs de résultats sont connectées à la planification de l'entreprise grâce à un couplage aux directives budgétaires comportant différents délais (plan sur 5, 3 ou 1 an(s)).

6.2.2.7 Apprentissage stratégique et feed-back

Le concept du tableau de bord prospectif contribue à ce qu'un apprentissage stratégique au sein de l'entreprise ait lieu et qu'un feed-back concernant la mise en œuvre de la stratégie soit donné.

L'apprentissage stratégique se rapporte à chaque collaborateur ainsi qu'à l'ensemble de l'entreprise comme organisation apprenante. Les collaborateurs doivent connaître la stratégie et veiller eux-mêmes à y adapter leurs actions. Des données de productivité, grâce auxquelles est mesurée la mise en œuvre de la stratégie, sont collectées via un processus de feed-back. Un contrôle de prémisse a ainsi lieu : des hypothèses se rapportant aux relations préalablement définies de cause à effet des objectifs stratégiques sont vérifiées.

Un apprentissage en double boucle survient lorsque les directives d'objectifs stratégiques sont soumises à une révision critique dans un processus du bas vers le haut, et qu'ensuite sont générées de nouvelles relations de cause à effet entraînant une poursuite efficace de la stratégie fondée sur une base d'informations s'élargissant continuellement.

Le feed-back prend alors une grande importance : il représente le processus qui ne se plie pas à l'adaptation des TBP. Le feed-back ne contient pas seulement des réactions provenant d'indicateurs sous forme de chiffres de progression prévus, mais aussi des propositions pour d'autres initiatives ainsi que de nouveaux rapports d'effet. Il indique ainsi, le cas échéant, des incohérences éventuelles dans la poursuite de la stratégie.

6.2.3 Avantages et utilité du tableau de bord prospectif

L'avantage de la méthode du tableau de bord prospectif se trouve dans la gestion d'entreprise orientée stratégie et conséquente, englobant les niveaux inférieurs de l'entreprise et le service d'initiatives opérationnelles conforme à la stratégie sur ces niveaux de l'entreprise. Toutes les ressources de l'entreprise et tous les collaborateurs sont orientés sur la stratégie de l'entreprise. Cette dernière est communiquée et contrôlée de manière conséquente grâce aux tableaux de bord prospectifs.

La méthode du tableau de bord prospectif ne possède pas les désavantages des systèmes d'indicateurs classiques pour la gestion d'entreprise :

- Pas de limitation à des indicateurs financiers
- Pas d'orientation pure vers le passé
- Prise en compte de données également qualitatives
- Réduction de complexité et consensus
- Objectifs stratégiques comme point de départ de l'approche du tableau de bord prospectif

- Orientation aux goulets d'étranglement des cadres supérieurs et non aux données existantes
- Accélération des modifications nécessaires

La méthode du tableau de bord prospectif est particulièrement indiquée pour la gestion d'entreprise orientée stratégie car elle permet de surmonter les barrières suivantes :

- Barrière de concrétisation : pour les approches de gestion classiques, la vision et la stratégie sont des formulations de souhaits et le service de formulation d'action concrète (opérationnelle) reste en suspens.
- Barrière de vision : la stratégie n'est pas toujours comprise des collaborateurs qui doivent l'appliquer.
- Barrière d'engagement : les stratégies ne sont pas reliées aux objectifs des services ou des individus.
- Barrière d'implémentation : le rapport est orienté vers des objectifs opérationnels monétaires plutôt que vers des objectifs stratégiques.
- Barrière opérationnelle : le processus de budgétisation est séparé du processus de planification stratégique.

6.3 Développement d'un tableau de bord prospectif avec ARIS BSC

6.3.1 Concepts et abréviations

Vision : une vision indique la direction d'activité stratégique à venir et la mission d'une entreprise et est réalisée par différentes stratégies.

Stratégie : des stratégies sont développées à partir de la vision. Les stratégies peuvent être divisées en stratégies partielles spécifiques.

Objectif stratégique : chaque stratégie est composée d'un certain nombre d'objectifs stratégiques. Les objectifs stratégiques sont généralement définis par les cadres supérieurs responsables lors de groupes de travail. Une relation de cause à effet et une logique temporelle existent entre les objectifs.

Facteur de succès : un facteur de succès constitue, pour le déroulement des affaires d'une entreprise, un facteur d'influence essentiel pour le succès. Ce dernier peut se trouver dans une structure d'effet par rapport à d'autres facteurs de succès et les influencer dans une certaine direction d'effet avec une certaine efficacité.

Vue/Perspective : une perspective est la concrétisation d'une vue déterminée de l'entreprise. Fondamentalement, la sélection de perspectives devrait comporter les angles de vues suivants sur une entreprise : vue orientée vers les qualités humaines, vue interne, vue orientée processus et vue externe.

Indicateur : des indicateurs pour la mesure de la productivité et l'atteinte de l'objectif sont affectés à tous les objectifs ou facteurs de succès stratégiques.

Valeur réelle : la valeur actuelle que peut prendre un indicateur, un objectif stratégique ou un facteur de succès. Le degré d'atteinte de l'objectif actuel peut être dérivé de la valeur réelle par une comparaison budget/réalisé.

Valeur prévue : la valeur prévue et devant être atteinte pendant cette période pour un indicateur, un objectif ou un facteur de succès. En général, pour les objectifs stratégiques, il s'agit d'un pourcentage qui découle des atteintes d'objectif pondérées des différents indicateurs. La valeur prévue peut être adaptée à différentes périodes et peut également prendre en compte des fluctuations périodiques. La valeur prévue correspond à un degré d'atteinte de l'objectif de 100 %.

Valeur cible : la valeur prévue et devant être atteinte pour une période cible future. En général, elle devient la valeur prévue lorsque la période cible est atteinte.

Valeur minimale : la plus petite valeur que peut prendre un objectif, un facteur de succès essentiel ou un indicateur. Dans ARIS, la valeur minimale 0 est prédéfinie. Elle peut être modifiée.

Valeur maximale : la valeur la plus importante (limite supérieure) que peut prendre un objectif stratégique, un facteur de succès ou un indicateur. En général, cette valeur est utilisée dans ARIS uniquement pour standardiser et rendre possible la comparaison de la représentation de plusieurs indicateurs.

Seuil d'avertissement : le seuil d'avertissement correspond à la valeur prévue, cela signifie que la valeur prévue correspond au seuil en dessous duquel un indicateur, un objectif stratégique ou un facteur de succès ne respectent plus la directive souhaitée.

Marge de tolérance : la marge de tolérance est une divergence négative de la valeur prévue, qui est encore considérée comme tolérable pour un objectif stratégique, un indicateur ou facteur de succès.

Seuil d'alarme : le seuil d'alarme correspond à la valeur prévue moins la marge de tolérance. Toutes les valeurs d'un objectif stratégique, d'un indicateur ou d'un facteur de succès situées sous le seuil d'alarme ne sont plus tolérées.

Initiative : une initiative se répercute sur un objectif ou sur plusieurs objectifs stratégiques. Les initiatives ont en général un responsable, un temps de départ et un temps de fin, des ressources, etc.

Type d'indicateur : un indicateur peut être du type **Indicateur précoce** ou **Indicateur retardé**. Un indicateur précoce mesure un pilote de prestation et est tourné vers le futur. Un indicateur retardé mesure un résultat (indicateur de résultat) et est rétrospectif. Les objectifs financiers sont en général des indicateurs retardés, les objectifs des perspectives de processus, d'apprentissage et de développement sont surtout des indicateurs précoces. En outre, les indicateurs retardés doivent également être saisis dans les perspectives afin de représenter la relation de cause à effet entre les indicateurs.

Source de données : chaque indicateur possède une source de données à partir de laquelle il est extrait dans le système du tableau de bord prospectif.

6.3.2 Élaboration des tableaux de bord prospectifs avec ARIS BSC

6.3.2.1 Définir les perspectives

Au début d'un projet de tableau de bord prospectif, les perspectives doivent être définies dans le cadre d'une planification stratégique. Ces perspectives sont utilisées pour tous les tableaux de bord prospectifs d'une entreprise.

Pour cela, le type d'objet **Perspective** est à votre disposition. Les perspectives peuvent être modélisées dans le diagramme cause à effet TBP.

Les perspectives sont reliées à l'aide de la liaison **est influencé par**. La définition de l'ordre de tri n'est pas nécessaire du point de vue de la méthode. En revanche, la définition logique des structures de perspectives (c.-à-d. l'ordre des différentes perspectives) permet de simplifier au plus haut point une modélisation de modèle automatique et de clarifier la structure logique de la chaîne de cause à effet.

6.3.2.2 Définir la structure du système de tableau de bord prospectif

Les modèles suivants constituent le cadre de l'élaboration du système de tableau de bord prospectif :

- Organigramme

  Un système de tableau de bord prospectif peut être élaboré sur la base de la structure organisationnelle d'une entreprise. Vous pouvez associer autant de diagrammes de cause à effet TBP que vous le souhaitez aux objets de l'organigramme (par ex. pour la constitution de variantes, la gestion de l'historique, etc.). Cette association permet d'affecter les objectifs nécessaires à la conversion de la stratégie aux unités structurelles correspondantes. Le système de tableau de bord prospectif d'une entreprise peut donc être adapté, grâce à l'orientation vers l'organigramme, du niveau le plus haut de l'entreprise jusqu'au niveau des différents services ou collaborateurs.
- Modèle de composition

Si une entreprise souhaite définir l'élaboration du système de tableau de bord prospectif non pas avec un organigramme, mais par exemple grâce à des domaines commerciaux spécifiques, le modèle de composition est le modèle source idéal pour la structure du tableau de bord prospectif. Dans ce cas, vous pouvez associer autant de diagrammes cause à effet TBP que vous le souhaitez à un objet de composition.

- Diagramme de chaînes de plus-value ou arbre de fonctions

  Étant donné que le tableau de bord prospectif est un instrument conçu pour la gestion et la mesure du rendement des entreprises, il peut également être élaboré de manière orienté plus-value dans ARIS. Pour cela, le diagramme de chaînes de plus-value ou l'arbre de fonctions peuvent être utilisés.

6.3.2.3 Définir les relations de cause à effet

Des diagrammes cause à effet TBP sont associés aux objets de l'organigramme, aux fonctions ou aux éléments de composition dans le modèle de composition qui définissent l'élaboration du système de tableau de bord prospectif à l'échelle de l'entreprise.

Dans le diagramme cause à effet TBP, les objectifs et facteurs de succès stratégiques devant être déterminés pour la conversion de la stratégie sont définis et leur influence réciproque est représentée dans les différentes perspectives du tableau de bord prospectif. Pour cela, les différentes perspectives qui sont utilisées dans toutes les chaînes de cause à effet de l'entreprise sont d'abord créées comme objets dans ARIS.

Le diagramme cause à effet TBP est un modèle Lane qui permet l'enregistrement des types d'objets **Perspective**dans la colonne **Perspectives pertinentes** et des types d'objets **Stratégie** dans la ligne **Stratégie**. Dans les colonnes de cause à effet, les types d'objets **Objectif**, **Facteur de succès** et **Indicateur** peuvent être modélisés. Dans les colonnes de cause à effet, la puissance d'effet des objectifs stratégiques et leurs facteurs de succès sont décrits. Plus la puissance d'effet augmente, plus les flèches correspondantes sont représentées en gros.

Dans le diagramme cause à effet TBP, des facteurs de succès peuvent également être définis pour un tableau de bord prospectif. L'utilisation du type d'objet **Facteurs de succès** permet de reprendre dans le tableau de bord prospectif des facteurs qui ne sont pas directement influençable par l'entreprise, par ex. la croissance du marché, ainsi que leurs indicateurs.

Vous trouverez le diagramme cause à effet TBP sur le niveau des règles de gestion de la vue de gestion.

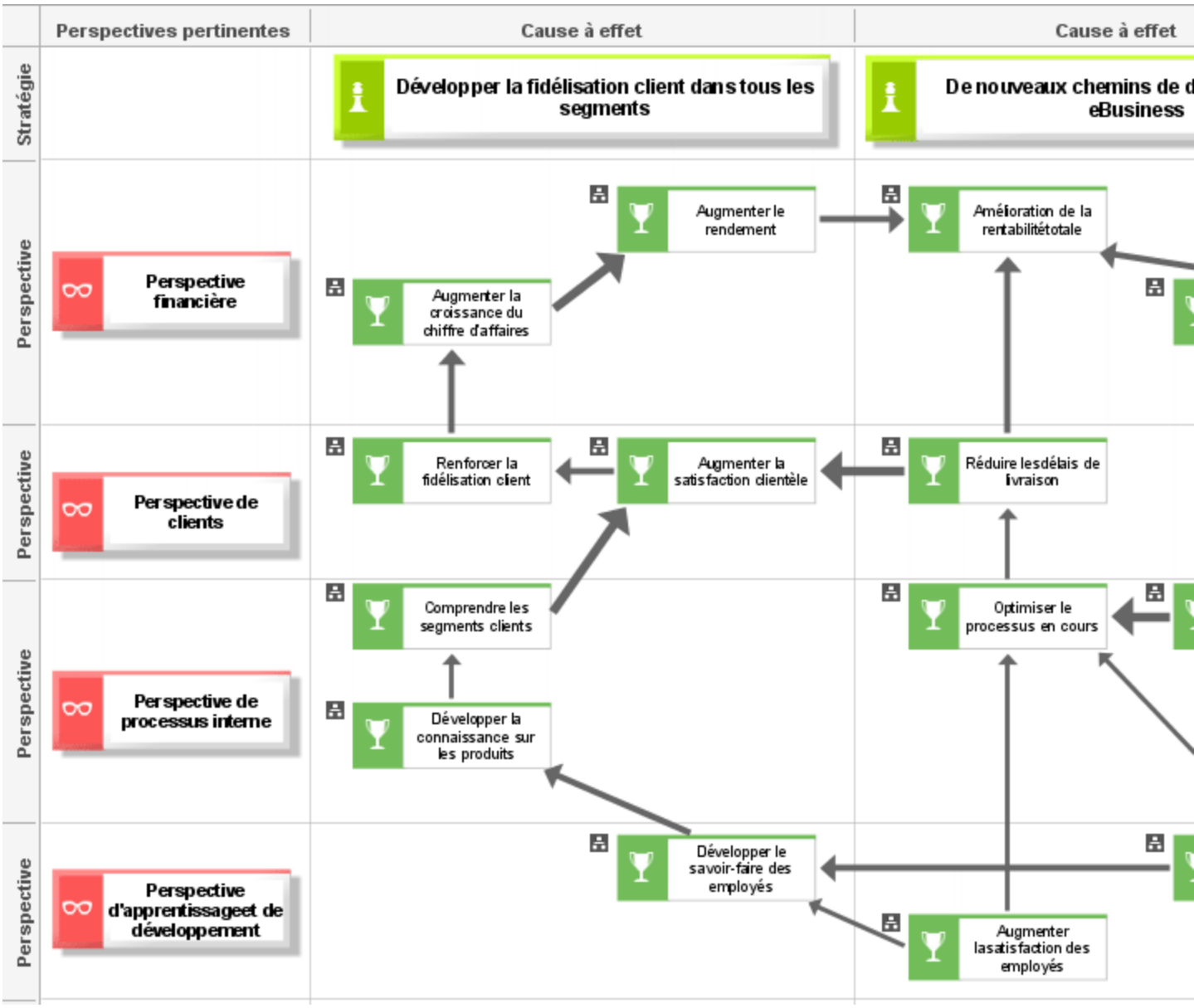


Illustration 123: Diagramme cause à effet TBP

Seul un diagramme d'affectation d'indicateurs TBP peut être associé aux types d'objets **Objectif** et **Facteur de succès** dans une colonne de modélisation du diagramme cause à effet TBP.

Les symboles suivants sont utilisés dans le diagramme cause à effet TBP :

Symbole	Type d'objet
	Perspective

Symbole	Type d'objet
	Stratégie
	Cible
	Facteur de succès
	Instance d'indicateur

6.3.2.4 Définir les initiatives et les indicateurs pour le contrôle de l'objectif

Après la définition des objectifs stratégiques et des facteurs de succès dans le diagramme cause à effet TBP, un diagramme d'affectation d'indicateurs TBP est affecté à chaque objet. Les indicateurs qui doivent servir de mesure pour l'objectif ou le facteur de succès y sont affectés aux objectifs et aux facteurs de succès correspondants. Ces indicateurs sont enfin calculés et canalisés. Si plusieurs indicateurs sont affectés à un objet, la liaison du type **est mesuré par/mesure** permet de définir une pondération de l'indicateur correspondant. Elle indique la pondération avec laquelle l'indicateur entre dans le degré d'atteinte de l'objectif ou de l'objectif du facteur de succès. Pour que l'exécution des scripts d'analyse d'ARIS TBP soit réussie, l'attribut **Pondération** doit être géré pour toutes les liaisons du type **est mesuré par/mesure**.

Outre les indicateurs, des objets pour la définition de l'origine des données peuvent également être affectés aux objectifs ou aux facteurs de succès dans le diagramme d'affectations d'indicateurs TBP, comme le type d'entité, fichier, document, base de données, support de données, etc. Cette affectation permet de créer une connexion à la méthode Data Warehouse d'ARIS. Vous pouvez y définir exactement les éléments de données de la méthode Data Warehouse avec lesquels l'indicateur de tableau de bord prospectif correspondant est connecté.

Grâce à la possibilité de décrire également la structure d'effets d'indicateurs grâce au type de liaison **influence/est influencé par**, les effets des indicateurs précoces sur les indicateurs retardés peuvent être décrits dans les diagrammes d'affectations d'indicateurs TBP. Pour cela, nous vous conseillons de créer un diagramme d'affectation d'indicateurs TBP supplémentaire, dans lequel vous mettrez au premier plan non pas l'affectation de différents indicateurs à des objectifs ou des facteurs de succès, mais la fusion et la définition des indicateurs utilisés provenant des autres diagrammes d'affectation d'indicateurs TBP ainsi que la description de leurs effets.

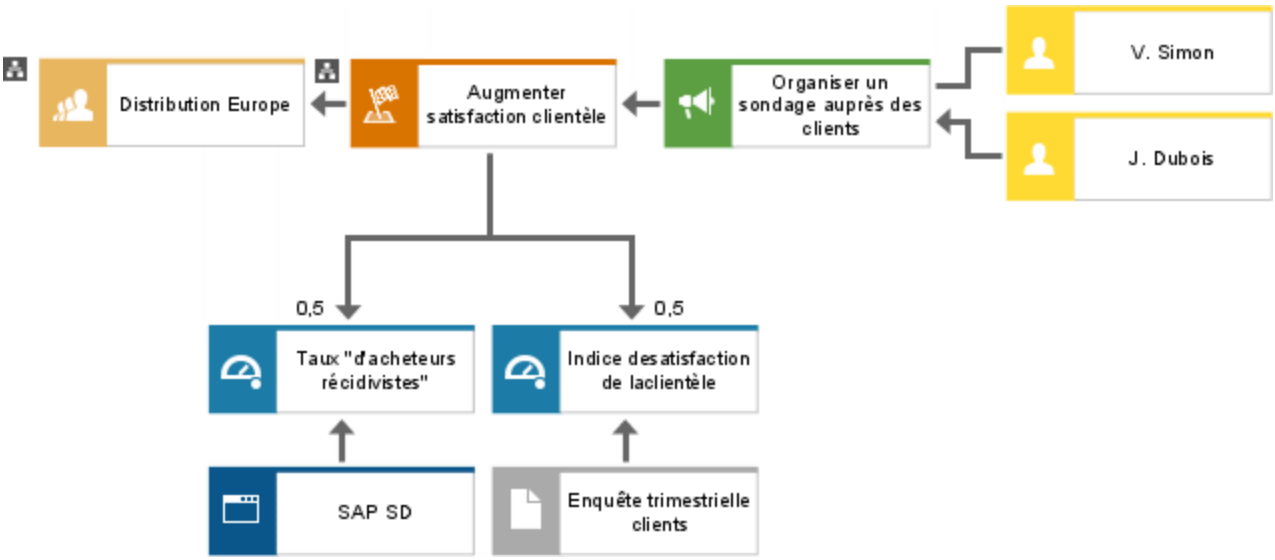
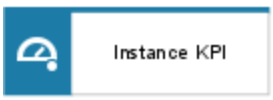
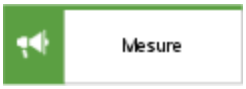
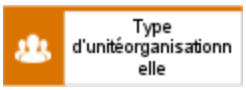
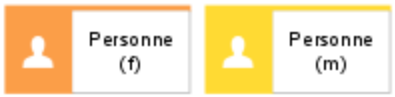
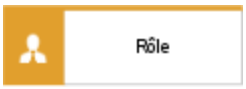
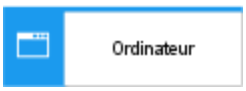
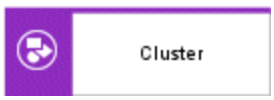


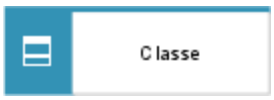
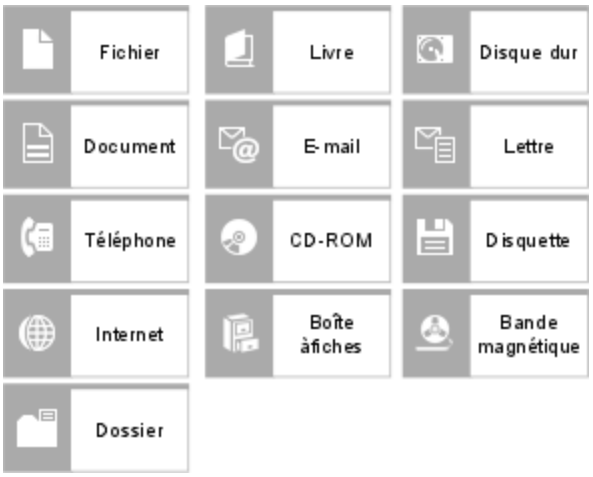
Illustration 124: Diagramme d'affectation d'indicateurs TBP

Vous trouverez le diagramme d'affectation d'indicateurs TBP sur le niveau des règles de gestion de la vue des processus.

Les symboles suivants sont utilisés dans le diagramme d'affectation d'indicateurs TBP :

Symbole	Type d'objet
	Cible
	Facteur de succès
	Instance d'indicateur
	Tâche

Symbole	Type d'objet
	Unité organisationnelle
	Type d'unité organisationnelle
	Poste de travail
	Personne
	Rôle
	Groupe
	Type d'application
	Terme spécifique
	Type d'entité
	Type de relation
	Attribut MER
	Cluster/Modèle de données
	Catégorie des connaissances

Symbole	Type d'objet
	Connaissances documentées
	Classe
	Support d'informations

6.3.2.5 Décrire les indicateurs et leurs rapports

Après la détermination des indicateurs pour la mesure d'objectifs stratégiques et des facteurs de succès critiques, ces indicateurs et leurs rapports peuvent être définis plus en détail grâce au type de modèle **Arbre d'indicateurs**.

Dans un arbre d'indicateurs, différents indicateurs sont structurés hiérarchiquement grâce à la liaison du type **influence**. L'attribut **Pondération** peut être géré pour ces liaisons, le calcul d'un indicateur total étant alors possible au sein d'un arbre d'indicateurs grâce aux pondérations. L'arbre d'indicateurs se voit attribuer l'instance d'indicateur qui représente l'indicateur total. L'association est prise en compte par les évaluations TBP **Créer vue de gestion** et **Exécuter comparaison budget/réalisé**.

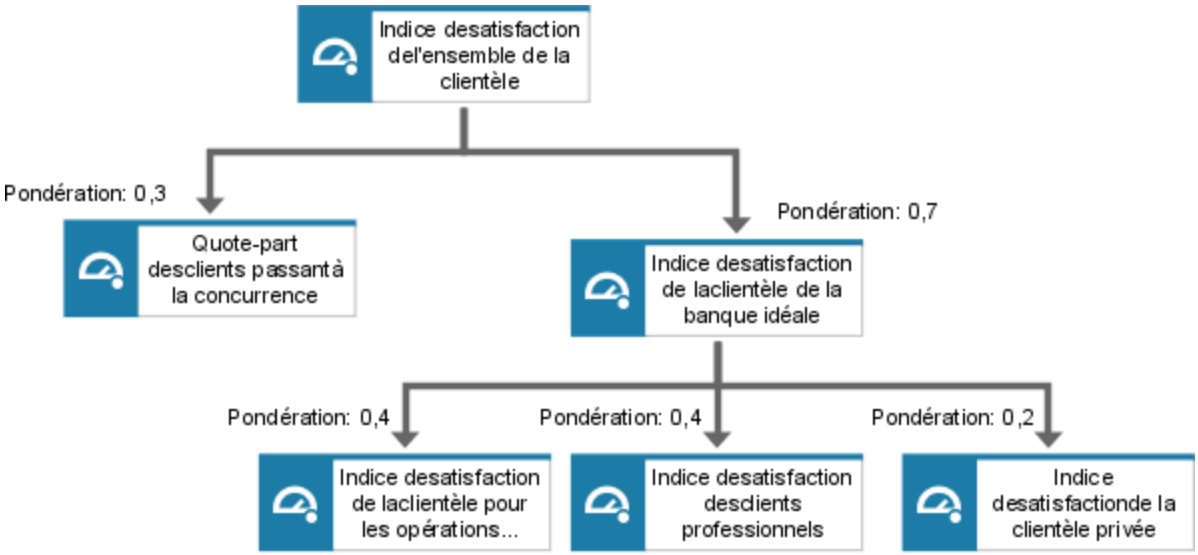


Illustration 125: Arbre d'indicateurs

Le type de modèle **Arbre d'indicateurs** utilise uniquement le type d'objet **Instance d'indicateur**.

Vous trouverez l'arbre d'indicateurs sur le niveau des règles de gestion de la vue des données.

6.3.3 Relations avec d'autres modèles

Les objets de la méthode du tableau de bord prospectif peuvent avoir des relations vers les modèles ARIS suivants :

- CPE
- Diagramme des objectifs
- Structure DW

7 eBusiness scenario diagram

7.1 Introduction

Il devient de plus en plus important que le déroulement des processus d'entreprise au sein de l'entreprise se fasse sans heurts. D'un côté on regarde le déroulement de processus déterminés aux interfaces des entreprises, de l'autre côté entre l'entreprise et ses clients. Les contacts devraient s'exécuter de manière claire, rapide, uniforme et directe.

Il devient également de plus en plus important de trouver rapidement les partenaires commerciaux adéquats du point de vue de l'entreprise et de bons fournisseurs du point de vue des consommateurs. La structuration optimale de ces processus signifie un avantage en matière de compétition. La plate-forme appropriée pour assister ces relations multilatérales est l'Internet. Puisque les processus au sein de l'environnement décrit précédemment sont multiples, il est nécessaire de définir ce que l'on entend par e-business.

E-business est un terme générique de la gestion de l'activité commerciale d'une entreprise au moyen des technologies de l'information et de la communication. Il englobe la gestion des relations et processus entre partenaires commerciaux, collaborateurs et clients via des outils électroniques (Herrmans, Sauter, 1999).

Ainsi, la création d'un site Web pour la présentation d'une entreprise, l'achat d'un article sur Internet, un projet très complexe de deux entreprises ou les relations multiples entre plusieurs partenaires se rencontrant sur une place de marché sont considérés comme faisant partie du e-business.

Termes affiliés :

B2B (BUSINESS TO BUSINESS)

Business to Business désigne les processus de transactions entre entreprises. Ces derniers sont rendus possibles par la connexion des chaînes d'approvisionnement des entreprises.

B2C (BUSINESS TO CONSUMER)

Business to Consumer désigne les processus de transactions entre les entreprises et leurs clients. Les achats des clients dans les magasins en ligne en constituent le meilleur exemple.

B2A/C2A (BUSINESS/CONSUMER TO ADMINISTRATION)

Business/Consumer to Administration décrit toutes les transactions entre les entreprises ou les particuliers et l'administration publique. On estime qu'il est possible de faire des économies de coûts substantielles, en particulier dans le domaine des contacts entre entreprises et administration.

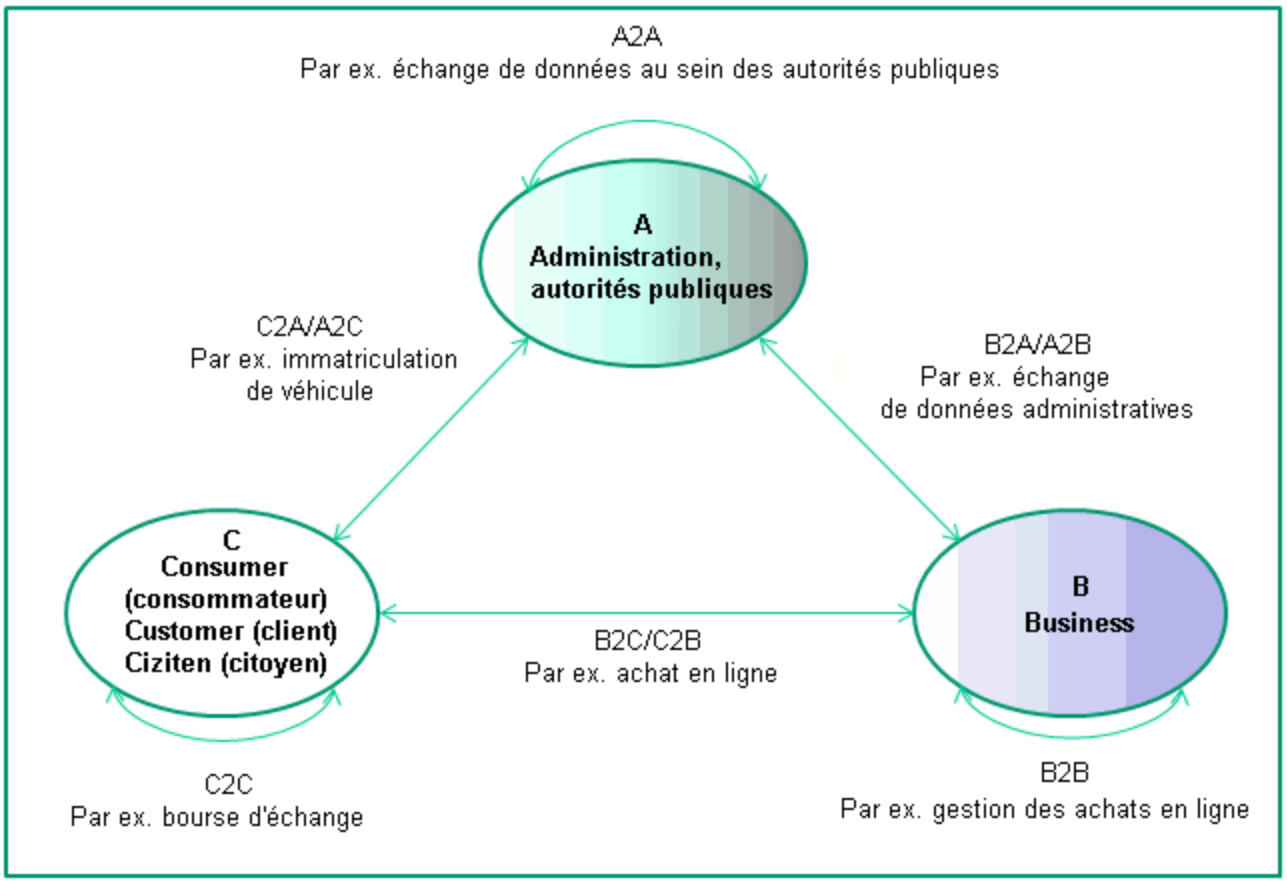


Illustration 126: Possibilités de transaction dans l'E-business

Outre la différenciation entre plusieurs partenaires, vous pouvez également faire une différence entre les différents partenaires en ce qui concerne l'ampleur des relations : one to one, one to many, many to many. Le scénario de la place du marché est celui qui a gagné le plus d'importance.

PLACES DU MARCHÉ

Les places du marché électroniques sont des places virtuelles sur lesquelles un nombre non limité d'acheteurs et de vendeurs négocient ouvertement des marchandises et des prestations de services et échangent des informations.

L'e-Business scenario diagram a été développé pour assister ces scénarios possibles. En collaboration avec d'autres méthodes et différents composants qu'ARIS met à votre disposition, la concrétisation de projets e-business peut être assistée de manière optimale. Ce chapitre, dédié aux diagrammes e-Business scenario, décrit d'abord la méthode avec tous les

objets et possibilités d'évaluation puis la connexion à d'autres méthodes. Enfin, un cas d'application est présenté, illustrant les possibilités complexes.

7.2 La méthode du e-Business scenario diagram

7.2.1 L'idée

Le fait de pouvoir observer globalement une chaîne de plus-value, c'est-à-dire du client final à toutes les entreprises participant au processus, doit permettre de développer un potentiel de développement. L'objectif est par exemple l'amélioration de la chaîne d'approvisionnement, la réduction des coûts d'approvisionnement et de distribution ou le perfectionnement de la structure des systèmes d'information. La représentation grâce au e-Business scenario diagram permet la visualisation des contenus qui doivent être examinés afin d'atteindre les objectifs nommés. La forme sélectionnée de l'affichage des colonnes permet de représenter de manière abstraite les interfaces entre les différents partenaires de processus sur les marges des colonnes. Différents rapports complètent les modèles et offrent d'importantes possibilités d'analyse.

7.2.2 Le modèle et ses objets

Les sujets économiques examinés dans le modèle sont insérés dans l'en-tête et nommés **Business participant**. Ils proviennent de la vue organisationnelle et il est possible de leur associer des organigrammes pouvant clarifier la structure de l'entreprise ou les relations entre les différents objets des colonnes.

Les objets centraux et structurants du modèle sont les processus uniques participant au processus global des sujets économiques et les interfaces qui les relient entre eux. Un processus unique, appelé également **Business process**, est un processus d'entreprise participant de manière décisive à des coopérations en allant au-delà de l'entreprise. La représentation et l'analyse exacte d'un tel processus unique a lieu grâce à l'association d'un modèle de processus. Tous les déroulements concernant une entreprise sont modélisés dans la ligne située sous le Business participant mais dans la même colonne. L'accord entre les entreprises exige également l'analyse précise des **applications** et du matériel utilisés par les différents objets économiques pour prendre en charge leurs processus uniques, par ex. les systèmes ERP. Ils sont représentés par des **Business components**. Pour harmoniser ces différents composants, il est nécessaire de déterminer les responsabilités exactes pour les systèmes. Pour cela, des objets du type **Organizational unit type** peuvent être utilisés. Les rôles des collaborateurs participant au processus peuvent également être définis. Ils sont désignés par **Employee role** dans le modèle. L'incorporation des interfaces représente en général un défi important pour la modélisation e-Business. Les marges des colonnes prennent

une signification essentielle car elles symbolisent les interfaces entre les participants du processus. Leur observation se fait selon différents angles de vue.

La transmission d'informations spécifiques au processus peut représenter un des points forts. C'est à cela que servent les **Business documents**, qui peuvent prendre la forme d'un document XML ou HTML. Au Business document peut être associé un modèle de la vue de données, par exemple une DTD. Vous pouvez également illustrer le flux monétaire ou de marchandises grâce aux objets **Money transaction** ou **Goods shipment**.

Un autre aspect revêt une grande importance : la sécurité des données transmises via Internet et surtout celle des paiements effectués électroniquement doit être garantie. Pour cela, différentes techniques de codage sont utilisées, par ex. SET (Secure Electronic Transaction) ou SSL (Secure Socket Layer). Cet aspect de la sécurité est pris en compte par l'incorporation de l'objet **Security protocol**. Des responsables, représentés également par des objets du type **Organizational unit type** , peuvent aussi être définis pour la sécurisation des transactions. L'analyse d'un aspect plutôt technique, celui de la forme technique de la transmission des données aux interfaces, peut également faire partie du point fort. Pour ce faire, différents supports d'informations sont utilisés dans le modèle. Les processus uniques peuvent être reliés par **Intranet**, **Extranet** ou **Internet**. La transmission de données peut se faire par **e-mail**. Comme support de transmission, le téléphone portable gagne énormément d'importance.

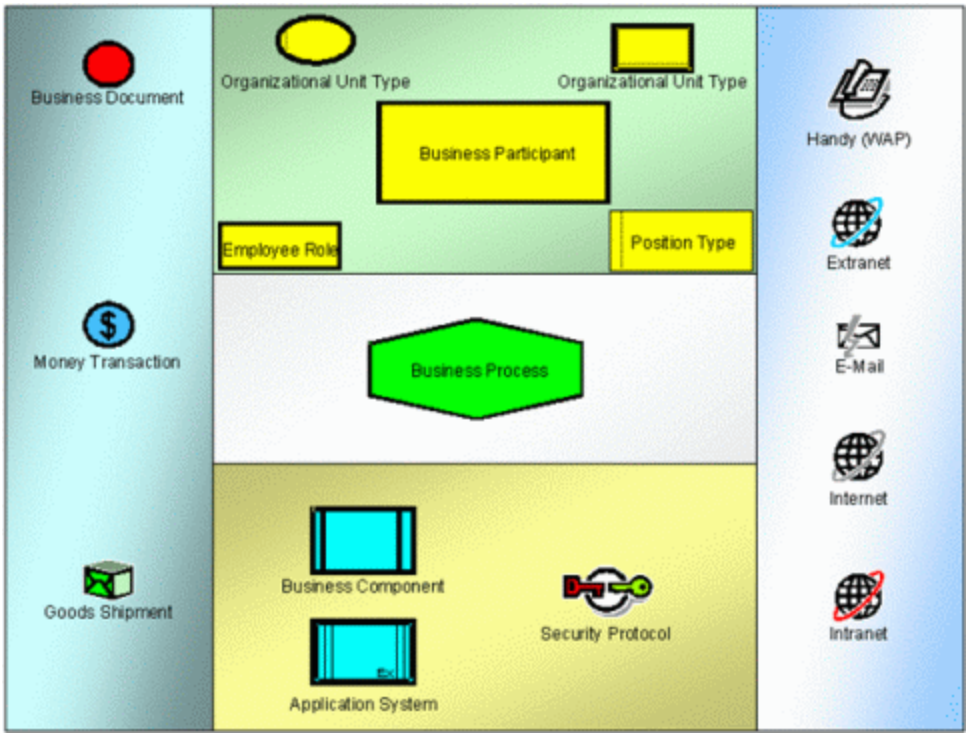


Illustration 127: Les objets dans le e-Business scenario diagram

7.2.3 Groupe d'attributs Mode de transmission

Les objets modélisés peuvent être spécifiés plus en détails par la gestion de leurs attributs. Un attribut est adapté spécialement aux exigences du e-business.

Le groupe d'attributs **Mode de transmission** des objets **Business document**, **Money transaction** et **Goods shipment** doit être mis en valeur. La gestion d'un attribut de mode de transmission permet non seulement de caractériser le chemin de transmission, mais également d'exprimer la nécessité de la sécurisation de la transaction. S'il s'agit par exemple d'une transmission en ligne, la sécurisation susnommée des informations et données secrètes ne doit pas manquer.

7.3 Evaluations par les rapports

Différentes possibilités d'évaluation supportent la modélisation des eBusiness Scenarios. Les évaluations se font sous forme de rapports. Plusieurs évaluations prédéfinies vous sont proposées par ARIS, mais vous pouvez également utiliser vos propres évaluations personnalisées. Les rapports expliqués ci-dessous sont fournis pour les eBusiness Scenarios.

7.3.1 Vérification de la sécurité des données

La sécurité des données transmises en ligne est l'un des points les plus importants influençant l'acceptation du e-business. La protection des données personnelles ou même des paiements contre les accès de personnes non autorisées est un problème qui doit être résolu si vous ne voulez pas perdre la confiance de vos clients et partenaires. Grâce à un rapport, toutes les prestations (Money transaction et Goods shipment) et données (Business documents) échangées peuvent être contrôlées à ce niveau. Pour ce faire, le groupe d'attributs **Mode de transmission** déjà nommé est évalué et si la transmission se fait en ligne, le programme contrôle si le codage de données se fait. Cela permet d'identifier d'éventuels problèmes au niveau de la sécurité et de rechercher les méthodes de codage désuètes, ce qui permet une mise à jour plus facile.

7.3.2 Support système

Un deuxième aspect important lors de la réalisation de projet e-business est la compatibilité des applications. De nombreuses questions se posent aux entreprises. Quels processus doivent être pris en charge par quels systèmes ? Qui est responsable du fonctionnement de

quels systèmes ? Où des problèmes de formation pourraient éventuellement se poser ? Quelles adaptations des systèmes existants doivent être entreprises ? Les réponses peuvent se trouver grâce à un rapport. Les différents processus sont répertoriés avec les systèmes s'y rapportant et leur responsable.

7.3.3 Flux d'informations

Contrairement aux autres modèles de processus, la transaction est au premier plan pour les eBusiness Scenarios. L'accent est particulièrement mis sur les données et les prestations/produits qui sont échangés. C'est la raison pour laquelle des évaluations vous sont proposées. La question de savoir quelles sont les données et les prestations/produits qui ont été créés à quel endroit et l'utilisation qu'on en fait est très importante. Un rapport sort les données et les prestations/produits modélisés, ainsi que les processus correspondants, auxquels les données et les prestations/produits se rapportent comme Input ou Output.

7.3.4 Collaborative business maps

Les Collaborative business maps utilisés par SAP représentent une forme spéciale de modèle. Ils mettent en valeur la vue sur différents partenaires. Une différenciation est faite entre deux vues qui sont orientées sur différents groupes cibles. Une "aggregated view" qui ne contient que des Business participants ainsi que des processus et qui est destinée aux cadres supérieurs, et une "detailed view" contenant les documents et les rôles des responsables processus pour les services. Les rapports autorisent à tout moment de transmettre les informations représentées dans le e-Business scenario diagramm dans les deux vues.

7.4 Connexion à d'autres méthodes et composants

Les différentes méthodes de modélisation vous permettent de consulter des informations sous différents angles de vue et de les rendre accessibles à différents groupes cibles sous un angle particulier. Le e-Business scenario est le point de départ de ces considérations. Ses objets permettent d'effectuer des présentations détaillées spécifiques aux groupes cible. Ceci permet d'atteindre la représentation de l'ensemble d'un projet E-business. En outre, les différents composants d'ARIS permettent d'effectuer des évaluations pour les modèles créés afin que ces derniers puissent être pris en charge de manière optimale dans l'environnement e-business.

EXEMPLE : INTRODUCTION D'UN MAGASIN EN LIGNE

La première étape consiste à définir les objectifs à accomplir par les activités e-business planifiées. Cette étape peut être exécutée à l'aide du composant ARIS Tableau de bord prospectif (cf. chapitre **Méthode du tableau de bord prospectif** (page 128)). Les processus devant être optimisés pour atteindre ces objectifs y sont identifiés. Dans l'exemple, la création d'un nouveau chemin de distribution, via Internet, est identifiée comme étant un objectif. Pour rendre ce chemin le plus praticable possible, une documentation et planification exactes sont nécessaires.

La réflexion ne doit pas se limiter au déroulement du processus en tant que tel, mais s'étendre à l'organisation des responsabilités, aux interfaces entre différents systèmes, à la sécurité des données, etc.

Le point de départ de la considération est le e-Business scenario diagram. Dans notre exemple, les Business participants sont d'une part l'entreprise qui souhaite implémenter le système de magasin et d'autre part le client qui va profiter de cette offre. L'ensemble du processus, de "l'entrée dans le magasin" jusqu'à "en sortir", est décomposé en sous-étapes. La représentation comprend la vue du client et celle de l'entreprise. Le e-Business scenario diagram permet l'accès au projet d'introduction. Cette figure contient une répartition de l'ensemble du processus.

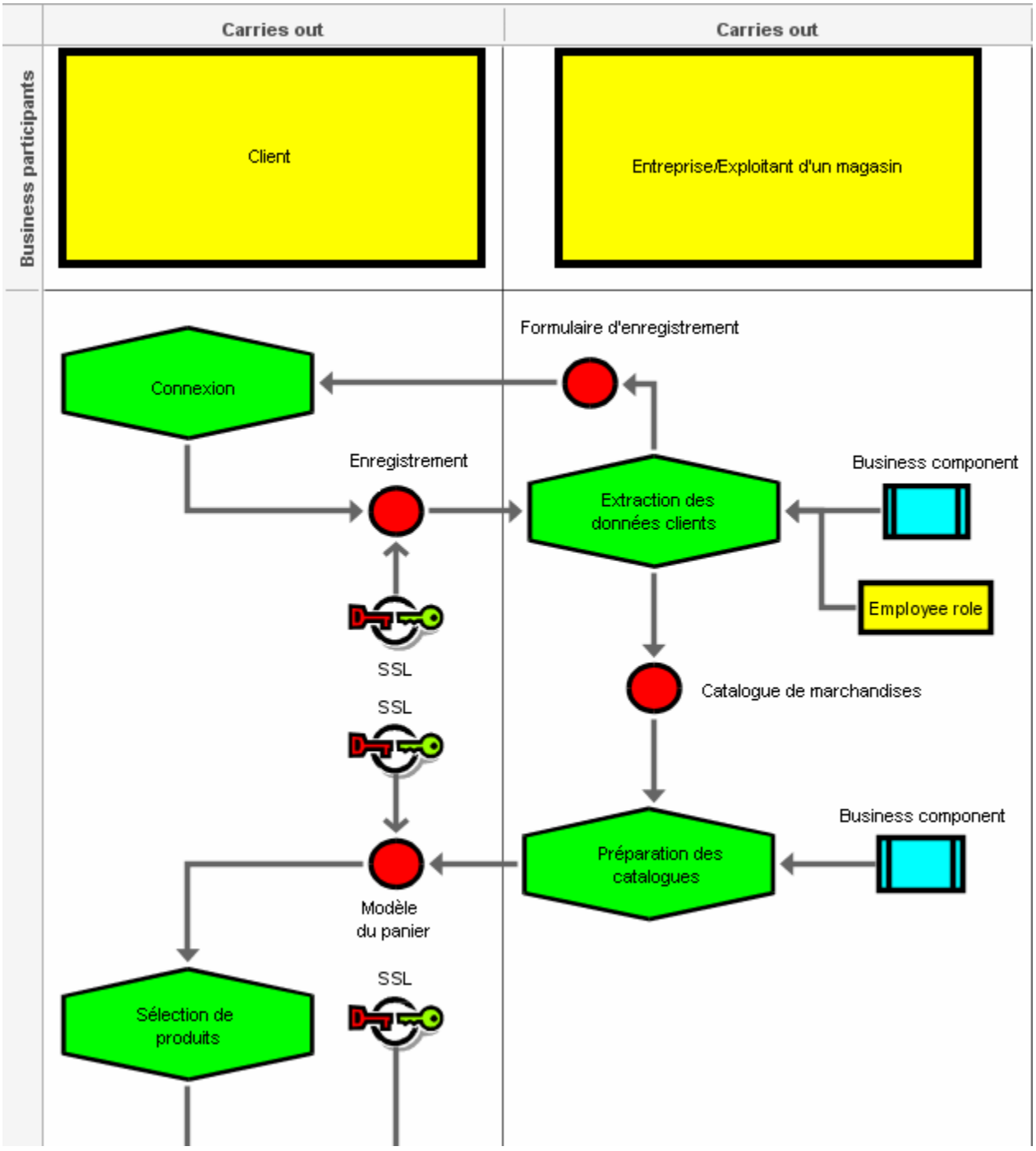


Illustration 128: Extrait du e-Business scenario "Magasin en ligne"

Les différentes étapes de processus peuvent maintenant être détaillées par des CPE, vérifiées par ex. avec le composant Simulation, être représentées par des diagrammes Pipeline après leur optimisation et transformées grâce à Intershop Enfinity en un système de magasin clé en main tout en continuant à être améliorées.

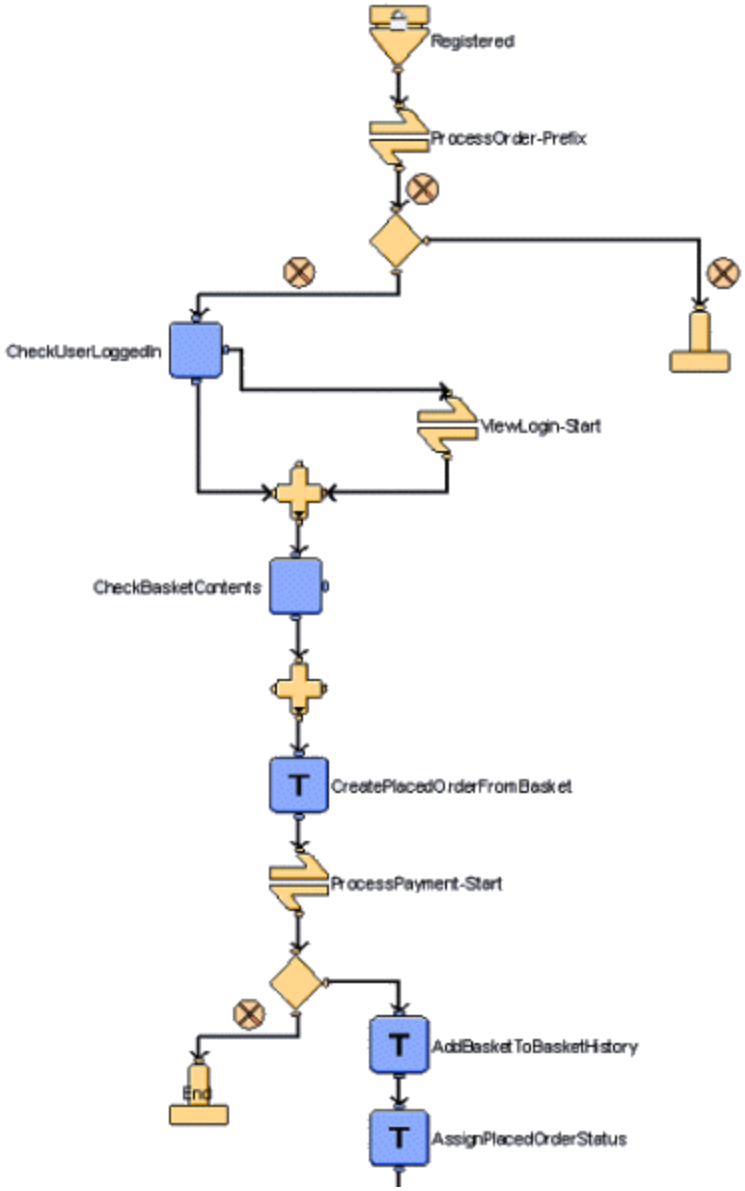


Illustration 129: Extrait d'un Pipeline diagram

Si un magasin doit être relié au système ERP (Enterprise Resource Planning System) sur lequel il se base, les formats corrects des données à transmettre sont nécessaires. Pour que ce soit possible, il existe différentes manières de définir un standard pour les documents et les données. Une des possibilités de la standardisation est l'utilisation du langage XML (Extensible Markup Language). Étant donné que XML est un langage qui continue à être développé de manière complètement hétérogène, une base commune est ici aussi obligatoire. Différentes organisations, composées d'entreprises et d'instituts scientifiques, aspirent à unifier XML pour différents secteurs.

L'utilisation de documents unifiés grâce à XML facilite la liaison de systèmes ERP.

La problématique de l'harmonisation des différents composants a déjà été exposée. Une visualisation des systèmes peut avoir lieu dans le diagramme de type d'application qui peut être associé aux Application systems ou aux Business components en tant que modèle et expliciter les rapports des systèmes.

La structure de l'organisation est elle aussi concernée par l'introduction du E-business. De nouvelles responsabilités doivent éventuellement être définies ou des affectations doivent être ordonnées. Dans le e-Business scenario diagram, des rôles et des unités organisationnelles peuvent être décrits pour les étapes du processus. La position qui est la vôtre au sein de la structure de l'entreprise ou dans le processus peut continuer à être analysée grâce à des organigrammes.

L'étape de l'implémentation commence avec la conversion des contenus modélisés. Lors de l'utilisation de la méthode Intershop, les contenus modélisés sont transformés avec Intershop Enfinity en un système qui fonctionne.

8 IT City Planning

8.1 Architecture d'entreprise et IT City Planning

IT City Planning est une architecture développée par le Français Jacques Sassoon dans les années 1980. Cette architecture doit permettre d'harmoniser une infrastructure de système hétérogène grâce à l'observation précise des interactions existantes, c'est-à-dire de l'échange d'informations entre applications au sein du système.

Sur la base des schémas directeurs d'aménagement et d'urbanisme, la procédure de création d'un IT City Plan est régie par la volonté de permettre une gestion IT stratégique à long terme qui tiendrait compte non seulement du présent, mais également du passé (systèmes hérités) et de l'avenir.

Dans ce contexte, il n'est pas question de redéfinir l'ensemble du système. Il s'agit au contraire de procéder à des modifications de façon progressive au fur et à mesure des projets.

De la même manière que pour l'architecture MDA, prise en charge par les produits ARIS, il s'agit pour commencer de s'appuyer sur des modèles qui décrivent le système d'information indépendamment des informations d'ordre technologique. Toutefois, dans le cadre de IT City Planning, le langage UML n'est pas utilisé, ce qui facilite l'initiation des personnes disposant de compétences techniques plus limitées et renforce l'adhésion à ces nouvelles méthodes.

8.2 A quelles entreprises s'adresse IT City Planning ?

IT City Planning s'adresse aux entreprises suivantes :

- Les entreprises qui disposent d'un vaste portefeuille d'applications.
- Les entreprises qui ont recours aux technologies de l'information depuis longtemps.
- Les entreprises pour lesquelles les technologies de l'information revêtent une importance stratégique.
- Les entreprises qui se regroupent.

Objectifs de IT City Planning :

- Réutilisation des ressources logicielles pour éviter la création de redondances.
- Réduction des frais de maintenance en permettant, grâce à la restructuration "par blocs" et à la définition de nouvelles ressources logicielles, le remplacement des différentes ressources utilisées jusqu'ici et la couverture des différents cas d'application.
- Consolidation des systèmes d'information.
- Préparation de l'utilisation des solutions EAI à un niveau plus élevé.

La création d'un IT City Plan est une mission assurée par le Centre de compétences en intégration (Integration Competence Center). Le plan en lui-même englobe à la fois le Design Pattern et l'architecture technologique et d'information.

EAI = Enterprise Application Integration. Intégration d'applications à l'échelle de l'entreprise. La technologie EAI fournit l'architecture des solutions E-business. Les middleware EAI sont indispensables pour la mise en oeuvre d'une stratégie e-business.

8.3 IT City Planning avec ARIS

ARIS prend en compte les vues suivantes d'un système d'information :

- Vue des données
- Vue des fonctions
- Vue organisationnelle
- Vue de prestations/produits
- Vue des processus

Chacune de ces vues est subdivisée dans les niveaux descriptifs **Règles de gestion**, **Concept informatique** et **Implémentation**. Elles sont centrées sur le cycle de vie d'un système d'information et le degré de proximité avec les techniques de traitement de l'information.

Les niveaux de concept informatique et d'implémentation décrivent pour l'essentiel le système logiciel. Les champs conceptuels de ces niveaux sont étroitement liés et la « traduction » ne pose aucun problème.

Il en va autrement pour la transition entre les règles de gestion et le concept informatique. Lors de l'élaboration du concept informatique, il convient de mettre l'aspect économique en perspective par rapport aux logiciels standard. Cette étape requiert des connaissances en gestion d'entreprise ainsi qu'en informatique (cf. Scheer, A.-W., ARIS - Vom Geschäftsprozess zum Anwendungssystem, 1998, 3ème édition, p. 7).

La vue du système d'information (vue SI) de IT City Planning sert ici d'intermédiaire entre les niveaux. Les types d'objets de la vue SI doivent être installés dans ARIS entre les fonctions et les applications et élargissent ainsi la vue des fonctions dans ARIS. Comme les fonctions, les éléments du système d'information (SI) sont mis en relation avec les différentes structures des vues de la maison ARIS. Ces extensions concernent essentiellement la vue des processus et la vue des données. Par « vue SI » ci-après, nous entendons les types de modèles des vues des fonctions et des processus de la maison ARIS, dans lesquels sont décrites les relations entre les éléments du SI ou qui servent à la description détaillée des éléments du SI dans le contexte des autres vues ARIS.

Les types d'applications, les types de fonctions informatiques et les prises sont considérés ci-après comme des éléments de ce que l'on appelle la vue IT. De la même manière que pour la vue SI, la vue technique englobe tous les types de modèles dans lesquels des relations entre

les types d'applications, les types de fonctions informatiques et le nouveau type d'objet **Prise** sont décrites ou qui servent à une description détaillée de l'un de ces éléments.

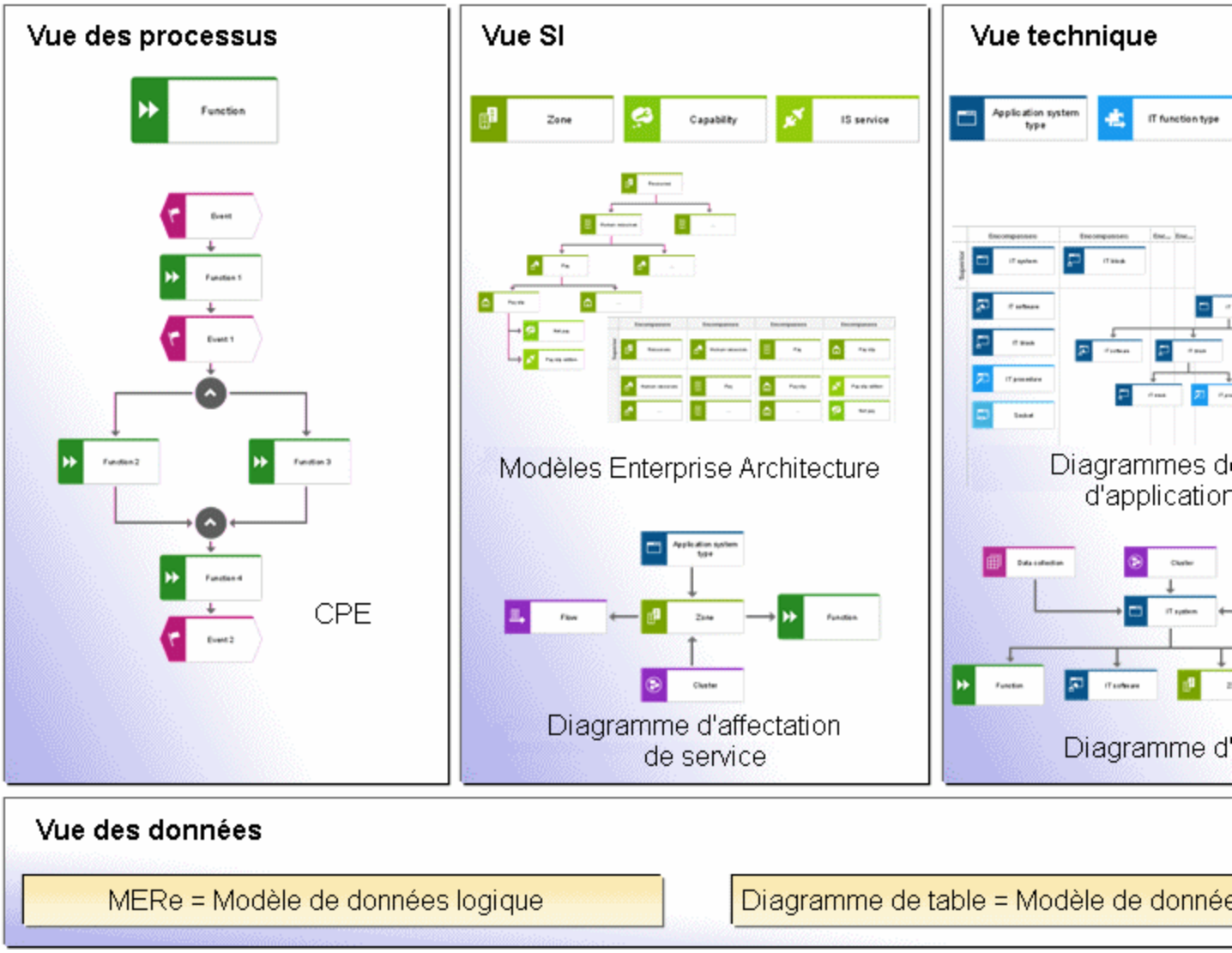


Illustration 130: Vue des processus, vue SI et vue technique

8.4 Vue des services

Pour la description de la vue des services, ARIS propose les types de modèles suivants :

- Diagramme d'architecture de service
- Diagramme d'affectation de service
- Diagramme de collaboration de service

Dans les deux modèles d'architecture de service, le système d'information est structuré sous forme hiérarchique

Une hiérarchie SI peut comporter les niveaux suivants :

- Zone
- Quartier
- Ilot
- Bloc fonctionnel
- Capacité
- Service SI
- Service d'entreprise

Les symboles zone, quartier, îlot, bloc fonctionnel et service d'entreprise appartiennent au type d'objet **Type de service**. Les types de services servent à diviser de façon fonctionnelle un système d'information en unités/blocs indépendants.

Chaque type de service est caractérisé par le fait qu'il est « propriétaire » des données qu'il utilise et des méthodes de traitement correspondantes. D'autres types de services ne peuvent accéder à ces données et méthodes de traitement qu'en appelant un service du « type de service propriétaire ».

Dans un type de service, des données similaires sont utilisées et des activités et fonctions d'entreprise identiques sont exécutées.

Au niveau le plus élevé, le système d'information est divisé en zones. Chaque zone correspond à un domaine d'activité, à un champ de développement,p. .ex.

La figure suivante illustre les différentes zones traditionnellement utilisées pour diviser le système d'information d'une entreprise.

Illustration 131: Zones d'un système d'information d'entreprise

Chaque zone peut à son tour être divisée en un ou plusieurs quartiers.

Les différents quartiers d'une zone sont caractérisés par des processus semblables ainsi que par une grande similitude par rapport aux critères de temps (des cycles de vie ou de traitement de l'information similaires par ex.). Les conditions de paiements, les conditions de prix, l'administration du personnel, les règles de déplacement etc. constituent des exemples de quartiers.

La zone **Ressources** peut par exemple comprendre les quartiers **Ressources humaines** et**Comptabilité** :

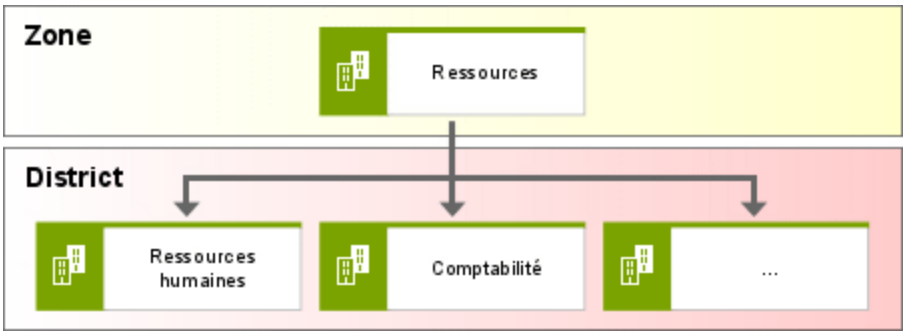


Illustration 132: Zone divisée en quartiers

Au sein d'un quartier, il existe un ou plusieurs îlots répondant à un objectif fonctionnel, p. ex. le paiement des salaires, la facturation, etc.

Le quartier Ressources humaines englobe les îlots Administration, Recrutement, Développement du personnel et Encadrement du personnel.

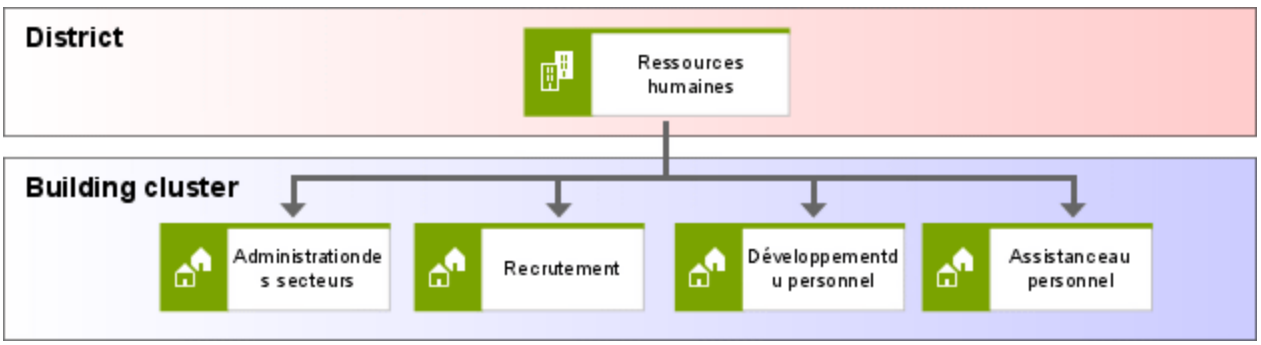


Illustration 133: Quartier divisé en îlots

Chaque îlot peut à son tour contenir un ou plusieurs blocs fonctionnels. Les blocs fonctionnels d'un îlot sont caractérisés par une grande similitude des types d'objets métier et des types d'événements qu'ils gèrent.

Un bloc fonctionnel représente un composant fonctionnel, indépendant et réutilisable. Dans un bloc fonctionnel, les capacités et les services SI sont regroupés en fonction des règles suivantes :

- Ils sont caractérisés par une forte cohérence entre les objets qu'ils gèrent et les fonctions qu'ils supportent.
- Leurs échanges avec d'autres blocs fonctionnels sont minimes.

L'îlot **Encadrement du personnel** dans l'exemple ci-dessous comprend les blocs fonctionnels **Gestion des données de base**, **Certificats**, **Contrôle de gestion** et **Salaires**.

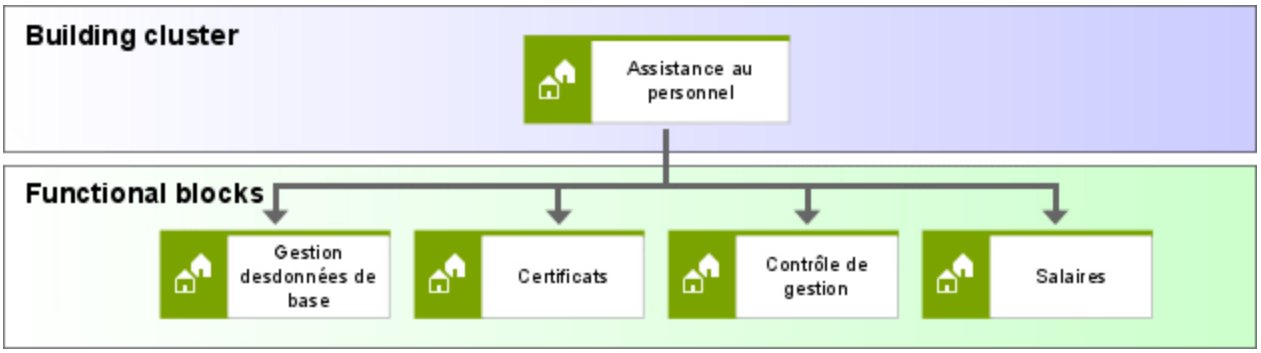


Illustration 134: Division de l'îlot « Encadrement du personnel » en blocs fonctionnels

Une capacité décrit une fonctionnalité élémentaire au sein d'un système. Elle soutient la réalisation d'une activité au sein d'un processus.

Un service SI décrit une interface d'un type de service ou d'une capacité. Les fonctions SI et les services SI assurent aux autres éléments du SI l'accès contrôlé aux données et méthodes de traitement de l'élément du SI qui met à disposition le service.

Ces interfaces permettent d'échanger des messages avec les autres éléments de la vue du SI.

La figure ci-dessous illustre les fonctions SI et les services SI du bloc fonctionnel **Salaires**.

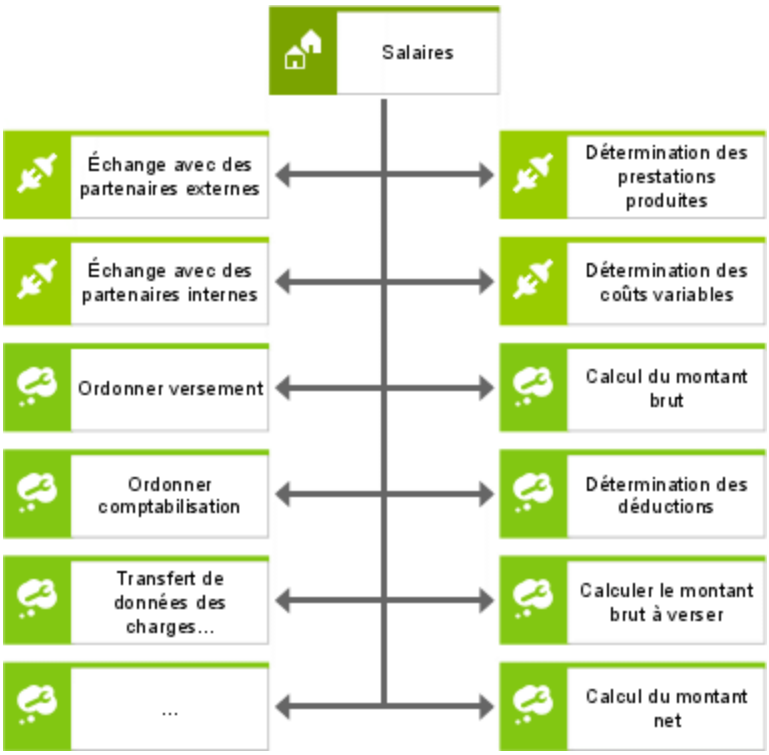


Illustration 135: Capacités et services SI du bloc fonctionnel « Salaires »

Afin de décrire la hiérarchie SI, il n'est pas nécessaire de modéliser entièrement tous les niveaux décrits ci-dessus. Les éléments du SI **Capacité** et **Service SI** ne sont pas considérés, dans le cadre d'IT City Planning, comme des éléments du plan d'urbanisme. Le champ

d'activité du responsable du plan d'urbanisme s'arrête au niveau des blocs fonctionnels. Les capacités et les services SI relèvent de la compétence de l'architecte (cf. Longépé, Christophe : Le projet d'urbanisation du système d'information, p. 18).

8.5 Les types de services et leurs données

Afin de décrire les types de données enregistrées au sein d'un type de service ou d'une capacité, on utilise le modèle MERe. Dans le cadre de City Planning, les symboles de la vue du SI sont proposés dans le MERe. A l'aide d'une liaison du type **est propriétaire de**, ces objets peuvent être mis en relation par le biais de types d'entités et de types de relations.

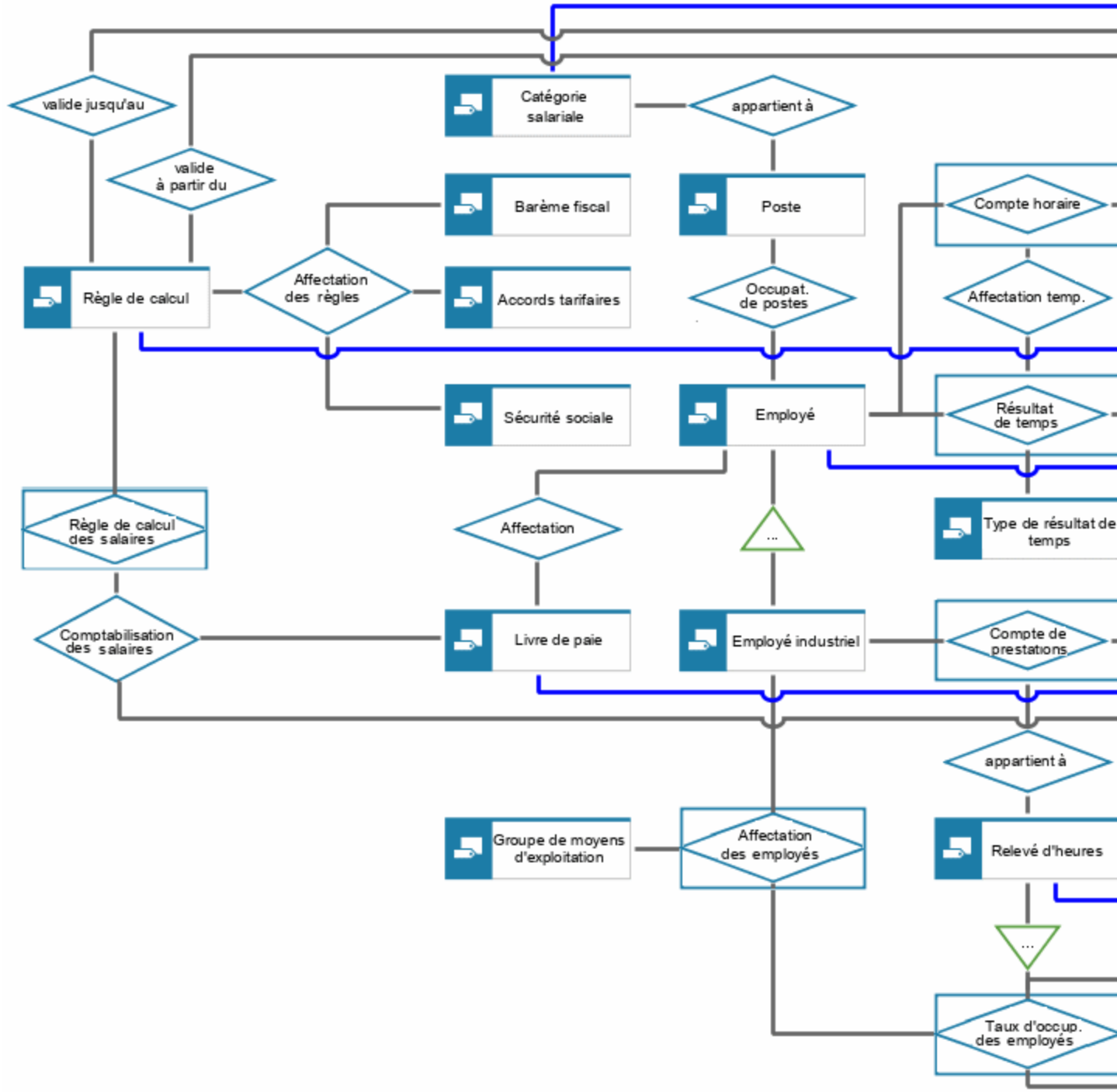


Illustration 136: Liaison « est propriétaire de » entre les symboles de la vue SI et les types de relations et d'entités

8.6 Description détaillée des types de services

Une description détaillée des types de services et capacités d'un système d'information est proposée dans le diagramme d'affectation de services. Voici les informations indiquées :

- les interfaces d'un bloc,
- les interactions entre les différents blocs,
- les applications qui supportent le bloc et
- les fonctions relatives à la gestion d'entreprise qui sont prises en charge par le bloc.

Les zones, quartiers, îlots, blocs fonctionnels et fonctions SI peuvent être reliés à un service SI grâce à la liaison **fournit**.

Les flux d'information entre les types de données peuvent être décrits à l'aide de liaisons d'entrée et de sortie entre les éléments SI et les éléments de données.

Les différents objets d'une application et d'un type de fonction informatique peuvent être affectés aux objets de la vue du système d'information par une liaison du type **assiste**. Si l'on interprète maintenant le City Plan de la même manière que le plan d'urbanisme d'une ville, cette liaison montre les zones du système d'information qui sont « peuplées » par des applications. La liaison **assiste** est également disponible entre les éléments du SI et la fonction.

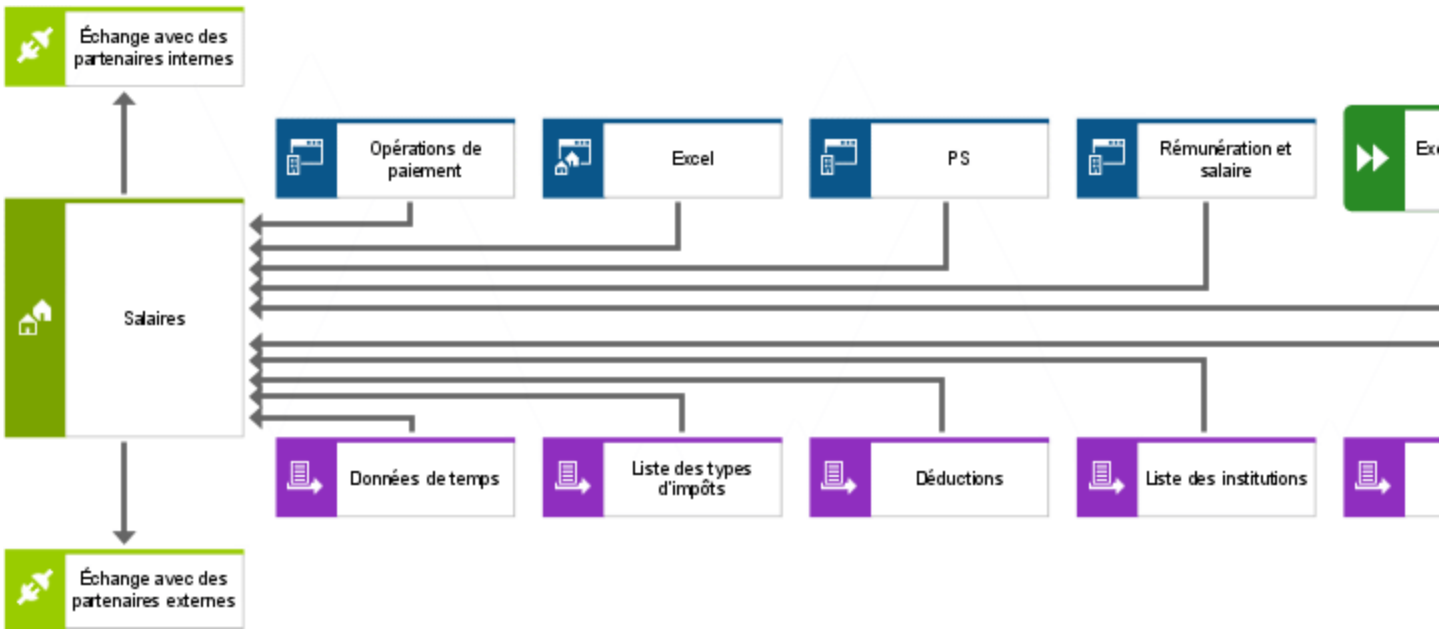


Illustration 137: Liaison « assiste » entre les éléments du SI et la fonction

8.7 Déroulements chronologiques entre les éléments du SI

Dans le diagramme d'affectation de service, il est possible d'établir pour les types de services, les capacités et les services SI spécifiés dans le diagramme d'architecture de service, leurs

relations avec les types d'objets de la vue organisationnelle, de la vue des données et de la vue des processus. Il n'est en revanche pas possible de représenter les éventuels déroulements chronologiques des éléments du SI.

Le diagramme de collaboration de service permet, dans le cadre d'IT City Planning, de représenter les déroulements chronologiques des éléments du SI, et par là même de décrire les aspects dynamiques au sein du système d'information. Il constitue le pendant du diagramme de déroulement de programme (cf. chapitre **Diagramme de déroulement de programme** (page 99)) de la vue technique. Ce type de modèle propose des événements permettant de représenter le déroulement. Comme pour l'affectation des éléments du SI et des événements dans le diagramme de déroulement de programme, il est possible de définir le déroulement des modules fonctionnels dans le diagramme de collaboration de service. L'événement ici doit être compris dans le sens d'un trigger, qui lance des types de services, des capacités ou des services SI. Les ramifications peuvent être représentées avec les règles que l'on trouve dans la CPE ou dans le diagramme de déroulement de programme. Il est également possible de définir les déroulements sans faire appel à des événements.

8.8 Vue technique

De la même manière que pour la vue SI, les types de modèles suivants sont disponibles dans la vue technique :

- Diagramme de type d'application
- Diagramme d'accès
- Diagramme de déroulement de programme

Hiérarchie des applications

Dans le cadre de City Planning, la hiérarchie actuelle des applications au sein de l'entreprise est représentée à l'aide du diagramme de type d'application.

Les niveaux suivants d'une hiérarchie de type d'application peuvent être représentés :

- Système informatique (IT)
- Sous-système
- Applicatif
- Bloc applicatif
- Procédure fonctionnelle
- Prise

Les symboles système informatique, sous-système, applicatif et bloc applicatif représentent des symboles du type d'objet **Type d'application**. La hiérarchie est établie via le type de relations **englobe**.

Au niveau le plus élevé de la hiérarchie du type d'application se trouvent les systèmes informatiques. Un système informatique décrit un ensemble structuré de différents éléments du système informatique, en général des sous-systèmes. L'administration et l'exploitation d'un système informatique sont assurées sous le pilotage d'une même entité organisationnelle.

Un sous-système décrit un composant d'un système informatique. Les composants d'un sous-système sont appelés « applicatifs ».

Un applicatif prend en charge un lot de fonctions homogène. Il est orienté utilisateur et supporte un ou plusieurs processus d'entreprise. Les blocs applicatifs font partie d'un applicatif.

Un bloc applicatif regroupe en général des procédures fonctionnelles qui accèdent aux mêmes données (bases de données, tableaux, fichiers etc.).

Les procédures fonctionnelles sont des objets du type **Type de fonction informatique**. Toute procédure fonctionnelle prend en charge une fonctionnalité précise.

Une prise correspond au Service SI, c.-à-d. qu'elle décrit une interface qui met un élément informatique à la disposition d'autres éléments informatiques afin de leur permettre d'accéder à ses données et méthodes de traitement.

La figure ci-dessous illustre la structure du sous-système du système DATEV :

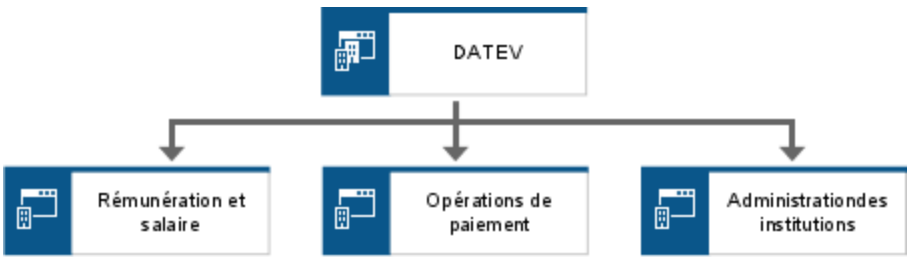


Illustration 138: Sous-système du système DATEV

8.9 Les éléments informatiques et leurs données

Tout comme pour les éléments SI, l'utilisation du type de liaison **est propriétaire de** du MERe entre les types d'applications, les types de fonctions informatiques ou les prises ainsi que le type de relation ou d'entité permet de décrire le type de données enregistrées au sein d'un élément du système informatique.

8.10 Description détaillée des éléments informatiques

Une description détaillée des éléments informatiques au sein de l'IT City Plan est fournie dans le diagramme d'accès. Il correspond au diagramme d'affectation de service de la vue du système d'information.

Il est possible de décrire les points suivants :

- Les relations d'entrée et de sortie de l'élément informatique concerné
- Les fonctions d'entreprise supportées
- Les éléments SI supportés
- L'appel d'autres éléments informatiques via l'élément concerné
- La plate-forme sur laquelle fonctionne l'élément informatique
- L'utilisateur de l'élément informatique

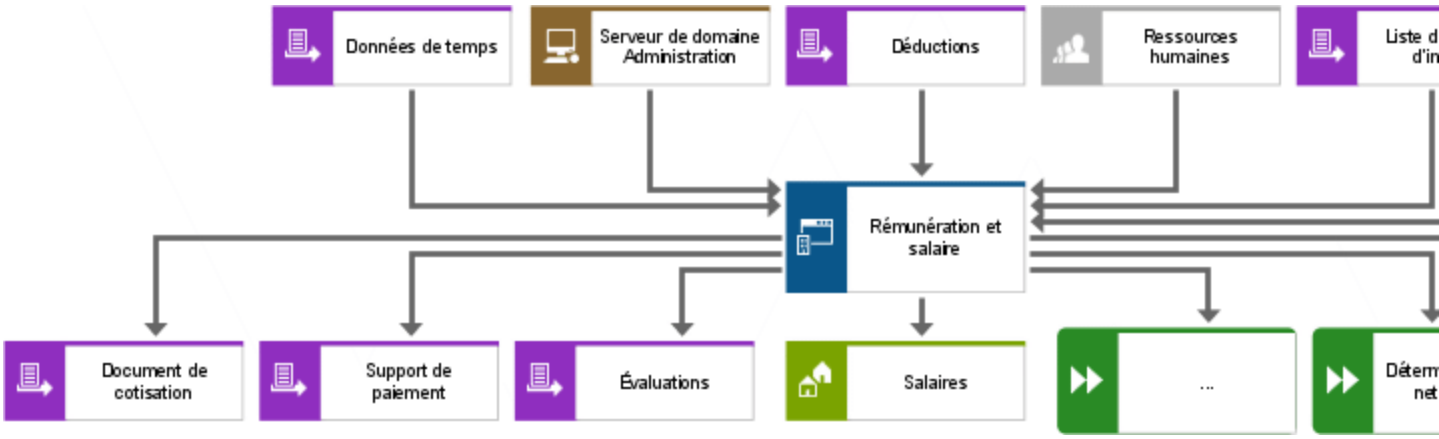


Illustration 139: Description détaillée des éléments informatiques dans le diagramme d'accès

8.11 Aspects organisationnels

Des informations de la vue organisationnelle apparaissent également dans la description détaillée d'un élément informatique. Il ne s'agit pas seulement des informations sur le type d'élément organisationnel pouvant être utilisateur d'un élément informatique. Grâce au diagramme de réseau, il est également possible d'indiquer les influences et les répercussions des différents aspects organisationnels au niveau de l'infrastructure technique.

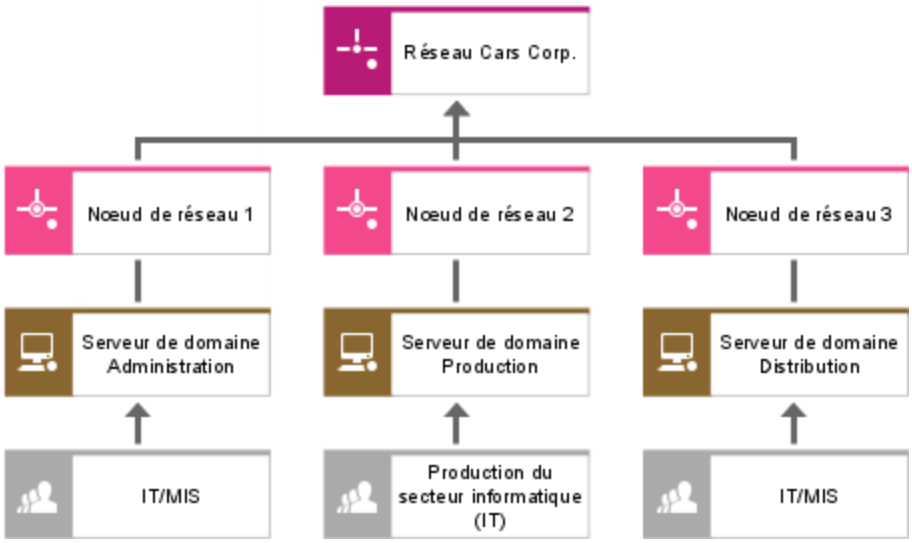


Illustration 140: Influences et répercussions sur l'infrastructure technique

8.12 Déroulements chronologiques entre les éléments informatiques

De la même manière que le diagramme de collaboration de service, le plan de déroulement de programme (PD) sert à décrire les déroulements chronologiques entre les différents éléments informatiques **Type d'application**, **Type de fonction informatique** et **Prise**.

8.13 Déroulements chronologiques au sein de l'architecture

Des objets sont à votre disposition dans les différents modèles de processus (toutes les variations de la CPE) et dans le diagramme de déroulement de programme afin de représenter l'intégration des éléments SI et IT dans un déroulement chronologique.

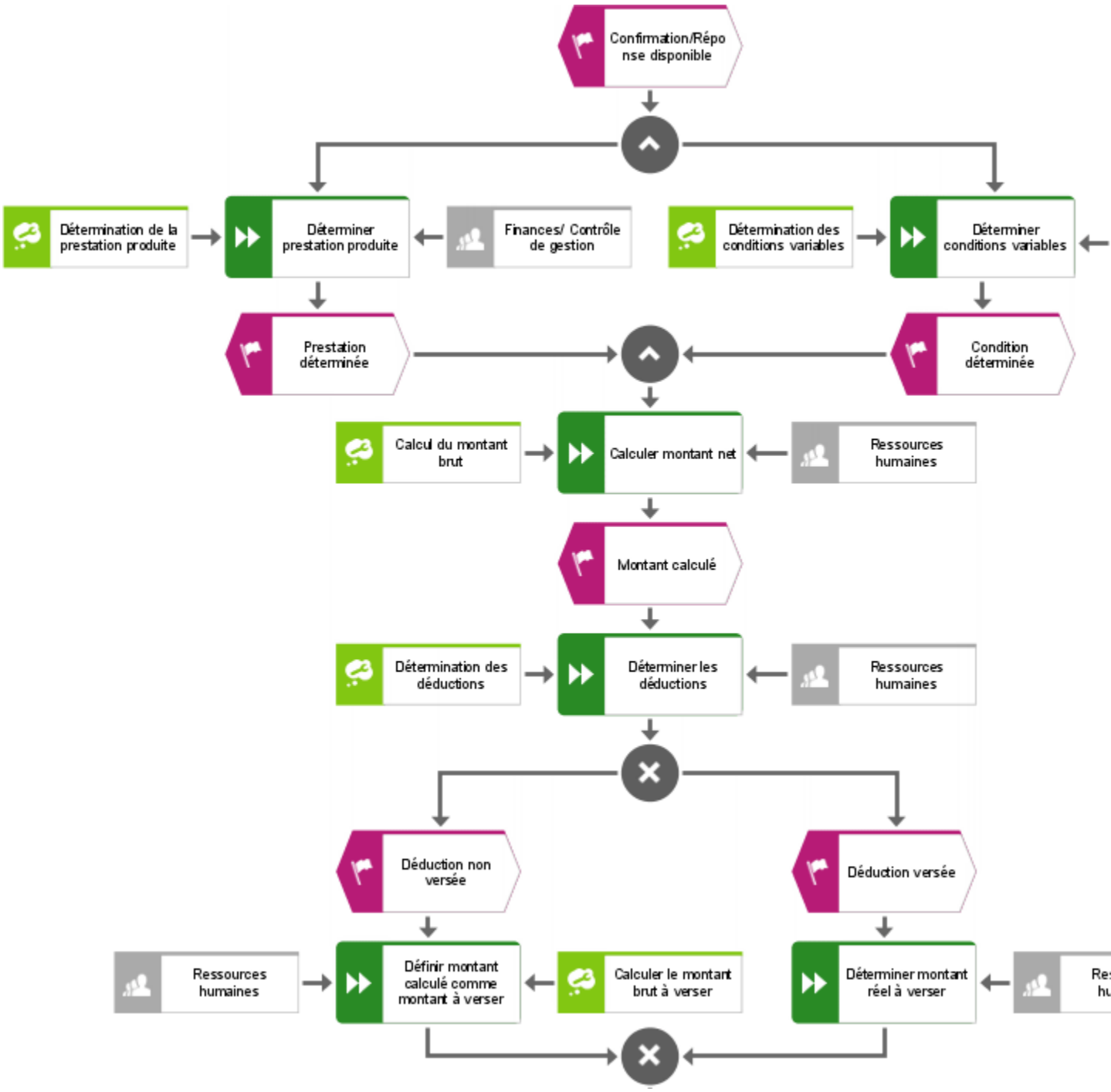


Illustration 141: Intégration des éléments SI et informatiques dans un déroulement chronologique

8.14 Evaluations possibles

Sur la base de la modélisation décrite plus haut, il est possible de procéder à des évaluations qui répondent aux questions suivantes et pourront être d'une aide précieuse lors de la conception de votre système d'information :

- Quelles données sont gérées par un élément donné du SI ?
- Quelles applications prennent en charge un élément du SI ?
- Quelles fonctions sont supportées par un élément donné du SI ?
- Quelles données sont utilisées par les éléments informatiques d'un élément SI donné ?
- Quelles données sont créées par les éléments du système informatique d'un élément SI donné ?
- Quels services SI sont mis à disposition par un élément informatique et dans quels processus sont-ils utilisés ?
- Sur quels composants matériels tournent les applications d'un élément SI donné ?

Pour des types d'applications, types de fonctions informatiques et prises sélectionnés, les évaluations suivantes sont proposées :

- Données utilisées par un élément du système informatique.
- Eléments SI supportés par un élément informatique.
- Fonctions prises en charge par un élément informatique.
- Données utilisées par les éléments SI qui sont pris en charge par un élément informatique.
- Données créées par les éléments SI qui sont pris en charge par un élément du système informatique.
- Composants matériels sur lesquels s'exécute un système informatique.

9 Business process modeling

Les nouvelles technologies d'information et de communication mènent à des interactions et à des transactions de plus en plus complexes entre les entreprises, leurs partenaires, leurs fournisseurs et leurs clients. Le développement et les performances des processus d'entreprise dépendent dans une large mesure de la qualité de la collaboration entre les différents partenaires commerciaux.

Les entreprises doivent mieux comprendre leurs propres activités et les activités de leurs partenaires commerciaux ; d'autre part, elles doivent s'adapter plus rapidement aux changements internes et aux fluctuations du marché. Un langage de modélisation des processus normalisé peut les aider à décrire de façon transparente et flexible leurs processus d'entreprise internes et externes. Les entreprises doivent en outre pouvoir communiquer les processus modélisés à leurs partenaires commerciaux d'une façon personnalisée, claire et compréhensible. Tous les partenaires doivent utiliser le même langage pour décrire les processus.

A ces fins, l'initiative Business Process Management Initiative (BPMI.org) propose un langage de modélisation normalisé et spécialement conçu à cet effet : Business Process Model and Notation (BPMN). BPMN est une notation graphique visant à décrire des processus d'entreprise.

Ce langage doit être compris par tous les utilisateurs. Il s'adresse aux analystes de processus d'entreprise et aux personnes qui surveillent et gèrent les processus d'entreprise, mais aussi aux programmeurs qui implémentent des technologies pour l'exécution des processus.

En outre, le BPMN doit permettre de lire les langages XML utilisés pour l'automatisation des processus d'entreprise, comme, par exemple, le Business Process Execution Language for Webservices (BPEL4WS).

9.1 Les classes de processus et le business process diagram

Pour la description des processus, le BPMN prévoit le type de modèle **Business process diagram (BPD)**. Ce modèle définit trois classes de processus d'entreprise et leurs relations :

- **Private business processes** (processus d'entreprise internes)
- **Abstract business processes** (processus d'entreprise publics)
- **Collaboration processes** (processus d'entreprise globaux)

Les private business processes sont des processus d'entreprise qui ne se déroulent qu'au sein d'une organisation. Ils sont aussi connus sous le nom de workflow ou de processus BPM.

Les différents processus d'entreprise internes sont modélisés sous la forme d'un flux de séquence (sequence flow) dans les limites de différentes pistes (voir le chapitre **Réalisation**

de BPMN dans ARIS (page 175)) dont l'interaction est représentée à l'aide d'un flux de message (message flow).

Le langage BPMN utilise les termes de flux de séquence et flux de message (sequence flow et message flow) plutôt que le terme de flux de contrôle car les processus ne sont pas uniquement contrôlés par les événements, mais aussi par les messages échangés.

Les processus d'entreprise abstraits décrivent les interactions entre les processus privés ou les objets de différentes partitions, ou entre des combinaisons des deux. À l'instar du flux de séquence au sein du processus privé, le flux de message entre les différents processus est particulièrement important. Les interactions sont modélisées à l'aide de flux de message.

Les processus d'entreprise abstraits sont intégrés dans différentes pistes et peuvent être modélisés séparément ou au sein d'un diagramme BPMN. Lorsqu'un processus d'entreprise abstrait apparaît dans le même modèle que le processus d'entreprise privé correspondant, ils peuvent être associés l'un à l'autre.

Les processus de collaboration décrivent exclusivement les interactions entre deux ou plusieurs business entities (partenaires commerciaux). Pour ce faire, une suite d'activités décrivant le modèle d'échange de messages entre les différents partenaires commerciaux est modélisée. Le flux de séquence ne joue plus aucun rôle ici.

Parmi les langages utilisés dans les collaborations, citons, par exemple, bXML BPSS, RosettaNet et W3C Choreography Working Group. La spécification de mappage est prévue dans des versions futures de la spécification BPMN.

Les processus de collaboration peuvent être intégrés dans des pistes. Les interactions des partenaires impliqués sont décrites dans des corridors spécifiques. De cette façon, les processus peuvent être modélisés séparément ou au sein d'un diagramme BPMN global. Lorsqu'une collaboration apparaît dans le même diagramme que l'un de ses processus internes, leurs activités communes peuvent être associées les unes aux autres.

À partir de ces trois classes de processus, il est possible de créer différents types de processus d'entreprise :

- Private business processes de haut niveau
- Private business processes de niveau détaillé (processus requis ou réels)
- Flux entre processus détaillés et entités externes
- Flux entre processus détaillés
- Flux entre processus détaillés et processus abstraits
- Flux entre processus détaillés et collaboration processes
- Flux entre processus abstraits
- Flux entre processus abstraits et collaborations
- Flux entre collaborations
- Flux entre plusieurs processus détaillés par le biais de leurs processus abstraits
- Flux entre plusieurs processus détaillés par le biais d'une collaboration

- Flux entre plusieurs processus détaillés par le biais de leurs processus abstraits et d'une collaboration

La figure ci-après représente un diagramme de collaboration BPMN avec deux partenaires commerciaux auxquels un processus est affecté. Les deux processus détaillés se composent d'un événement de départ, d'activités, de liaisons de flux de séquence et d'un événement de fin. Des liaisons de flux de message sont modélisées entre les activités des deux processus détaillés.

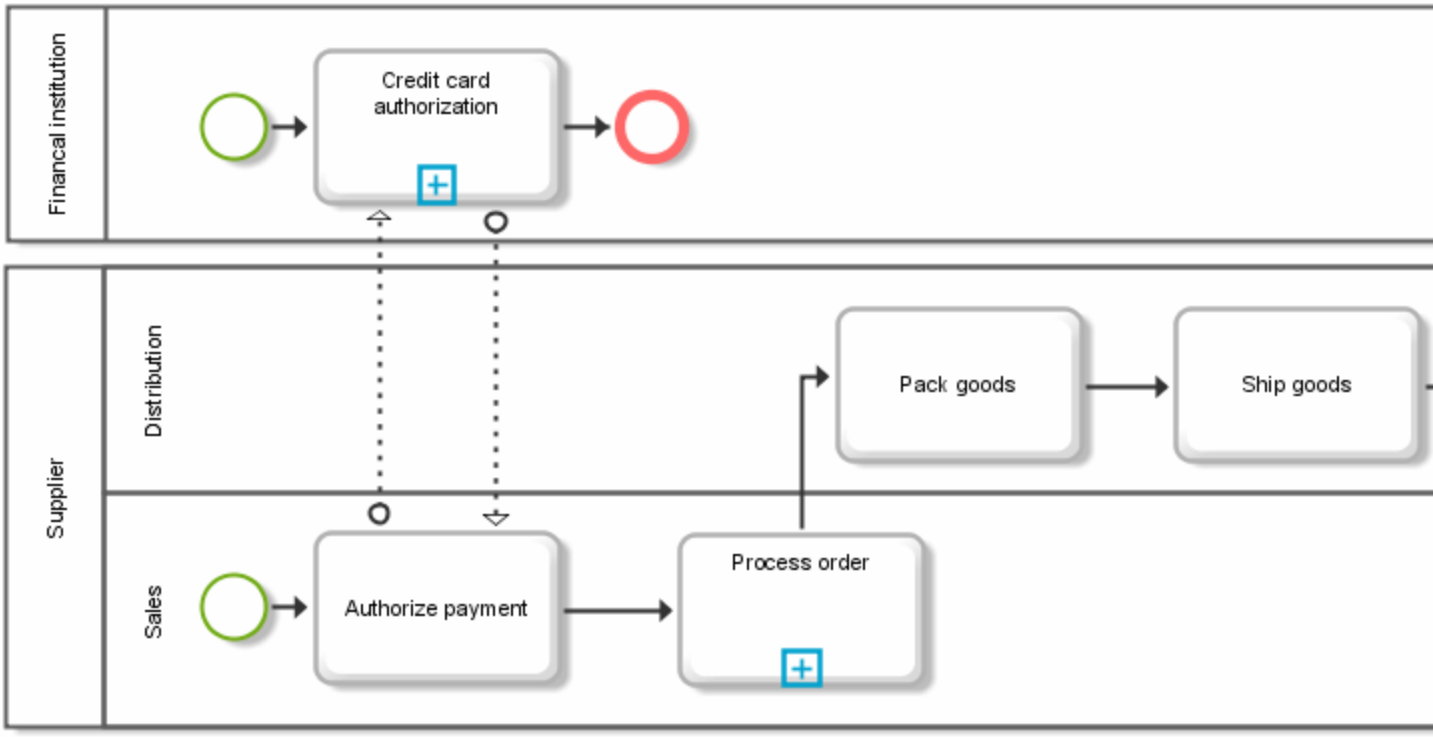


Illustration 142: Deux partitions avec un flux de séquence et un flux de message

Comme plusieurs partenaires commerciaux peuvent être représentés au sein d'un processus de diagramme de collaboration BPMN et que chaque partenaire a une vue différente d'un même processus, il est utile d'indiquer le point de vue. La spécification BPMN n'indique pas comment mettre en évidence le point de vue dans un diagramme de collaboration BPMN. La solution la plus simple consiste à gérer les noms des business entities (partenaires commerciaux) affectés dans l'attribut **Description/Définition** (cf. figure).

9.2 Réalisation de BPMN dans ARIS

Bien que le type de modèle **Business process diagram (BPD)** soit le seul prévu dans la spécification BPMN, deux types de modèles peuvent être utilisés dans ARIS : la CPE et le nouveau type de modèle **Business process diagram (BPD)**. De la sorte, les processus existants peuvent être réutilisés comme private processes (processus privés) dans ARIS. La CPE possède tous les attributs de modèle prévus par la spécification BPMN pour le BPD. Si

vous utilisez malgré tout le type de modèle Business process diagram (BPD), les modèles de type CPE existants ne reprennent aucun des aspects du contexte B2B. Ainsi, la complexité des modèles CPE n'est pas accentuée lorsque vous ajoutez des types de relations supplémentaires.

Le nouveau BPD hérite à son tour de tous les attributs de modèle de la CPE pertinents pour le langage BPMN et de tous les objets, liaisons et symboles pertinents pour le flux de séquence. Dans le nouveau type de modèle Business process diagram (BPD), des concepts pertinents pour le flux de séquence de la CPE peuvent être réutilisés. En outre, des pistes, des corridors et des flux de message peuvent également être représentés.

9.3 Les éléments du business process diagram

9.3.1 Pistes et corridors

Un diagramme de processus d'entreprise est structuré à l'aide de pistes.

Une piste est un conteneur graphique dans lequel est résumé un jeu d'activités d'un business entity.

Un business entity peut être une fonction, une application, un élément organisationnel (qui englobe l'unité organisationnelle, le type d'unité organisationnelle, le groupe, le type de personne, le poste, la personne, le site, l'unité organisationnelle système et le type d'unité organisationnelle système) **ou un élément de données** (qui englobe le terme spécifique, le cluster/modèle de données, le type d'entité, le type de relation, l'attribut MER, l'objet d'affaires, le type d'objet complexe, l'attribut Type d'objet complexe, la classe et le support d'information).

Dans le langage BPMN, deux pistes représentent deux business entities différents. La structuration d'un modèle en pistes est généralement utilisée dans le contexte B2B.

Une piste lie les différentes activités d'un partenaire de processus qui sont structurées et organisées à l'aide de corridors. De la sorte, les activités des partenaires de processus sont délimitées (voir la figure **Deux pistes avec un flux de séquence et un flux de message** (page 173)).

Dans un BPD, une piste ne doit pas nécessairement contenir des éléments de processus. Vous pouvez également insérer une piste vide (« black box ») dans un modèle lorsque vous souhaitez, par exemple, intégrer dans un processus global les liens d'un sous-processus impliqué, mais dont les détails ne sont pas connus (par exemple, un sous-processus d'un partenaire commercial). Vous pouvez aussi renoncer à représenter les détails d'un sous-processus par souci de simplicité (voir la figure **Processus de vote électronique** (page 188)).

Les pistes sont composées d'au moins un corridor. Un corridor peut lui-même contenir des corridors imbriqués ou définis comme une matrice. Si une piste ne comporte qu'un seul

corridor, la piste porte le même nom que le corridor. Lorsqu'une piste comporte plusieurs corridors, vous devez indiquer les différents noms de corridor et un nom de piste spécifique.

Illustration 143: Partitions avec deux sous-partitions en langage BPMN

Dans ARIS, les pistes et les corridors sont des types d'objets distincts qui doivent d'abord être placés dans le modèle. Au sein de la piste, le processus peut alors être modélisé comme une CPE. Tous les événements ainsi que toutes les fonctions et règles du processus sont placés sur l'objet de piste. La liaison **appartient à** peut être utilisée pour décrire l'affectation de ces objets à une piste. Il est recommandé de la créer en tant que relation implicite. L'objet de piste est mis en relation avec un élément organisationnel, un objet de type Type d'application, un élément de données ou une fonction via la relation de type **illustre**. Tenez compte du fait qu'une piste ne peut posséder qu'une liaison de ce type dans toute la base de données. Ces relations doivent également être créées de manière implicite.

D'après le langage BPMN, une piste ne doit pas être représentée par un symbole dans le modèle. Il est également possible de masquer les limites d'**une** piste, en particulier lorsque le diagramme n'en contient qu'une seule (voir la figure **Processus de vote électronique** (page 188)). Pour simplifier la lisibilité d'un diagramme comportant plusieurs pistes, il est recommandé de renoncer à utiliser ces options.

9.3.2 Directives de modélisation pour les pistes et les corridors

- Au sein d'un diagramme, il ne peut y avoir qu'une seule piste dont les limites ne sont pas visibles.
- Lorsque l'attribut **Pool type** est réglé sur **Collaboration**, vous ne devez pas gérer de propriétaire (attribut **Responsable**).
- Un corridor ne peut avoir qu'une seule piste supérieure.

9.3.3 Flux de séquence

Un processus sous forme de flux de séquence décrit l'ordre dans lequel les activités d'un processus sont exécutées. Le flux de séquence relie entre eux les types d'objets **Evénement**, **Activité** et **Branchement**. Les flux de séquence ne sont autorisés que dans les limites d'une piste et ne peuvent pas franchir ces dernières (voir la figure suivante).

Le flux de séquence est représenté par une ligne continue pourvue d'une flèche noire :

Illustration 144: Liaison de flux de séquence

En fonction du type des objets source et cible de la liaison, les types de liaisons correspondants sont gérés (par ex. active, est évalué par, génère, crée une liaison avec ou conduit à).

9.3.4 Directives de modélisation pour les liaisons de flux de séquence

- Pour les flux de séquence qui suivent un branchement du type XOR (basé sur des données) ou un branchement inclusif, une valeur doit être définie pour l'attribut **Condition**.
- Lorsque l'attribut **Condition** a la valeur **Expression**, vous devez placer le diamant comme symbole au début de la liaison.
- Lorsque l'attribut **Condition** a la valeur **Default** et que l'objet source est une fonction, vous devez placer une barre oblique inverse **** comme symbole au début de la liaison.
- La barre oblique inverse **** ne peut pas être placée lorsque l'objet source est un branchement.

- Lorsque l'objet source correspond à l'un des symboles suivants, vous ne devez pas définir de condition :
  - Branchement événementiel
  - Branchement complexe
  - Branchement parallèle
  - Événement de départ
  - Événement intermédiaire
- Lorsque la valeur **Default** de l'attribut **Condition** est activée pour une liaison de flux de séquence, aucune condition ne peut être gérée.
- L'attribut **Condition** peut avoir la valeur **Default** si l'objet source est une fonction ou un branchement XOR (événementiel).
- Lorsque la valeur **Expression** est définie pour l'attribut **Condition**, l'attribut **Expression** doit également être géré.

9.3.5 Flux de message

Un flux de message décrit l'échange d'informations entre deux pistes. Le flux de message peut se trouver soit directement entre les deux objets de piste soit entre des objets dans le flux de séquence des processus de la piste. Seuls les flux de message peuvent franchir les limites de piste et une liaison de flux de message ne peut pas être créée entre deux objets de la même piste (cf. la figure **Deux pistes avec un flux de séquence et un flux de message** (page 173)).

La liaison est représentée par une ligne pointillée. Un cercle apparaît en début de ligne et une flèche blanche à la fin.

Illustration 145: Liaison de flux de message

Chaque flux de message est composé d'un objet d'envoi, d'une liaison de type **envoie**, d'une liaison de type **est reçu par** et du destinataire. Aucune liaison de flux de message ne part d'un événement de départ ni d'un événement intermédiaire. En outre, un événement de fin ne reçoit pas de flux de message, mais peut en envoyer. Les corridors, les branchements, les objets de données et les annotations de texte sont exclus des flux de message.

9.3.6 Directives de modélisation pour les liaisons de flux de message

L'objet source et l'objet cible doivent appartenir à des partitions différentes.

9.3.7 Association

Une association permet d'associer des informations aux composants d'un flux de séquence ou d'un flux de message. Ces informations peuvent être textuelles ou graphiques. Lorsque plusieurs processus différents apparaissent dans le même diagramme, les différents éléments de processus qu'ils contiennent peuvent être associés à l'aide de liaisons.

Une association est représentée par une ligne pointillée.Vous pouvez y adjoindre des pointes de flèches ouvertes, si nécessaire. C'est notamment le cas pour les affectations d'artéfacts du type **Objet de données**.

Illustration 146: Liaisons d'association

En fonction du type des objets source et cible de la liaison, les types de liaisons correspondants sont gérés (par exemple, **a pour sortie**, **est l'entrée pour**, **fournit l'entrée pour** ou **génère une sortie sur**).

L'affectation d'artéfacts de type Objet de données à des activités est particulièrement importante dans la spécification BPMN.

Cette affectation est orientée et décrit comment les informations sont utilisées et modifiées au sein d'un processus. Elle est réalisée dans le BPD (BPMN) à l'aide des relations suivantes :

Fonction - génère une sortie sur - Eléments de données (notamment, des supports d'informations)

Elément de données (notamment, des supports d'informations) - fournit l'entrée pour - Fonction

9.3.8 Événements

Un événement (event) est un état qui survient lors du déroulement du processus d'entreprise. Les événements influencent le déroulement du processus. En général, ils représentent des déclencheurs ou des effets au sein du processus. En fonction du moment où un événement survient, on distingue trois catégories d'événements : les événements de départ, les événements intermédiaires et les événements de fin. Ces trois catégories d'événements sont représentées par des symboles différents dans le langage BPMN :

Illustration 147: Catégories d'événements

Au sein de ces catégories, on distingue encore des sous-catégories. Les symboles des trois catégories d'événements peuvent être étendus à l'aide de symboles supplémentaires par le biais de l'attribut **Type d'événement**, comme le montrent les trois exemples qui suivent :

Illustration 148: Exemples de types d'événements

Tous les attributs pertinents pour le type d'objet **Evénement** sont rassemblés dans le groupe de types d'attributs **BPMN**.

9.3.9 Directives de modélisation pour les événements

- Pour les événements de départ (start events), le type d'attribut **Type d'événement** ne peut prendre qu'une des valeurs suivantes : **Message**, **Minuterie**, **Règle**, **Lien** ou **Multiple**.
- Pour les événements de fin (end events), le type d'attribut **Type d'événement** ne peut prendre qu'une des valeurs suivantes : **Message**, **Exception**, **Annulation**, **Compensation**, **Règle**, **Lien**, **Multiple** ou **Arrêt**.
- Pour les événements intermédiaires (intermediate events), le type d'attribut **Type d'événement** ne peut prendre qu'une des valeurs suivantes : **Message**, **Minuterie**, **Exception**, **Annulation**, **Compensation**, **Règle**, **Lien** et **Multiple**.
- Les attributs correspondants doivent être gérés avec des informations supplémentaires qui varient en fonction du type d'événement défini.
- Un événement de départ peut posséder plusieurs liaisons de flux de séquence sortantes. Aucune valeur ne peut être définie pour l'attribut **Condition** de ces liaisons.
- Les événements intermédiaires qui affichent une valeur Exception ou une valeur Compensation doivent être placés au bord de la fonction.
- Si un événement intermédiaire est placé au bord d'une fonction, une valeur, mais pas la valeur **Lien**, doit être gérée.
- Les valeurs **Multiple**, **Règle** et **Annulation** ne peuvent pas être définies pour les événements intermédiaires qui se trouvent dans un flux de séquence normal d'un processus.
- La valeur **Annulation** ne peut pas être définie, si
  - l'événement intermédiaire est placé au bord d'une fonction et si l'attribut **Transaction** de la fonction n'est pas activé ou
  - l'événement ne fait pas partie d'un processus qui décrit une transaction.
- Si un événement intermédiaire est placé au bord d'une fonction, il ne peut pas être l'objet cible d'une liaison de flux de séquence.
- Si un événement intermédiaire se situe à l'intérieur du flux de séquence normal d'un processus, c.-à-d. qu'il n'est pas placé au bord d'une fonction, il peut posséder une seule liaison de flux de séquence entrante. Pour l'attribut **Type d'événement** de l'événement, seule une des valeurs suivantes peut être gérée : **Message**, **Minuterie**, **Exception**, **Lien**, **Compensation**.
- La valeur **Lien** peut être définie pour les événements intermédiaires dans le flux de séquence normal, si l'objet source est un branchement ayant un attribut **Type de branchement** pour lequel la valeur **XOR (événementiel)** est gérée.

- Chaque événement intermédiaire doit posséder une seule liaison de flux de séquence sortante.
- Un événement intermédiaire ayant un attribut **Type d'événement** pour lequel la valeur **Message** est gérée, peut posséder un flux de message entrant (liaison entrante de type **est reçu par**).
- Un événement intermédiaire ne peut pas posséder de flux de message sortant (liaison sortante de type **envoie**).

9.3.10 Activités

Une activité est une tâche qui est exécutée au sein d'un processus. Une activité peut être atomique ou non atomique (composée). La spécification BPMN reconnaît trois catégories d'activités : les processus (process), les sous-processus (subprocess) et les tâches (task).

Le langage BPMN représente les activités à l'aide des symboles suivants :

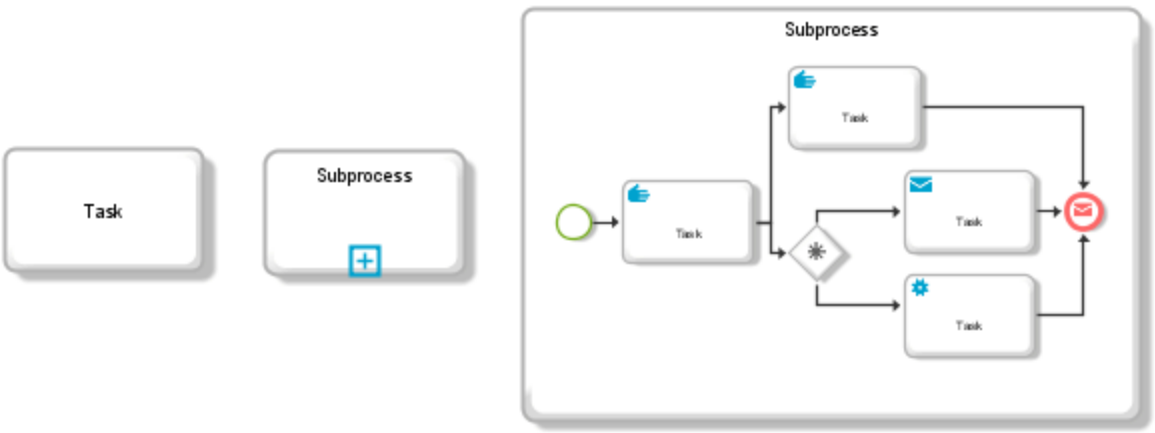


Illustration 149: Activités en langage BPMN

Dans ARIS, les activités sont représentées par défaut par des fonctions :

Illustration 150: Fonction associée en tant qu'activité dans ARIS

La fonction reçoit tous les attributs prévus pour les processus, les sous-processus et les tâches dans la spécification BPMN. Comme pour les événements, le groupe de types d'attributs **BPMN** est utilisé ; il contient des sous-groupes supplémentaires pour les différents types d'activités.

Au niveau du BPMN, un processus décrit une activité qui est effectuée au sein d'une entreprise ou d'une organisation. Un processus est décrit à l'aide d'un graphe contenant des

objets de flux représentant un ensemble d'autres activités et objets de commande. Les processus sont structurés hiérarchiquement et peuvent ainsi être définis à chaque étape de spécification. Contrairement à un processus, un processus d'entreprise (business process) décrit dans le BPMN un ensemble d'activités qui sont exécutées au-delà des frontières de l'entreprise/organisation.

Au niveau du BPMN, un sous-processus est une activité composée qui possède une description détaillée. Un sous-processus apparaît sous forme d'objet au cours d'un déroulement de processus (Process flow).

En principe, un processus détaillé est associé à un sous-processus. Dans ARIS, une activité associée n'est pas identifiée par un signe plus, comme dans la spécification BPMN, mais aussi par le symbole d'association.

En plus de l'identification d'une fonction associée, la spécification BPMN permet d'afficher le processus détaillé au niveau du processus supérieur suivant. Pour ce faire, cliquez sur le signe plus.

9.3.11 Directives de modélisation pour les activités

PROCESSUS

- En présence de l'attribut **Ad hoc = True**, vous devez gérer l'attribut **Completion condition**.
- Lorsqu'un processus Ad hoc est affiné, vous ne pouvez pas modéliser de flux de séquence au sein du modèle associé.

SOUS-PROCESSUS

- Lorsque la valeur **Indépendant** est utilisée pour l'attribut **Type de sous-processus**, l'attribut **Référence de processus** doit également être géré.
- Lorsque l'attribut **Transaction** est activé pour un sous-processus, l'attribut **ID de transaction** doit également être géré.
- Lorsque l'attribut **Type de boucle** est géré, l'attribut **Condition de boucle** doit également être géré.
- Si les modèles doivent être transférés vers BPEL4WS, vérifiez que l'attribut **Maximum** est géré pour les processus dont l'attribut **Type de boucle** est défini sur **Standard**.
- Lorsque la valeur **Standard** est gérée pour l'attribut **Type de boucle**, l'attribut **Test préalable** doit également être géré. L'attribut **Test préalable** doit être désactivé par défaut.
- Lorsque la valeur **Multi instance** est gérée pour l'attribut **Type de boucle**, l'attribut **Parallel instance generation** doit également être géré. L'attribut **Parallel instance generation** doit être désactivé par défaut.

- Si l'attribut **Type de boucle** d'un sous-processus possède la valeur **Multi instance** et si, en même temps, l'attribut **Parallel instance generation** est activé, l'attribut **Loop flow condition** doit également être géré.
- Lorsque la valeur **Complexe** est définie dans un processus pour l'attribut **Loop Flow Condition**, vous devez gérer une expression pour l'attribut **Complexe** qui détermine le nombre de marques de processus devant être transmis au sous-processus et le moment auquel cette transmission doit avoir lieu.

TÂCHE

- Lorsque la valeur **Réception** est gérée pour l'attribut **Type de tâche**, la fonction ne doit pas posséder de liaison de flux de message sortante.
- Lorsque la valeur **Envoi** est gérée pour l'attribut **Type de tâche**, la fonction ne doit pas posséder de liaison de flux de message entrante.
- Si l'attribut **Type de tâche** n'est pas géré ou si les valeurs **Script** ou **Manuel** sont définies, la fonction ne doit posséder aucune liaison de flux de message sortante et entrante.
- Pour les fonctions dont l'attribut **Type de tâche** est défini sur **Abstrait**, l'attribut **Type abstrait** doit également être géré. En outre, elles ne peuvent être gérées que dans des partitions de type **Abstrait** ou dans des **collaborations**.

9.3.12 Branchement

Les branchements décrivent la façon dont les flux de séquence convergent ou se ramifient au sein d'un processus. Ils déterminent le comportement des liaisons entrantes et sortantes. Dans ARIS, ils sont représentés sous la forme d'objets du type **Règle**.

Comme les événements, les branchements peuvent être de différents types. En fonction du type, des symboles supplémentaires sont représentés au centre du symbole de branchement.

Quelques symboles de branchements :

Illustration 151: Types de branchements

La spécification BPMN permet de définir un certain nombre de portes (gates) pour chaque branchement. Dans ARIS, le nombre de portes est déterminé par le nombre de liaisons entrantes et sortantes. C'est pourquoi les attributs qui dépendent des portes sont gérés dans les liaisons de flux de séquence entrantes et sortantes de la règle.

Le branchement complexe constitue un cas à part car les attributs spéciaux **Condition entrante** et **Condition sortante** y sont spécifiés. Ces attributs doivent être gérés lorsqu'il y a plusieurs liaisons de flux de séquence entrantes ou sortantes dans le branchement. Le contenu de l'attribut Condition entrante peut contenir des noms de flux de séquence et des propriétés de processus (données). L'attribut Condition sortante contient des références à des ID de flux de séquence et des propriétés de processus (données).

9.3.13 Directives de modélisation pour les branchements

- Branchements du type XOR (basé sur des données) : pour toutes les liaisons sortantes d'un branchement du type XOR (basé sur des données), la valeur **Expression** doit être définie pour l'attribut **Condition** et une expression valide doit être utilisée pour l'attribut **Expression de condition**.

Flux de séquence, en particulier après les branchements

- Pour chaque branchement XOR du type **XOR (basé sur des données)**, l'attribut **Branchement par défaut** doit être géré sur exactement une liaison de flux de séquence sortante (type de liaison **active**). Il est impossible de définir plusieurs liaisons sortantes avec cet attribut.
- Chaque branchement XOR du type **XOR (événementiel)** doit disposer d'au moins deux liaisons de flux de séquence sortantes (type **active** ou **conduit à**).
- Pour toutes les liaisons sortantes d'un branchement du type XOR (événementiel), aucune valeur ne doit être gérée pour l'attribut **Condition**. L'attribut **Expression de condition** ne doit pas être géré.
- Les objets cible suivants sont autorisés pour les liaisons de flux de séquence sortantes d'un branchement du type XOR (événementiel) :
  - Fonction pour laquelle le type de tâche **Réception** est défini.
  - Événements intermédiaires dont le type d'attribut **Type d'événement** possède une valeur autre que **Compensation** ou **Multiple**.
- Si une fonction figure parmi les objets cible, ces derniers ne peuvent pas comprendre d'événement de type **Message**.
- Si un branchement de type **OR** possède une seule ou aucune liaison de flux de séquence entrante, au moins deux liaisons de flux de séquence sortantes doivent exister.
- Pour toutes les liaisons de flux de séquence sortantes d'un branchement de type OR, la valeur **Expression** doit être définie pour l'attribut **Condition** et une expression valide doit être utilisée pour l'attribut **Expression de condition**. L'expression doit être univoque au sein du branchement.

- Si un branchement OR possède une seule liaison de flux de séquence sortante, aucune valeur ne doit être gérée pour l'attribut **Condition** de cette liaison.
- Si un branchement de type **Complexe** possède une seule ou aucune liaison de flux de séquence entrante, au moins deux liaisons de flux de séquence sortantes doivent exister.
- Pour toutes les liaisons sortantes d'un branchement complexe, la valeur **Aucun** doit être gérée pour l'attribut **Condition**, en particulier lorsqu'il n'y a qu'une liaison sortante.
- Lorsqu'un branchement complexe possède plusieurs liaisons de flux de séquence entrantes, une condition contenant les noms de flux de séquence et les propriétés de processus (données) doit être gérée pour l'attribut **Condition entrante**.
- Lorsqu'un branchement complexe possède plusieurs liaisons de flux de séquence sortantes, une condition contenant les noms de flux de séquence et les propriétés de processus (données) doit être gérée pour l'attribut **Condition sortante**.
- Si un branchement de type AND possède une seule ou aucune liaison de flux de séquence entrante, au moins deux liaisons de flux de séquence sortantes doivent exister.
- Pour toutes les liaisons sortantes de flux de séquence d'un branchement AND, aucune valeur ne peut être gérée pour l'attribut **Condition**.

9.3.14 Artifact

Les artefacts fournissent des informations sur le processus qui n'appartiennent ni au flux de séquence ni au flux de message. On distingue trois types d'artefacts : **Objets de données**, **Groupes** et **Annotations** (la liste des types peut être librement élargie).

Les **objets de données** sont comparables au support d'information ou aux éléments de données dans ARIS. Mais, au sens le plus large, ils peuvent comprendre toutes les affectations. Les objets de données n'influencent ni le flux de séquence ni le flux de message, mais ils fournissent des informations sur ce qui se passe pendant le processus. Ils donnent des renseignements sur la façon dont les documents, données et objets sont modifiés pendant le processus.

Un **groupe** permet de mettre en relief des éléments de processus qui forment un groupe. Dans ARIS, on trouve à cet effet des objets graphiques comme le rectangle ou la ligne brisée.

Des groupements peuvent également être pris en considération. Toutefois, cela n'est utile que si le groupement comprend un graphique.

Les **annotations** correspondant à des remarques sur les objets ou les liaisons, comme dans l'exemple suivant **Time out [1week]**. Dans ARIS, elles sont souvent représentées par l'attribut **Remarque/Exemple**. Il est important que cet attribut soit placé dans le modèle comme décrit dans l'exemple suivant pour **Yes** et **No**.

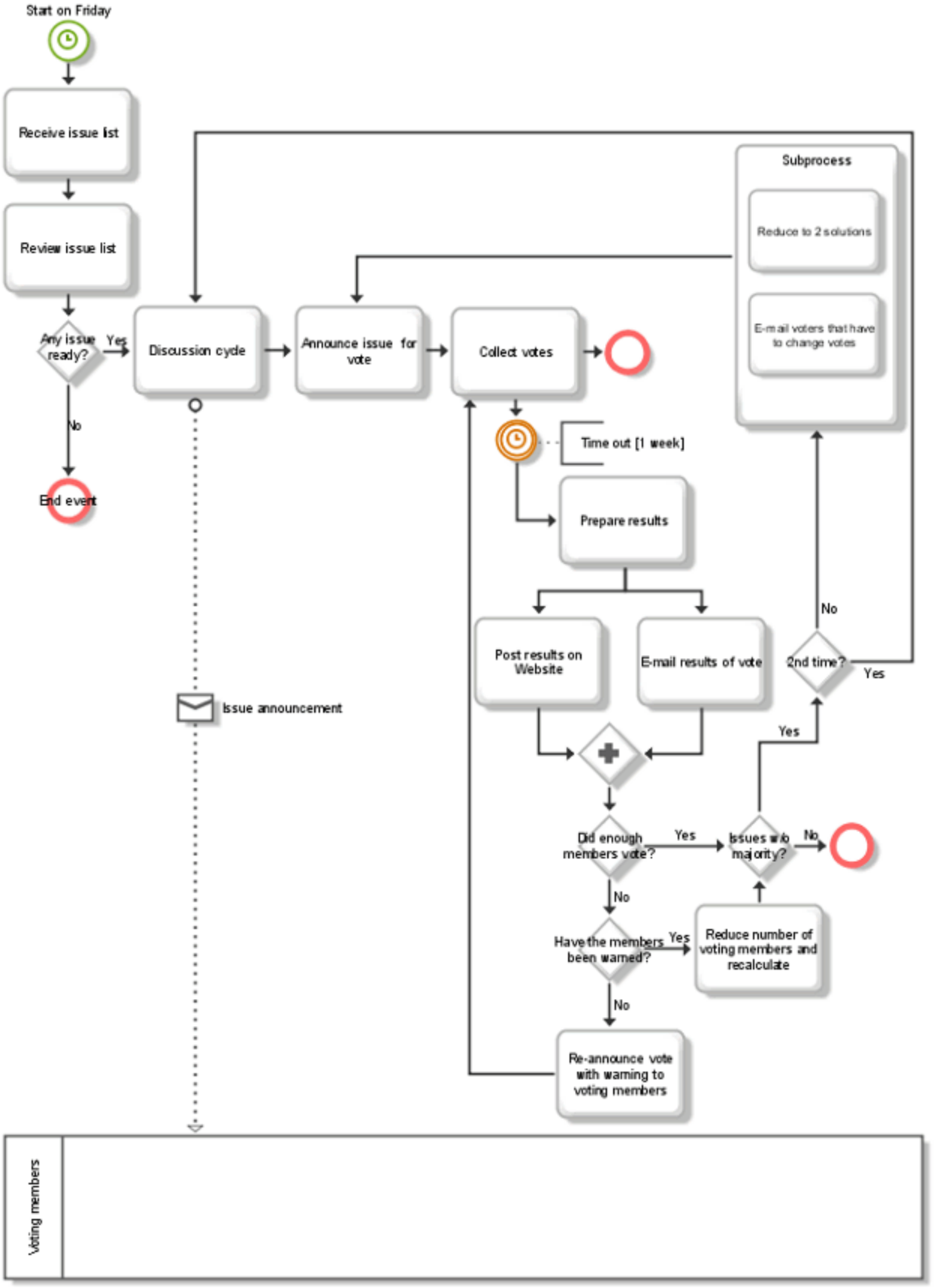


Illustration 152: Processus de vote électronique

Cette illustration montre un exemple de réalisation d'un diagramme de collaboration d'entreprise (business collaboration diagram) selon BPMN dans ARIS. Le diagramme comprend deux pistes ; les bords de la piste supérieure sont masqués. La piste inférieure ne présente pas les divers éléments de processus.

9.3.15 Sources des figures

Figure **Deux pistes avec un flux de séquence et un flux de message** (page 173) :

Business Process Modeling Notation, Working Draft (1.0) ; BPMI.org ; 25 août 2003 ; page 85.

Figure **Piste avec deux corridors en langage BPMN** (page 176) :

Business Process Modeling Notation, Working Draft (1.0) ; BPMI.org ; 25 août 2003 ; page 87.

Figure **Catégories d'événements** (page 181) et figure **Exemples de types d'événements** (page 181) :

Business Process Modeling Notation, Working Draft (1.0) ; BPMI.org ; 25 août 2003 ; page 27.

Figure **Activités en langage BPMN** (page 183) :

Business Process Modeling Notation, Working Draft (1.0) ; BPMI.org ; 25 août 2003 ; page 28.

Figure **Types de branchements** (page 185) :

Business Process Modeling Notation, Working Draft (1.0) ; BPMI.org ; 25 août 2003 ; page 28.

10 Modeling BPMN 2.0

10.1 Introduction

10.1.1 Initial situation and objective

BPMN (Business Process Modeling and Notation) has emerged as a widely adopted standard for process modeling. Its popularity is based on the fact that it is has been developed by the Object Management Group (OMG), a consortium of organizations that also released other important modeling standards like UML.

The primary goal of BPMN is to provide a notation that is understandable by all users: business analysts designing and documenting business processes, developers implementing these business processes, and business end users executing, managing and monitoring their business processes. The most recent version released by the OMG is BPMN 2.0. This is the standard supported by ARIS. In a first step, the objective is to focus on process modeling conformance, one of four conformance types defined by the OMG.

The four conformance types are described in detail in the BPMN specification: Business Process Model and Notation (BPMN), Version 2.0.

10.1.2 Purpose of this chapter

Unfortunately the BPMN specification has increased an order of magnitude in technical complexity and fails to distinguish those elements needed for business process modeling from those required for process execution.

The purpose of this chapter is to describe the ARIS implementation of the BPMN 2.0 elements that are part of business process modeling documenting the process flow. Those parts that are needed for executable design are ignored. The elements relevant for business process modeling are essentially those displayed in a diagram.

The mapping described in the chapters of this document is based on the BPMN specification **Business Process Model and Notation (BPMN)**, **version 2.0** (http://www.bpmn.org).

The attribute and model association tables are also taken from the BPMN 2.0 specification and extended to describe the implementation in ARIS.

10.2 BPMN core elements and their implementation in ARIS

The BPMN core consists of four packages:

- Foundation,
- Infrastructure,
- Common Elements,
- Service.

It provides the basis for modeling processes, collaborations, choreographies and conversations. These packages are described in detail in chapter 8 of the BPMN specification.

In the following sections the core constructs and their attributes and associations are mapped to ARIS constructs.

10.2.1 Infrastructure

The infrastructure package consists of two elements that are particularly relevant for import and export. Thus, their attributes and model associations are not included in the current version of the BPMN 2.0 implementation.

See: Business Process Model and Notation (BPMN), Version 2.0.

10.2.2 Foundation

The foundation package contains classes that are shared amongst other packages in the BPMN core. The foundation package consists of eight classes: BaseElement, Documentation, RootElement, Extension, ExtensionDefinition, ExtensionAttributeDefinition, ExtensionAttributeValue, and Relationship. See: Business Process Model and Notation (BPMN), Version 2.0.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
BaseElement	id: string	The ARIS GUID of the corresponding modeling construct represents the BPMN ID. For imported BPMN elements an attribute type in the attribute type group **Attributes of external system** will be used.
	documentation: Documentation [0..*]	see below: Documentation
	extensionDefinitions: ExtensionDefinition [0..*]	Méthode ARIS can be enhanced, for example, by user-defined attributes.
	extensionValues: ExtensionAttributeValue [0..*]	The Méthode ARIS can be enhanced, for example, by user-defined attributes.
Documentation	inherits from BaseElement	

Class	BPMN attribute name	Implementation in ARIS
	text: string	All ARIS attribute types assigned to model types, object types, and connection types can be used for documentation purposes. The attribute types Description/Definition (AT_DEC) and Remark/Example (AT_REM) should be used to provide general information. Specific attribute types should be used to store specific information.
Extension	mustUnderstand: boolean [0..1] = False	Currently not implemented.
	definition: ExtensionDefinition	
ExtensionDefinition	name: string	Currently not implemented.
	extensionAttributeDefinitions: ExtensionAttributeDefinition [0..*]	
ExtensionAttributeDefinition	name: string	Currently not implemented.
	type: string	
	isReference: boolean [0..1] = False	
ExtensionAttributeValue	value: [Element [0..1]	Currently not implemented.
	valueRef: [Element [0..1]	
	extensionAttributeDefinition: ExtensionAttributeDefinition	
Relationship	inherits from BaseElement	Currently not implemented.
	type: string	

Class	BPMN attribute name	Implementation in ARIS
	direction: RelationshipDirection {none | forward | backward | both}	
	sources: [Element [1..*]	
	targets: [Element [1..*]	
RootElement	inherits from BaseElement	RootElement is an abstract class, it has no direct representation in ARIS. For example, ARIS object types are root elements, ARIS attribute types are not.

10.2.3 Common Elements

Common Elements are basic elements that may be used in more than one type of diagram, for example, Process, Collaboration, Conversation, and Choreography. The Common Elements are categorized into seventeen different groups.

10.2.3.1 Artifacts

Artifacts are used to depict additional information in a BPMN process diagram (BPMN2.0) or BPMN collaboration diagram (BPMN2.0) that is not directly related to the sequence flow or message flow. BPMN 2.0 provides three standard artifacts:

- Associations,
- Groups, and
- Text annotations

Data objects are no longer artifacts, they are concepts of their own (see chapter Items and Data (page 244)).

See: Business Process Model and Notation (BPMN), version 2.0.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Association	inherits from BaseElement	various connection types
	associationDirection: AssociationDirection = None {None | One | Both}	This attribute is represented by the direction and the style of the corresponding ARIS connection type.
	sourceRef: BaseElement	Corresponds to the source object type of the connection type representing the association.
	targetRef: BaseElement	Corresponds to the target object type of the connection type representing the association.

Class	BPMN attribute name	Implementation in ARIS
Group	inherits from BaseElement	Object type: Structural element (OT_STRCT_ELMT) Symbol: Structural element in model type **Structuring model** (MT_STRCT_DGM) Symbol: Group (ST_BPMN_GROUPING_1)
	categoryValueRef: CategoryValue [0..1]	Attribute type **Name** (AT_NAME) of object type Structural element (OT_STRCT_ELMT)
Category	inherits from BaseElement	Object type Structural element (OT_STRCT_ELMT) in model type Structuring model (MT_STRCT_DGM)
	categoryValue: CategoryValue [0..*]	Connection type in model type Structuring model: * Structural element (representing the category) contains structural element (representing the category value).
CategoryValue	inherits from BaseElement	Object type: Structural element (OT_STRCT_ELMT) Symbol: Structural element in model type Structuring model (MT_STRCT_DGM) Symbol: Group (ST_BPMN_GROUPING_1) in BPMN 2.0 diagrams
	value: string	Attribute type **Name** of object type Structural element
	category: Category [0..1]	Connection type in model type Structuring model: * Structural element (representing the category) contains structural element (representing the category value).

Class	BPMN attribute name	Implementation in ARIS
	categorizedFlowElements: FlowElement [0..*]	Connection type **belongs to** [CT_BELONGS_TO_1] in the BPMN process diagram (BPMN 2.0) and BPMN collaboration diagram (BPMN 2.0): Target object type: Structural element (OT_STRCT_ELMT; ST_BPMN_GROUPING_1) Source object types: * Function (OT_FUNC) representing activities * Event (OT_EVT) * Rule (OT_RULE) representing Gateways * Cluster/data model (OT_CLST) representing data objects * Information carrier (OT_INFO_CARR) representing data stores
Text annotation	inherits from BaseElement	

Class	BPMN attribute name	Implementation in ARIS
	text: string	For object types in the BPMN process diagram (BPMN 2.0), BPMN collaboration diagram (BPMN 2.0), and BPMN conversation diagram (BPMN 2.0): * Text annotation (OT_BPMN_ANNOTATION) with symbol **Text annotation** (ST_BPMN_ANNOTATION_1) is associated with <target object type>. Target object types are all object types available in the corresponding model type. For connection types in the BPMN process diagram (BPMN 2.0), BPMN collaboration diagram (BPMN 2.0), and BPMN conversation diagram (BPMN 2.0): three attribute types in the attribute type group BPMN 2.0 attributes/Text annotation attributes: * Text annotation 1 (AT_BPMN_TEXT_ANNOTATION_1) * Text annotation 2 (AT_BPMN_TEXT_ANNOTATION_2) * Text annotation 3 (AT_BPMN_TEXT_ANNOTATION_3)

10.2.3.2 Association

Associations are used to associate information and artifacts with other BPMN elements. Thus, associations are (usually) represented by connection types in ARIS. The relevant connection types are described in the context of the object types being associated.

10.2.3.3 Group

BPMN 2.0 uses three different classes to represent groupings, but there is only one symbol: **Group**. Thus, a group is the graphical representation of a category value.

Categories and their category values are modeled in an auxiliary model of type **Structuring model**.

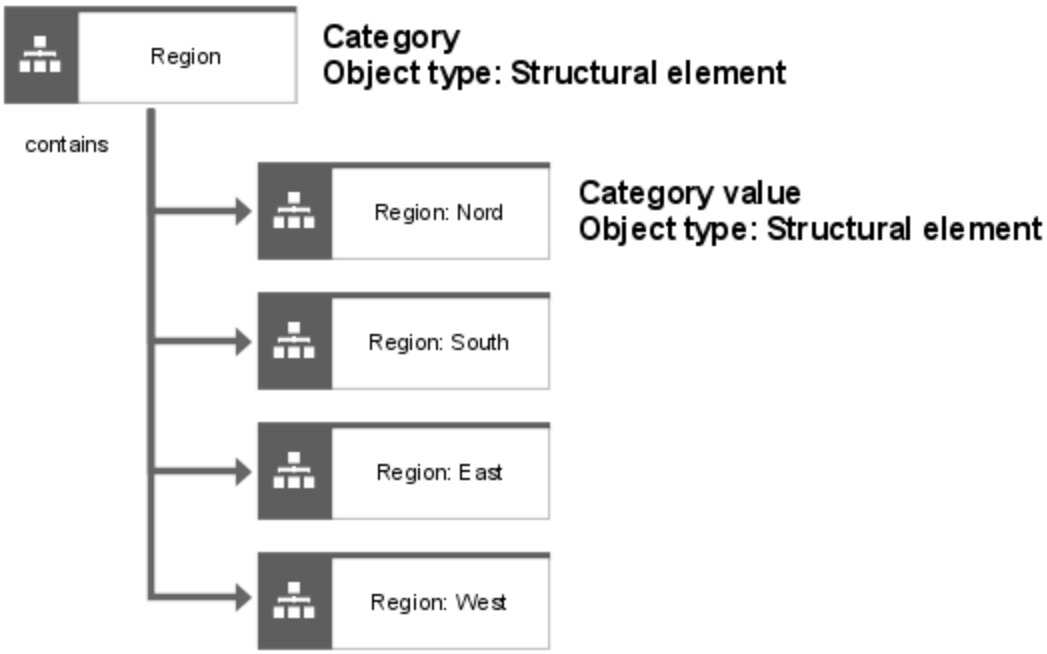


Illustration 153: Structuring model: Categories and their values

In ARIS the graphical element **Group** is an occurrence copy of a category value object and is depicted by a special symbol in the BPMN 2.0 models. The symbol name is **Group**.

Illustration 154: Group symbol

10.2.3.4 Text annotation

Text annotations are used to provide additional textual information for the reader of a BPMN model. They can be associated with graphical elements in a model, ARIS objects and connections.

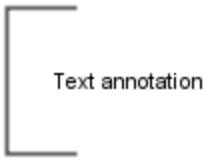


Illustration 155: Symbol representing text annotations

Text annotations are implemented in ARIS in 2 different ways:

TEXT ANNOTATIONS ASSOCIATED WITH ARIS OBJECTS

The object type **Text annotation** and the connection type **is associated with** is used to annotate objects (occurrences) in a model.

TEXT ANNOTATIONS ASSOCIATED WITH ARIS CONNECTIONS

Objects (here: Text annotation) cannot be assigned to connections. Thus, the program provides a new functionality: The modeler selects the text annotation symbol in the **Symbols** bar, places it on/near by the connection he/she wants to annotate and enters the text. The program draws a line looking like an association and stores the text in a **Text annotation** attribute of the corresponding connection. In the first step three text annotation attributes are provided in the attribute type group **BPMN 2.0 attributes/BPMN text annotations**:

Text annotation 1 (AT_BPMN_TEXT_ANNOTATION_1)

Text annotation 2 (AT_BPMN_TEXT_ANNOTATION_2)

Text annotation 3 (AT_BPMN_TEXT_ANNOTATION_3)

10.2.3.5 Callable Elements

Callable Element is an abstract class and has four specialized classes: Process, Global task, Choreography, and Choreography task. Only processes and global tasks are relevant for business process modeling compliance. They are represented by the object type Function.

See: Business Process Model and Notation (BPMN), version 2.0.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Callable Element	inherits from BaseElement	Object type: Function (OT_FUNC) Symbol: Call activity (ST_BPMN_CALL_ACTIVITY)
	name: string [0..1]	Attribute type Name (AT_NAME) of object type **Function** (OT_FUNC)
	supportedInterfacesRefs: Interface [0..*]	Currently not implemented.
	ioSpecification: InputOutputSpecification [0..1]	Currently not implemented.
	ioBinding: InputOutputBinding [0..*]	Currently not implemented.
InputOutputBinding	inputData: DataInput	Currently not implemented.
	outputData: DataOutput	Currently not implemented.
	operationRef: Operation	Currently not implemented.

10.2.3.6 Event

Events are described in detail in the context of the BPMN process diagram (see chapter Events (page 248)).

10.2.3.7 Expression

FormalExpressions belong to the execution design level and are not included in the current version of the BPMN 2.0 implementation.

However, natural-language expressions are used to allow the modeler to specify conditions. They are described in the context of the corresponding BPMN elements (object types and connection types).

10.2.3.8 Flow Element

Flow Elements are described in detail in the context of the BPMN process diagram (see chapter Process (page 219)).

See: Business Process Model and Notation (BPMN), version 2.0.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
FlowElement	inherits from BaseElement	No direct representation in ARIS -> abstract class
	name: string [0..1]	Attribute type **Name** (AT_NAME) of the object types representing flow nodes.
	auditing: Auditing [0..1]	Currently not implemented.
	monitoring: Monitoring [0..1]	Currently not implemented.

10.2.3.9 Flow Elements Container

A FlowElementsContainer is an abstract super class for BPMN diagrams (or views). So, Processes and Subprocesses as well as Choreographies and Choreography subprocess are FlowElementsContainers.

The specific attributes and model associations of a process and subprocess are described in detail in the context of the BPMN process diagram.

See: Business Process Model and Notation (BPMN), version 2.0.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
FlowElementsContainer	inherits from BaseElement	Model type **BPMN process diagram (BPMN 2.0)** (MT_BPMN_PROCESS_DIAGRAM)
	flowElements: FlowElement [0..*]	Occurrences of the object types and connection types allowed in a BPMN process diagram (BPMN 2.0).
	artifacts: Artifact [0..*]	Occurrences of the object types and attribute types representing groups and text annotations as well as their connection types allowed in the BPMN process diagram (BPMN 2.0).

10.2.3.10 Gateways

Gateways are described in detail in the context of the BPMN process diagram (see chapter Gateways (page 263)).

See: Business Process Model and Notation (BPMN), version 2.0.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Gateway	inherits from FlowElement	Object type: Rule (OT_RULE)
	gatewayDirection: GatewayDirection = unspecified { unspecified | converging | diverging | mixed }	The number of incoming and outgoing sequence flows depends on the modeling context, that is, the position of the gateway in the process. Thus, there is no ARIS attribute type representing the gateway direction. Gateways whose direction is unspecified or mixed should be avoided.

10.2.3.11 Message

Messages normally represent information exchanged between two participants in a BPMN collaboration diagram.

A message is represented by the symbol **Message** of the ARIS object type **Message**.

Illustration 156: Message symbol

See: Business Process Modeling Notation (BPMN), version 2.0, page 93 and the following.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Message	inherits from BaseElement	Object type: Message (OT_MSG_FLW) Symbol: Message (ST_BPMN_MESSAGE_2)
	name: string	Attribute type **Name** (AT_NAME) of object type Message (OT_MSG_FLW)
	structureRef : ItemDefinition [0..1]	Currently not implemented.

10.2.3.12 Message flow

The message exchange between participants is shown by a message flow that connects two pools or the objects within the pools.

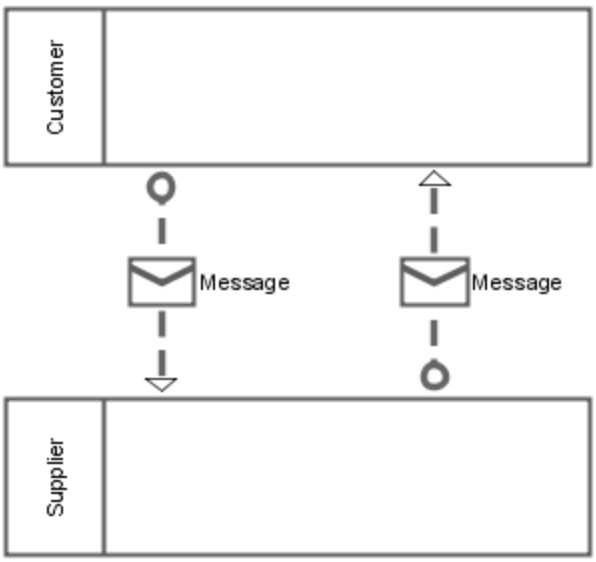


Illustration 157: Message flow between participants/pools

A message flow is represented in ARIS by the connection type **message flow**. If the message sent from one participant to another should be displayed in the diagram, the connection type **message flow** is replaced by the object type **Message** (symbol **Message**) and two connection types:

- <Source object type> sends message.
- Message is received from <target object type>.

More details can be found in chapter Message flow (page 273).

Message flow associations are used to map message flows modeled in two different diagrams, for example, in a conversation and a collaboration diagram. These associations are realized in ARIS by occurrence copies of the message flow connections.

Message flow is also described in the context of the BPMN collaboration diagram (chapter Message flow (page 273)) and the BPMN conversation diagram (chapter Message flow in a conversation (page 278)).

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
message flow	inherits from BaseElement	Connection type: message flow (CT_BPMN_MESAGE_FLOW)
	name: string	Attribute type **Connection role** of connection type **message flow** (CT_BPMN_MESSAGE_FLOW)
	sourceRef: MessageFlowNode	Source object type of connection type **message flow** (CT_BPMN_MESAGE_FLOW) (Participant, Function, Event)
	targetRef: MessageFlowNode	Target object type of connection type **message flow** (CT_BPMN_MESAGE_FLOW) (Participant, Function, Event)
	messageRef: Message [0..1]	Object type: Message (OT_MSG_FLW) Symbol: Message (ST_BPMN_MESSAGE_2) Connection types in the BPMN collaboration diagram (BPMN 2.0): * Participant sends (CT_SENDS_2) message. * Event sends (CT_SENDS_2) message. * Function sends (CT_SENDS_2) message. * Message is received from (CT_IS_RECEIVED_FROM) participant. * Message is received from (CT_IS_RECEIVED_FROM) function. * Message is received from (CT_IS_RECEIVED_FROM) event.
Flow node		Object types that can be the source or target of **message flow** (CT_BPMN_MESSAGE_FLOW) connection type: Participant (OT_BPMN_POOL), Function (OT_FUNC), Event (OT_EVT)
Message flow association	inherits from BaseElement	This association is used to map message flows modeled in a collaboration and a conversation diagram.
	innerMessageFlowRef: Message Flow	Occurrence copy of a message flow connection in a BPMN collaboration diagram and BPMN conversation diagram.

Class	BPMN attribute name	Implementation in ARIS
	outerMessageFlowRef: Message Flow	Occurrence copy of a message flow connection in a BPMN collaboration diagram and BPMN conversation diagram.

10.2.3.13 Participant

A participant represents a Partner entity and/or a Partner role that participates in a collaboration. Participants may be modeled in a BPMN collaboration diagram or a BPMN conversation diagram.

The assignment of a Partner entity and/or a Partner role to a participant is transferred to the BPMN allocation diagram (BPMN 2.0) assigned to the participant.

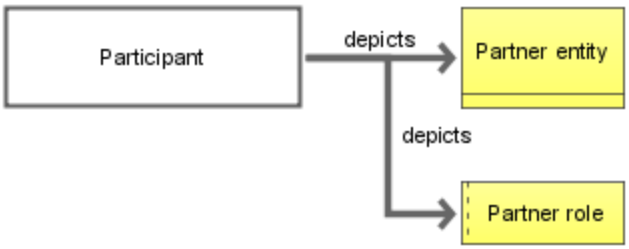


Illustration 158: BPMN allocation diagram (BPMN 2.0): Participant and partner entity/partner role

The usage of participants is described in the context of the BPMN collaboration diagram (see chapter Pool and participant (page 272)) and the BPMN conversation diagram (see chapter Participant (page 276)).

Participant, Partner entity and Partner role inherit from base element

See: Business Process Model and Notation (BPMN), version 2.0.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Participant	inherits from BaseElement	Object type: Participant (OT_BPMN_POOL) Symbol: Pool (ST_BPMN_POOL_1)
	name: string [0..1]	Attribute type **Name** (AT_NAME) of object type Participant (OT_BPMN_POOL)

Class	BPMN attribute name	Implementation in ARIS
	processRef: Process [0..1]	BPMN process diagram (BPMN 2.0) assigned to the participant (OT_BPMN_POOL Process displayed within in the pool
	partnerRoleRef: PartnerRole [0..1]	Model type: BPMN allocation diagram (BPMN 2.0): Object type: Role (OT_PERS_TYPE) Symbol: Partner role (ST_BPMN_PARTNER_ROLE) Connection type: depicts (CT_DEPICTS_1) Role
	partnerEntityRef: PartnerEntity [0..1]	Model type: BPMN allocation diagram (BPMN 2.0): Object type: Organizational unit (OT_ORG_UNIT) Symbol: Partner entity (ST_BPMN_PARTNER_ENTITY) Connection type: depicts (CT_DEPICTS_1) organizational unit
	interfaceRef: Interface [0..*]	Currently not implemented.
	participantMultiplicity: participantMultiplicity [0..1]	Attribute type in the attribute type group **BPMN 2.0 attributes/Participant multiplicity attributes** of the object type **Participant** (OT_BPMN_POOL): * Multi-instance participant (AT_BPMN_MI_PARTICIPANT) The mini-symbol (three vertical lines) is displayed by the program if the value of the attribute type **Multi-instance participant** is set to **true**.
	endpointRefs: EndPoint [0..*]	Currently not implemented.

Class	BPMN attribute name	Implementation in ARIS
Partner entity	inherits from BaseElement	Object type: Organizational unit (OT_ORG_UNIT) Symbol: Partner entity (ST_BPMN_PARTNER_ENTITY)
	name: string	Attribute type **Name** (AT_NAME) of object type **Organizational unit** (OT_ORG_UNIT)
Partner role	inherits from BaseElement	Object type: Role (OT_PERS_TYPE) Symbol: Partner role (ST_BPMN_PARTNER_ROLE)
	name: string	Attribute type **Name** of object type **Role** (OT_PERS_TYPE)
Participant Multiplicity	minimum: integer [0..1] = 2	Attribute type in the attribute type group **BPMN 2.0 attributes/Participant multiplicity attributes** of the object type **Participant** (OT_BPMN_POOL): * Minimum participant multiplicity (AT_BPMN_MINIMUM_MI_PARTICIPANT)
	maximum: integer [0..1] = 2	Attribute type in the attribute type group **BPMN 2.0 attributes/Participant multiplicity attributes** of the object type **Participant** (OT_BPMN_POOL): * Maximum participant multiplicity (AT_BPMN_MAXIMUM_MI_PARTICIPANT)
Participant Association	inherits from BaseElement	
	innerParticipantRef: Participant	Occurrence copy of the relevant participant.
	outerParticipantRef: Participant	Occurrence copy of the relevant participant.

10.2.3.14 Resource

Resources can be human resources as well as any other resource assigned to activities during process execution. A direct mapping of the BPMN resources to ARIS constructs is not possible - due to the semantically different object types representing resources in ARIS. ARIS does not only provide different object types, but also different connection types.

BPMN 2.0 only knows one object type called **Resource**. The BPMN ActivityResource and its specialized sub-classes correspond to ARIS connection types in combination with object types. Therefore, resources are not included in the current version of the BPMN 2.0 implementation.

See: Business Process Model and Notation (BPMN), version 2.0.

10.2.3.15 Sequence flow

The BPMN Sequence flow is mapped to nine different ARIS connection types, which are used to depict the control flow in traditional ARIS process models.

See: Business Process Model and Notation (BPMN), version 2.0.

Source object type	Connection type	Target object type
Event	occurs before	Event
Event	activates	Function
Event	is evaluated by	Rule
Function	creates	Event
Function	is predecessor of	Function
Function	leads to	Rule
Rule	leads to	Event
Rule	activates	Function
Rule	links	Rule

BPMN distinguishes three types of sequence flow:

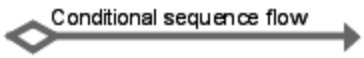
- **Unconditional sequence flow**

The unconditional sequence flow means the **normal** flow, no specific conditions apply. In other words: its condition has always the value **true**. It is depicted by a solid line with a solid arrowhead.

- **Conditional sequence flow**

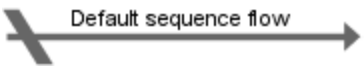
  The conditional sequence flow from an activity is drawn with a little diamond at the beginning of the connector, signifying a data condition. A conditional sequence flow from a gateway shares the same shape as a normal sequence flow.

  Conditional sequence flow from an activity:



- **Default sequence flow**

  The default sequence flow, denoted by a slash marker at the beginning of the connector means **otherwise**, that is, it is enabled if no other sequence flow condition evaluates to **true**.



All connection types used in BPMN diagrams must hold attributes for recording text annotations (page 201). Connection types emerging from activities and gateways need additional attributes for recording sequence flow conditions.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Sequence flow	inherits from FlowElement	The sequence flow is depicted by nine different connection types in the model types BPMN process diagram (BPMN 2.0) (MT_BPMN_PROCESS_DIAGRAM) and BPMN collaboration diagram (BPMN 2.0) (MT_BPMN_COLLABORATION_DIAGRAM): * event occurs before (CT_SUCCEED) event * event activates (CT_ACTIV_1) function * event is evaluated (CT_IS_EVAL_BY_1) by rule * function creates (CT_CRT_1) event * function is predecessor of (CT_IS_PREDEC_OF_1) function * function leads (CT_LEADS_TO_1) to rule * rule leads to (CT_LEADS_TO_2) event * rule activates (CT_ACTIV_1) function * rule links (CT_LNK_2) rule
	name: string	Attribute type Connection role of connection type Message flow (CT_BPMN_MESSAGE_FLOW)
	sourceRef: FlowNode	Source object of a sequence flow connection. Object types are: * Function * Event * Rule
	targetRef: FlowNode	Target object of a sequence flow connection. Object types are: * Function * Event * Rule

Class	BPMN attribute name	Implementation in ARIS
	conditionExpression : Expression [0..1]	Attribute type Condition expression (AT_BPMN_CONDITION_EXPRESSION) in attribute type group **BPMN 2.0 attributes** of the following connection types: * activates (CT_ACTIV_1) * creates (CT_CRT_1) * links (CT_LNK_2) * leads to (CT_LEADS_TO_1) * leads to (CT_LEADS_TO_2) * is predecessor of (CT_IS_PREDEC_OF_1) The value of the attribute type **Sequence flow condition** in the attribute type group **BPMN 2.0 attributes** must be set to **Conditional sequence flow**.
	isImmediate: boolean	Currently not implemented.
Flow node	incoming: Sequence Flow [0..*]	Incoming connections representing the sequence flow of the flow node object (object types: function, event, rule)
	outgoing: Sequence Flow [0..*]	Outgoing connections representing the sequence flow of the flow node object (object types: function, event, rule)

10.2.3.16 Elements not included in the current implementation

The following elements belong to the execution design level and are not included in the current version of the BPMN 2.0 implementation.

- Correlations (See: Business Process Modeling Notation (BPMN), version 2.0, page 136 and the following.)
- Conversation Associations (See: Business Process Modeling Notation (BPMN), version 2.0, page 135 and the following.)
- Error (See: Business Process Modeling Notation (BPMN), version 2.0, page 81 and the following.)
- Interaction node (See: Business Process Modeling Notation (BPMN), version 2.0, page 123.)
- Item definition (See: Business Process Modeling Notation (BPMN), version 2.0, page 91 and the following.)
- Services (See: Business Process Modeling Notation (BPMN), version 2.0, page 104 and the following.)

10.3 BPMN diagrams and ARIS model types: An overview

According to the BPMN 2.0 specification three diagram types are required for process modeling conformance: Process diagram, Collaboration diagram and Conversation diagram (See: Business Process Modeling Notation (BPMN), version 2.0. Page 2).

A careful consideration of these BPMN diagrams shows that the modeling constructs of the process diagram are a subset of the modeling constructs used in the collaboration diagram. There are also overlapping constructs in the collaboration and conversation diagram.

The model types listed in the following table are available in ARIS.

The BPMN allocation diagram allows the mapping of BPMN attributes and associations to the semantically richer Méthode ARIS where graphical elements are often used to represent BPMN attributes and associations.

BPMN diagram	ARIS model type
Process diagram	BPMN process diagram (BPMN 2.0)
Process diagram	Enterprise BPMN process diagram
Collaboration diagram	BPMN collaboration diagram (BPMN 2.0)
Collaboration diagram	Enterprise BPMN collaboration diagram
Conversation diagram	BPMN conversation diagram (BPMN 2.0)
	BPMN allocation diagram (BPMN 2.0)

In models of types **Enterprise BPMN process diagram** and **Enterprise BPMN collaboration diagram**, the following object types of the Méthode ARIS are available as lane symbols in addition to the BPMN 2.0 specification:

- Application system type
- Organizational unit
- Position
- Role
- Group

In models of type **Enterprise BPMN process diagram** and **Enterprise BPMN collaboration diagram**, the following additional connections to task objects are available in addition to the BPMN 2.0 specification:

- Application system type **supports** Task
- Organizational unit **supports** Task
- Position **carries out** Task
- Role **carries out** Task

- Group **carries out** Task

10.4 Process

See: Business Process Model and Notation (BPMN), version 2.0.

The BPMN process diagram depicts a BPMN process. A process is a specialization of a FlowElementsContainer. So, it contains the following elements:

- flow nodes (event, activity, and gateway)
- sequence flow
- artifacts (see chapter Artifacts (page 196))

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Process	inherits from CallableElement inherits from FlowElementsContainer	Model type: BPMN process diagram (BPMN 2.0)
	processType: ProcessType = none { none | executable | non-executable | public }	Attribute type in the attribute type group **BPMN 2.0 attributes** of model type BPMN process diagram (BPMN 2.0): * Process type (AT_BPMN_PROCESS_TYPE) Attribute values: * Undefined (= none), * Executable process (AVT_BPMN_EXECUTABLE), * Non-executable process (AVT_BPMN_NON_EXECUTABLE) * Public process (AVT_BPMN_PUBLIC)
	auditing: Auditing [0..1]	Currently not implemented.
	monitoring: Monitoring [0..1]	Currently not implemented.
	laneSets: LaneSet [0..*]	Object type **Lane** (OT_BPMN_LANE) Symbol: Lane (ST_BPMN_LANE_1)

Class	BPMN attribute name	Implementation in ARIS
	IsClosed: boolean = false	Attribute type **Is closed** (AT_BPMN_IS_CLOSED) in attribute type group **BPMN 2.0 attributes** of the BPMN process diagram
	supports: Process [0..*]	Currently not implemented.
	properties: Property [0..*]	Currently not implemented.
	definitionalCollaborationRef: Collaboration [0..1]	The BPMN collaboration diagram (BPMN 2.0) that contains the process

A process is a particular construct: On the one hand it is a model. On the other hand a process can be visualized within a pool in a collaboration. But a pool is not identical with a process, and vice versa. A pool represents a participant in a collaboration (see chapter Collaboration (page 271)). A pool may contain the process the participant uses in a specific collaboration.

The core elements for modeling a BPMN process are those constructs which can be connected to each other by sequence flow. They are called flow nodes. The corresponding ARIS object types and their symbols provided in the **Symbols** bar are listed in the table below.

BPMN element	ARIS object type	ARIS symbol	API name
Event	Event (OT_EVT)	Start event Intermediate event End event	ST_BPMN_START_EVENT ST_BPMN_INTERMEDIATE_EVENT ST_BPMN_END_EVENT
Activity	Function (OT_FUNC)	Task Subprocess Call activity	ST_BPMN_TASK ST_BPMN_SUBPROCESS ST_BPMN_CALL_ACTIVITY
Gateway	Rule (OT_RULE)	Gateway	ST_BPMN_RULE_1

These constructs are described in detail in the separate chapters below.

10.4.1 Activities

The BPMN activity is represented by the ARIS object type **Function**.

BPMN 2.0 differentiates three basic types of activities: task (atomic activity), subprocess (non-atomic activity) and call activity. The symbols depicting these activity types are provided in the ARIS **Symbols** bar.

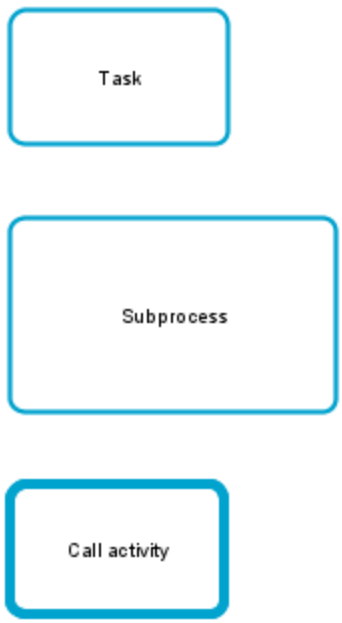


Illustration 159: Symbols representing activities in the Symbols bar

When the modeler places an activity symbol, the software sets the corresponding value of the ARIS attribute type **Activity type** (AT_BPMN_ACTIVITY_TYPE). This activity type controls the correct behavior of the symbol. For example: A subprocess may have **embedded** flow elements, a task must not; a call activity may reference another task or process, tasks and subprocesses must not.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Activity	inherits from FlowElement	Object type: Function (OT_FUNC) Attribute type **Activity type** (AT_BPMN_ACTIVITY_TYPE) in the attribute type group **BPMN 2.0 attributes** of object type **Function** Attribute values: * Task (AVT_BPMN_TASK) * Subprocess (AVT_BPMN_SUBPROCESS) * Call activity (AVT_BPMN_CALL_ACTIVITY)
	Compensation activity:: boolean = false	Attribute type **Compensation activity**: (AT_BPMN_COMPENSATION_ACTIVITY.TRM=Compensation activity) in the attribute type group **BPMN 2.0 attributes** of object type **Function**
	loopCharacteristics: LoopCharacteristics [0..1]	see below: Loop characteristics
	resources: ActivityResource [0..*]	Currently not implemented.
	default: SequenceFlow [0..1]	Attribute type **Sequence flow condition** (AT_BPMN_SEQ_FLOW_CONDITION) in the attribute type group **BPMN 2.0 attributes** of the following conenction types: * Activity creates ..., * Activity is predecessor of ..., * Activity leads to ... The attribute value must be set to **Default sequence flow**.
	ioSpecification: InputOutputSpecification [0..1]	Currently not implemented.
	properties: Property [0..*]	Currently not implemented.
	boundaryEventRefs: BoundaryEvent [0..*]	Connection type: Function can trigger event CT_BPMN_CAN_TRIGGER

Class	BPMN attribute name	Implementation in ARIS
	dataInputAssociations: DataInputAssociation [0..*]	Currently not implemented.
	dataOutputAssociations: DataOutputAssociation [0..*]	Currently not implemented.
	startQuantity: integer = 1	Currently not implemented.
Class	**completionQuantity**: integer = 1	Currently not implemented.

10.4.1.1 Resource assignment

Resource assignments are not included in the current version of the BPMN 2.0 implementation. They will be dealt with in detail when implementing the execution design level in ARIS.

See: Business Process Model and Notation (BPMN), version 2.0.

10.4.1.2 Performer

Resource assignments are not included in the current version of the BPMN 2.0 implementation. They will be dealt with in detail when implementing the execution design level in ARIS.

See: Business Process Model and Notation (BPMN), version 2.0.

10.4.1.3 Activity type: Task

See: Business Process Model and Notation (BPMN).

BPMN 2.0 distinguishes eight task types which are represented by different symbols (see). Only the Abstract task is available in the **Symbols** bar. The symbols of the remaining seven special task types are not available in the **Symbols** bar, they are handled by the program.

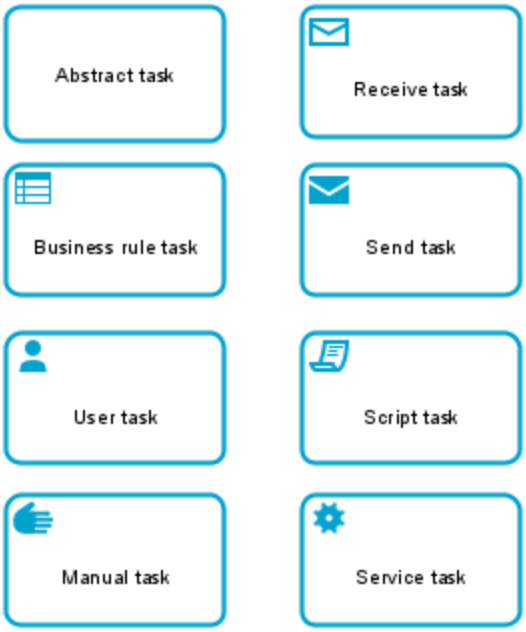


Illustration 160: Task symbols

When the modeler selects a specific task symbol the software sets the corresponding value of the ARIS attribute type **Task type**. This attribute type is read-only. It provides the following values: Abstract task, Business rule task, Manual task, Script task, Send task, Service task, Receive task, and User task.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Task	inherits from Activity	The value of the attribute type **Activity type** (AT_BPMN_ACTIVITY_TYPE) is set to **Task** in the attribute type group **BPMN 2.0 attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbol: Task (ST_BPMN_TASK) or a special task symbol (see below)

Class	BPMN attribute name	Implementation in ARIS
Service task	inherits from Activity	The value of the attribute type **Task type** (AT_BPMN_TASK_TYPE) is set to **Service task** in the attribute type group **BPMN 2.0 attributes/Task attributes** of object type **Function** Object type: Function (OT_FUNC) Symbol: Service task (ST_BPMN_SERVICE_TASK)
	implementation: Implementation = Web Service {Web Service | Other | Unspecified}	Currently not implemented.
	operationRef: Operation [0..1]	Currently not implemented.
Send task	inherits from Activity	The value of the attribute type **Task type** (AT_BPMN_TASK_TYPE) is set to **Send task** in the attribute type group **BPMN 2.0 attributes/Task attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbol: Send task (ST_SEND_TASK)
	messageRef: Message [0..1]	Connection type in the BPMN collaboration diagram (BPMN 2.0) * Function sends message.
	operationRef: Operation [0..1]	Currently not implemented.
	implementation: Implementation = Web Service {Web Service| Other | Unspecified}	Currently not implemented.

Class	BPMN attribute name	Implementation in ARIS
Receive task	inherits from Activity	The value of the attribute type **Task type** (AT_BPMN_TASK_TYPE) is s**et to Receive task in the attribute type group BPMN 2.0 attributes/Task attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbol: Receive task (ST_RECEIVE_TASK)
	messageRef: Message [0..1	Connection type in the BPMN collaboration diagram (BPMN 2.0) * Message is received from function
	Instantiate: boolean = False	Currently not implemented.
	operationRef: Operation [0..1]	Currently not implemented.
	implementation: Implementation = Web Service {Web Service | Other | Unspecified}	Currently not implemented.
User task	inherits from Activity	The value of the attribute type **Task type** (AT_BPMN_TASK_TYPE) is set to **User task** in the attribute type group **BPMN 2.0 attributes/Task attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbol: User task (ST_USER_TASK)
	Implementation: UserTaskImplementation = Other {HumanTaskWebService | WebService | Other | Unspecified}	Currently not implemented.
	renderings: Rendering [0..*]	Currently not implemented.

Class	BPMN attribute name	Implementation in ARIS
Manual task	inherits from Activity	The value of the attribute type **Task type** (AT_BPMN_TASK_TYPE) is set to **Manual task** in the attribute type group **BPMN 2.0 attributes/Task attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbol: Manual task (ST_MANUAL_TASK)
Business Rule Task	inherits from Activity	The value of the attribute type **Task type** (AT_BPMN_TASK_TYPE) is set to **Business rule task** in the attribute type group **BPMN 2.0 attributes/Task attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbol: Business rule task (ST_BUSINESS_RULE_TASK)
	Implementation: BusinesRuleTaskImplementation = Other {BusinessRuleWebService | WebService | Other | Unspecified}	Currently not implemented.
Script task	inherits from Activity	The value of the attribute type **Task type** (AT_BPMN_TASK_TYPE) is set to **Script task** in the attribute type group **BPMN 2.0 attributes/Task attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbol: Script task (ST_SCRIPT_TASK)
	scriptLanguage: string [0..1]	Currently not implemented.
	script: string [0..1]	Currently not implemented.

10.4.1.4 Human interactions

User tasks and manual tasks are relevant for modeling human interactions. Their attributes and model associations can also be found in chapter Activity type: Task (page 224). They will be dealt with in detail when implementing the execution design level in ARIS.

See: Business Process Model and Notation (BPMN), version 2.0.

10.4.1.5 Activity type: Subprocess

See: Business Process Model and Notation (BPMN), version 2.0.

BPMN 2.0 knows four types of subprocesses:

- Subprocess (standard)
  (A ([standard] subprocess corresponds to the embedded subprocess in BPMN 1.x.)
- Event subprocess
- Transaction, and
- Ad hoc subprocess

Each type of a subprocess can be displayed as

- Subprocess (collapsed) or
- Subprocess (expanded)

Collapsed subprocesses have a special marker displayed at the bottom of the corresponding subprocess symbol:

10.4.1.6 Subprocess type: Subprocess

A standard subprocess shares the same shape as a task. In the collapsed form, the subprocess object uses the **+**-marker to distinguish it from a task. Expanded subprocesses have no marker, they reveal their **embedded** objects.

The symbol representing the expanded subprocess is available in the **Symbols** bar, the symbol representing the collapsed subprocess is handled by the software.

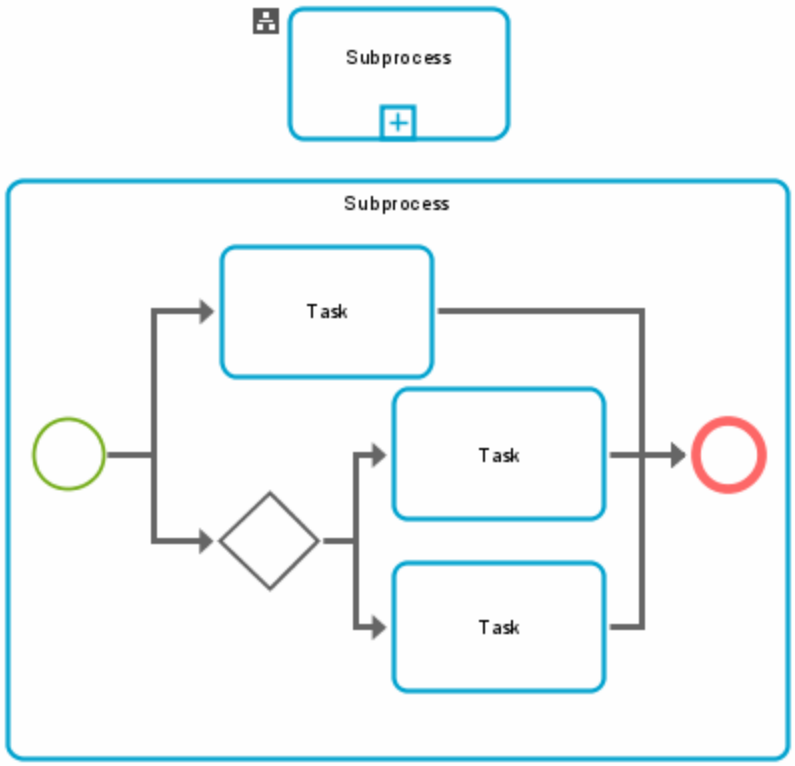


Illustration 161: Symbols of a standard subprocess

The attributes and model associations of a subprocess and their mapping to ARIS constructs are listed in the table below.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Subprocess	inherits from Activity inherits from FlowElementsContainer	The value of the attribute type **Activity type** (AT_BPMN_ACTIVITY_TYPE) is set to **Subprocess** in the attribute type group **BPMN 2.0 attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbols: * Subprocess (ST_BPMN_SUB_PROCESS) * Subprocess collapsed (ST_BPMN_SUB_PROCESS_COLLAPSED) * or a special subprocess symbol (see below)
	triggeredByEvent: boolean = false	Attribute type **Event subprocess** (AT_BPMN_EVENT_SUB_PROCESS) in the attribute type group **BPMN 2.0 attributes/Subprocess attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbols: * Event subprocess (ST_BPMN_EVENT_SUBPROCESS) * Event subprocess (collapsed) (ST_BPMN_EVENT_SUBPROCESS_COLLAPSED) The symbols are rendered by the program.

10.4.1.7 Type de sous-processus : sous-processus d'événement

Un sous-processus d'événement est un sous-processus spécialisé utilisé dans un processus ou un sous-processus. Contrairement à un sous-processus par défaut qui utilise le flux de son processus parent en tant que déclencheur, un sous-processus d'événement ne fait pas partie du flux normal de son processus parent. Il n'y a pas de flux de séquence entrant ni sortant. Un sous-processus d'événement possède un événement de départ déclencheur. Le sous-processus d'événement est lancé à chaque fois que l'événement de départ est déclenché alors que le processus parent est activé.

Les symboles d'un sous-processus d'événement sont représentés ci-dessous. Si le sous-processus d'événement est réduit, son événement de départ est utilisé comme marque dans le coin supérieur gauche du symbole. Le programme reproduit cette marque.

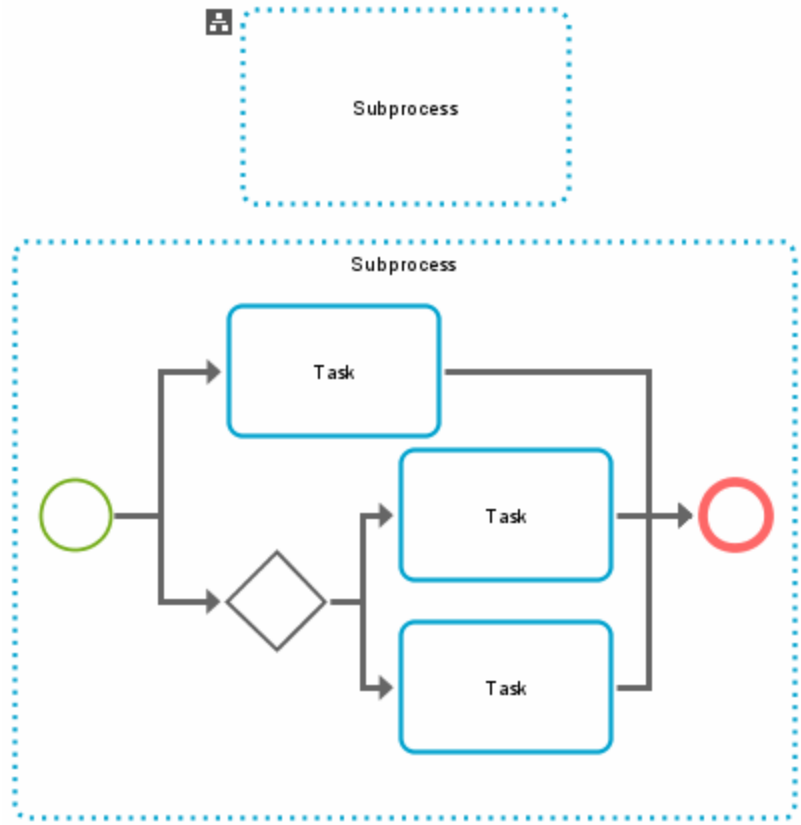


Illustration 162: Symboles d'un sous-processus d'événement

Il existe un type d'attribut booléen ARIS **Sous-processus d'événement** représentant l'attribut BPMN **triggeredByEvent** (cf. table dans Type de sous-processus : sous-processus (page 230)). Le logiciel utilise ce type d'attribut protégé en écriture mais visible.

10.4.1.8 Subprocess type: Transaction

A transaction subprocess, denoted with a double-lined boundary, is a specialized type of subprocess. In a transaction subprocess all activities must either complete successfully or the subprocess must be rolled back to its original consistent state. A transaction subprocess has a special behavior: It is associated with a transaction protocol that has to verify that all activities have been successfully completed. The symbols are not available in the **Symbols** bar, they are handled by the software. The program also sets the value of the ARIS attribute type **Subprocess type** to **Transaction**.

Illustration 163: Symbol for a collapsed transaction

A transaction inherits from **Activity**. The attributes and model associations of a transaction subprocess and their mapping to ARIS constructs are shown in the table below.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Transaction	inherits from Activity	The value of the attribute type **Subprocess type** (AT_BPMN_SUBPROCESS_TYPE) is set to **Transaction** in the attribute type group **BPMN 2.0 attributes/Subprocess attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbols: * Transaction (ST_BPMN_TRANSACTION) * Transaction (collapsed) (ST_BPMN_TRANSACTION_COLLAPSED_1) The symbols are rendered by the software.
	protocol: string [0..1]	Currently not implemented.
	method: TransactionMethod = compensate { compensate | store | image }	Currently not implemented.

10.4.1.9 Subprocess type: Ad hoc subprocess

A transaction subprocess, denoted with a double-lined boundary, is a specialized type of subprocess. In a transaction subprocess all activities must either complete successfully or the subprocess must An Ad hoc subprocess, denoted with a tilde marker, is a specialized type of subprocess. It contains a set of activities that could be performed. Sequence flow between activities is optional in an Ad hoc subprocess. What activities are performed as well as the sequence and the number of performances is determined by the performers of the activities. During execution of the (parent) process, any one or more of the activities may be active.

The Méthode ARIS provides the tilde marker as mini-symbol for the value **Ad hoc subprocess** of the attribute **Subprocess type**. The program will render the symbols for the Ad hoc subprocess.

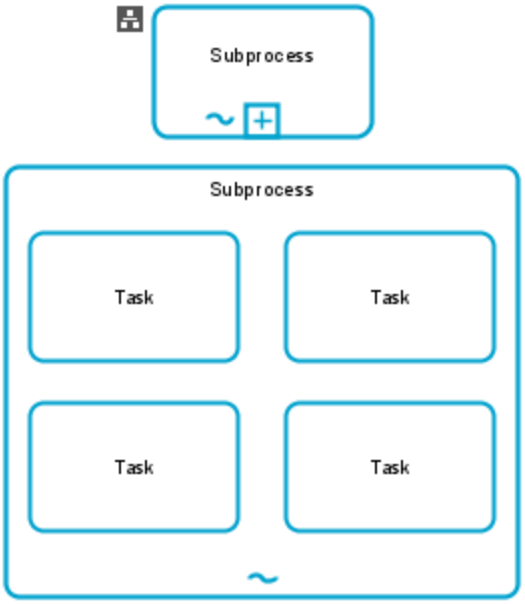


Illustration 164: Symbol for a collapsed and expanded Ad hoc subprocess

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Ad hoc subprocess	inherits from Activity	The value of the attribute type **Subprocess type** (AT_BPMN_SUB_PROCESS_TYPE) is set to **Ad hoc subprocess** in the attribute type group **BPMN 2.0 attributes/Subprocess attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbols: The mini-symbol **tilde** is rendered by the program.
	completionCondition: Expression	Attribute type **Ad hoc completion condition** (AT_BPMN_COMPLETION_CONDI) in the attribute type group **BPMN 2.0 attributes/Subprocess attributes/Ad hoc subprocess attributes** of object type **Function**.
	ordering: AdHocOrdering = parallel { parallel | sequential }	Currently not implemented.
	cancelRemainingInstances : Boolean = True	Currently not implemented.

10.4.1.10 Subprocess type: Call Activity

A call activity represents the invocation of either a reusable global task or a process. The call activity represents the calling element, and the global task or process represents the called element.

The symbol **Call activity** is available in the **Symbols** bar. If the modeler places this symbol, the value of the attribute **Activity type** is set to **Call activity** and the software provides a dialog where the modeler selects the task or the process being called. Depending on this selection, the value of the attribute type **Called element** is set to **Global task** or **Global process**. The program renders the symbol for the call activity. It corresponds to the symbol for the called task or process, but it is drawn with a thick border.

If a task is selected the program automatically creates a connection (call activity invokes task) on definition level. If a process is selected, the related process diagram is assigned to the call activity.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
CallActivity	inherits from Activity	The value of the attribute type **Activity type** (AT_BPMN_ACTIVITY_TYPE) is set to **Call activity** in the attribute type group **BPMN 2.0 attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbol: The symbol depends on the activity being called. The program will render the symbol.

Class	BPMN attribute name	Implementation in ARIS
	calledElement: CallableElement [0..1]	**For tasks**: The value of the attribute type **Called element** (AT_BPMN_CALLED_ELEMENT) is set to **Global task**. The program creates the connection type **Function invokes [CT_INVOKES] function** on definition level. **For processes**: The value of the attribute type **Called element** (AT_BPMN_CALLED_ELEMENT) is set to **Global process**. The BPMN process diagram of the called process is assigned to the Call activity. In both cases the software provides an appropriate dialog.

10.4.1.11 Global task

The global task is described in chapter Callable Elements (page 202).

A global task has no specific attributes and model associations, it inherits from callable elements.

10.4.1.12 Loop characteristics

BPMN 2.0 provides two alternatives to model repeating activities (both tasks and subprocesses):

- Loop activity (= standard loop)
- Multi-instance activity

10.4.1.13 Loop characteristics representations

An activity can be specified to repeat based on a condition. That is called standard loop activity in BPMN. A standard loop is equivalent to the **do while** and **do until** structure in programming. The number of iterations is unknown.

A multi-instance activity is another type of repeating activity useful for performing actions on a list of items. A multi-instance activity is equivalent with a **for each** structure in programming. The number of iterations is known when the activity starts. It is the number of items in the list. Iterations of a multi-instance activity can be performed concurrently or sequentially.

The marker for a standard loop is a circular arrow at the bottom center of the activity symbol.

Illustration 165: Symbols of Standard loop activities

The markers for multi-instance activities are three bars at the bottom center of the task or subprocess symbol.

Vertical bars are used to represent concurrent/parallel performances:

Illustration 166: Symbols of BPMN multi-instance (parallel) activities

Horizontal bars are used to represent sequential performances:

Illustration 167: Symbols for activities of the BPMN multi-instance (parallel)

Loop characteristics has no specific attributes, it inherits the attributes and associations of base element. The attribute type **Loop type** is used in ARIS to specify whether the loop is a standard loop, a multi-instance parallel loop, or a multi-instance sequential loop. The attribute values are visualized by mini-symbols.

10.4.1.14 Standard and multi-instance loop characteristics and complex behavior definition

The attributes and model associations of standard activities, multi-instance loop activities, and complex behavior definition are summarized in the table below.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
LoopCharacteristics	inherits from BaseElement	Attribute type group **Loop characteristics** (AT_BPMN_LOOP_CHARACTERISTICS) in attribute type group **BPMN 2.0 attributes** of object type **Function** (OT_FUNC).

Class	BPMN attribute name	Implementation in ARIS
StandardLoopCharacteristics	inherits from BaseElement	The value of the attribute type **Loop type** (AT_BPMN_LOOP_TYPE_2) is set to **Standard loop** (AVT_BPMN_STANDARD_LOOP) in the attribute type group **BPMN 2.0 attributes/Loop characteristics** of object type **Function**.
	testBefore: boolean = False	Attribute type **Test before** (AT_BPMN_LOOP_TEST_TIME) in the attribute type group **BPMN 2.0 attributes/Loop characteristics/Standard loop attributes** of object type **Function**.
	loopMaximum: Expression [0..1]	Attribute type **Loop maximum** (AT_BPMN_MAX_LOOP) in the attribute type group **BPMN 2.0 attributes/Loop characteristics/Standard loop attributes** of object type **Function**.
	loopCondition: Expression [0..1]	Attribute type **Loop condition** (AT_BPMN_LOOP_CONDITION) in the attribute type group **BPMN 2.0 attributes/Loop characteristics/Standard loop attributes** of object type **Function**.
MultiInstanceLoop Characteristics	inherits from BaseElement	The value of the attribute type **Loop type** (AT_BPMN_LOOP_TYPE_2) is set to **Multi-instance sequential loop** (AVT_BPMN_MULTI_INSTANCE_SEQUENTIAL_LOOP) or **Multi-instance parallel loop** (AVT_BPMN_MULTI_INSTANCE_PARALLEL_LOOP) in the attribute type group **BPMN 2.0 attributes/Loop characteristics** of object type **Function**.
	isSequential: boolean = False	isSequential = true corresponds to: Loop type = Multi-instance sequential loop isSequential = false corresponds to: Loop type = Multi-instance parallel loop

Class	BPMN attribute name	Implementation in ARIS
	loopCardinality: Expression [0..1]	Attribute type **Loop cardinality** (AT_BPMN_LOOP_CARDINALITY) in the attribute type group **BPMN 2.0 attributes/Loop characteristics/Multi-instance loop attributes** of object type **Function**.
	loopDataInput: DataInput [0..1]	Currently not implemented.
	loopDataOutput: DataOutput [0..1]	Currently not implemented.
	inputDataItem: Property [0..1]	Currently not implemented.
	outputDataItem: Property [0..1]	Currently not implemented.
	completionCondition: Expression [0..1]	Currently not implemented.
	behavior: MultiInstanceBehavior = all { none | one | all | complex }	Currently not implemented.
	complexBehaviorDefinition: ComplexBehaviorDefinition [0..*]	Currently not implemented.
	oneBehaviorEventRef: EventDefinition [0..1]	Currently not implemented.
	noneBehaviorEventRef: EventDefinition [0..1]	Currently not implemented.

Class	BPMN attribute name	Implementation in ARIS
ComplexBehaviorDefinition	inherits from BaseElement	Currently not implemented.
	condition: Formal Expression	
	event: ImplicitThrowEvent	

10.4.2 Items and Data

As mentioned above, the current implementation of BPMN 2.0 in ARIS focuses on the business process level. Therefore, only data objects and data stores are provided - as input or output of activities. Detailed data modeling aspects (for example data structures, data states, data associations) are omitted.

See: Business Process Model and Notation (BPMN), version 2.0.

10.4.2.1 Data object

In BPMN 1.x data were considered as an artifact; in BPMN 2.0, data objects were upgraded to objects in the BPMN semantic model.

On the business level, where the data structures and mappings are not included, data objects are represented in ARIS by six symbols of the object type **Cluster/Data model**:

- Data object
- Data collection
- Data input
- Data input collection
- Data output
- Data output collection

When you place a Data object the object symbols **Data object**, **Data input**, and **Data output** are provided. You can select the object symbols **Data collection**, **Data input collection**, and **Data output collection** using the **Object appearance** page of the **Object properties** dialog.

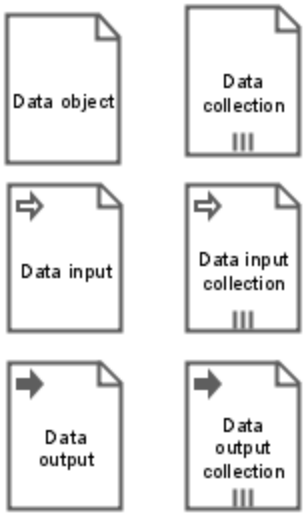


Illustration 168: Symbols of data objects

Only the **Data object** symbol is available in the **Symbols** bar.

Data objects can represent the input or output of activities by using the following connection types:

- Cluster/Data model **is input for** function
- Function **has as output** Cluster/Data model

The data input symbols must not be the target of a **has as output** connection, and the data output symbols must not be the source of an **is input for** connection. The software ensures this.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
ItemAware Element	inherits from BaseElement	
	itemSubjectRef: ItemDefinition [0..1]	Currently not implemented.
	dataState: DataState [0..1]	Currently not implemented.

Class	BPMN attribute name	Implementation in ARIS
Data object	inherits from FlowElement & ItemAwareElement	Object type: Cluster/Data model (OT_CLST) Six symbols: * Data object (ST_BPMN_DATA_OBJECT) * Data collection (ST_BPMN_DATA_COLLECTION) * Data input (ST_BPMN_DATA_INPUT) * Data input collection (ST_BPMN_DATA_INPUT_COLLECTION) * Data output (ST_BPMN_DATA_OUTPUT) * Data output collection (ST_BPMN_DATA_OUTPUT_COLLECTION)
	isCollection: Boolean = False	Represented by special symbols of the object type **Cluster/Data model** (OT_CLST)

10.4.2.2 Data store

Unlike data objects, which live only as long as the process instance is running, a Data store represents information that persists beyond the lifetime of a particular process. On the business level, a Data store is represented by the symbol **Data store** of the ARIS object type **Information carrier**. This symbol is available in the **Symbols** bar.

On the business level, where the data structures and mappings are not included, data objects are represented in ARIS by six symbols of the object type **Cluster/Data model**:

- Information carrier **provides input for** function

- Function **creates output to** information carrier.

Illustration 169: Symbol for a data store

A Data store inherits from FlowElement and ItemAwareElement.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
DataStore	inherits from FlowElement and ItemAwareElement	Object type: Information carrier (OT_INFO_CARR) Symbol: Data store (ST_BPMN_DATA_STORE)
	name: string	Attribute type **Name** (AT_NAME) of object type **Information carrier** (OT_INFO_CARR)
	capacity: Integer [0..1]	Currently not implemented.
	isUnlimited: Boolean = False	Currently not implemented.
DataStoreReference	inherits from FlowElement and ItemAwareElement	
	dataStoreRef: DataStore	Occurrence copies of the (referenced) data store.

10.4.3 Events

BPMN events are represented in ARIS by the object type **Event**. Altogether there are sixty-three symbols available in BPMN 2.0. The main event types are:

- Start event
- Intermediate event
- End event

Only these three events are provided in the **Symbols** bar in ARIS (see type = None). The remaining symbols are provided as symbols in the Méthode ARIS.

BPMN events: (See: Business Process Modeling Notation (BPMN), version 2.0, page 233 and the following).

Types	Start			Intermediate				End
	Top level	Event Sub-process Inter-rupting	Event Sub-process Non-Inter-rupting	Catching	Boundary Inter-rupting	Boundary Non-Inter-rupting	Throwing	
None								
Message								
Timer								
Error								
Escalation								
Cancel								
Compensation								
Conditional								

Types	Start			Intermediate				End
Link								
Signal								
Terminate								
Multiple								
Parallel multiple								

10.4.3.1 Catch events and throw events

Events can be:

- catch events (all start and a number of intermediate events)
- throw events (all end events a number of intermediate)

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Event		Object type: Event (OT_EVT) Symbols: sixty-three different symbols (see below)
Catch Event	inherits from FlowElement	Object type: Event (OT_EVT) Symbols: different start or intermediate event symbols
	eventDefinitionRefs: EventDefinition [0..*]	Occurrence copy of the corresponding throw event.
	eventDefinitions: EventDefinition [0..*]	Attribute type **Event definition** (AT_BPMN_EVENT_DEFINITION) in the attribute type group **BPMN 2.0 attributes** of object type **Event** (OT_EVT). The values of this attribute type are: None, Message, Timer, Error, Escalation, Cancel, Compensation, Conditional, Link, Signal, Multiple, Parallel multiple (as special case of Multiple). Each event definition has a specific marker inside the event symbol.
	dataOutputAssociations: DataOutputAssociation [0..*]	Currently not implemented.
	dataOutput: dataOutput [0..*]	Connection type in the BPMN process diagram (BPMN 2.0) and the BPMN collaboration diagram (BPMN 2.0): Event (symbol: only catch events) has as output Cluster/data model

Class	BPMN attribute name	Implementation in ARIS
	outputSet: OutputSet [0..1]	Currently not implemented.
Throw event	inherits from FlowElement	Object type: Event (OT_EVT) Symbols: different intermediate or end event symbols
	eventDefinitionRefs: EventDefinition [0..*]	Occurrence copy of the corresponding catch event.
	eventDefinitions: EventDefinition [0..*]	Attribute type **Event definition** (AT_BPMN_EVENT_DEFINITION) in the attribute type group **BPMN 2.0 attributes** of object type **Event** (OT_EVT). The values of this attribute type are: None, Message, Error, Escalation, Cancel, Compensation, Link, Signal, Terminate, Multiple. Each event definition has a specific marker inside the event symbol.
	dataInputAssociations: DataInputAssociation [0..*]	Currently not implemented.
	dataInput: DataInput [0..*]	Connection type in the BPMN process diagram (BPMN 2.0) and the BPMN collaboration diagram (BPMN 2.0): Cluster/data model is input for event (symbol: only throw events)
	inputSet: InputSet [0..1]	Currently not implemented.
Implicit Throw Event	inherits from ThrowEvent	Currently not implemented.

10.4.3.2 Start event

The symbols of start events are depicted in chapter events (page 248). Only the start event **None** is available in the **Symbols** bar. When placing this start event, the modeler is guided by a special functionality of the program.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Start event	inherits from CatchEvent	Object type: Event (OT_EVT) Symbol: Start event (ST_BPMN_START_EVENT)
	isInterrupting: boolean	Interrupting start events are represented by specific event symbols.

10.4.3.3 Intermediate events

The symbols of intermediate events are depicted in chapter events (page 248). Only the intermediate event **None** is available in the **Symbols** bar. When placing this start event, the modeler is guided by a special functionality of the program. Intermediate events have no specific attributes and associations.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
IntermediateEvent		Object type: Event (OT_EVT) Symbol: Intermediate event (ST_BPMN_INTERMEDIATE_EVENT)

Some types of intermediate events can be attached to the boundary of activities, they are called **boundary events** (see column **Boundary Interrupting** and **Boundary Non-interrupting** in chapter events (page 248).

Boundary events are always catch events. Their attributes and model associations are shown in the table below.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Boundary events	inherits from CatchEvent	Boundary events are specific intermediate events.
	AttachedTo: Activity	Connection type in the BPMN process diagram (BPMN 2.0) and the BPMN collaboration diagram (BPMN 2.0): Function can trigger (CT_BPMN_CAN_TRIGGER) event (symbol: **intermediate event**)
	CancelActivity: boolean	(Non-)Interrupting events are represented by specific event symbols.

10.4.3.4 End event

The symbols of end events are depicted in chapter events (page 248). Only the none end event is available in the **Symbols** bar. When placing this start event, the modeler is guided by a special functionality of the program.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
EndEvent		Object type: Event (OT_EVT) Symbol: End event (ST_BPMN_END_EVENT)

10.4.3.5 Event definitions

BPMN 2.0 distinguishes the following event definitions: None, Message, Timer, error, Escalation, Cancel, Compensation, Conditional, Link, Signal, Terminate and Multiple (**Parallel multiple** is a special case **Multiple**). The different definitions are visualized by specific markers placed within the **None start**, **Intermediate** and **End event** symbol.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
EventDefinition	inherits from BaseElement	Attribute type **Event definition** (AT_BPMN_EVENT_DEFINITION) in the attribute type group **BPMN 2.0 attributes** of object type Event (OT_EVT) Attribute values: None, Message, Timer, Error, Escalation, Cancel, Compensation, Conditional, Link, Signal, Terminate, Multiple. This attribute is read-only and set automatically by the software.
CancelEventDefinition	inherits from BaseElement	Object type: Event (OT_EVT) Symbols: * Cancel intermediate event (ST_BPMN_CANCEL_INTERMEDIATE_EVENT) * Cancel end event (ST_BPMN_CANCEL_END_EVENT)

Class	BPMN attribute name	Implementation in ARIS
CompensationEvent Definition	inherits from BaseElement	Object type: Event (OT_EVT) Symbols: * Compensation start event (ST_BPMN_COMPENSATION_START * Compensation intermediate event (catch)(ST_BPMN_COMPENSATION_INTERMEDIATE_CATCH * Compensation intermediate event (throw) (ST_BPMN_COMPENSATION_INTERMEDIATE_THROW) * Compensation end event (ST_BPMN_COMPENSATION_END_EVENT)
	activityRef: Activity [0..1]	Attribute type **Wait for completion** (AT_BPMN_WAIT_FOR_COMPLETION) in the attribute type group **BPMN 2.0 attributes/Compensation event attributes** of object type **Event** (OT_EVT).
ConditionalEvent Definition	inherits from BaseElement	Object type: Event (OT_EVT) Symbols: * Conditional start event (ST_BPMN_RULE_START_EVENT) * Conditional start event (non-interrupting) (ST_BPMN_CONDITIONAL_START_NI) * Conditional intermediate event (ST_BPMN_RULE_INTERMEDIATE_EVENT) * Conditional intermediate event (non-interrupting) (ST_BPMN_CONDITIONAL_INTERMEDIATE_NI)

Class	BPMN attribute name	Implementation in ARIS
	condition: Expression	Attribute type **Condition** (AT_BPMN_RULE_EXPRESSION) in the attribute type group **BPMN 2.0 attributes/Conditional event attributes** of object type **Event** (OT_EVT).
ErrorEventDefinition	inherits from BaseElement	Object type: Event (OT_EVT) Symbols: * Error start event (ST_BPMN_ERROR_START) * Error intermediate event (ST_BPMN_ERROR_INTERMEDIATE_EVENT * Error end event (ST_BPMN_ERROR_END_EVENT)
	errorCode: string	Currently not implemented.
	error: Error [0..1]	Currently not implemented.

Class	BPMN attribute name	Implementation in ARIS
EscalationEvent Definition	inherits from BaseElement	Object type: Event (OT_EVT) Symbols: * Escalation start event (ST_BPMN_ESCALATION_START) * Escalation start event (non-interrupting) (ST_BPMN_ESCALATION_START_NI) * Escalation intermediate event (catch) (ST_BPMN_ESCALATION_INTERMEDIATE_CATCH) * Escalation intermediate event (non-interrupting) (ST_BPMN_ESCALATION_INTERMEDIATE_NI) * Escalation intermediate event_throw (ST_BPMN_ESCALATION_INTERMEDIATE_THROW) * Escalation end event (ST_BPMN_ESCALATION_END)
	escalationCode: string	Currently not implemented.
	escalationRef: Escalation [0..1]	Currently not implemented.
LinkEventDefinition	inherits from BaseElement	Object type: Event (OT_EVT) Symbols: * Link intermediate event (catch) (ST_BPMN_LINK_INTERMEDIATE_CATCH) * Link intermediate event (throw) (ST_BPMN_LINK_INTERMEDIATE_THROW) Catch and throw link events are referred to each other by occurrence copies.
	name: string	Attribute type **Name** (AT_NAME) of object type **Event** (OT_EVT)

Class	BPMN attribute name	Implementation in ARIS
MessageEvent Definition	inherits from BaseElement	Object type: Event (OT_EVT) Symbols: * Message start event (ST_BPMN_MESSAGE_START_EVENT) * Message start event (non-interrupting) (ST_BPMN_MESSAGE_START_NI) * Message intermediate event (catch) (ST_BPMN_MESSAGE_INTERMEDIATE_CATCH) * Message intermediate event (non-interrupting) (ST_BPMN_MESSAGE_INTERMEDIATE_NI) * Message intermediate event (throw) (ST_BPMN_MESSAGE_INTERMEDIATE_THROW) * Message end event (ST_BPMN_MESSAGE_END_EVENT)
	MessageRef: Message [0..1]	Currently not implemented.
	operationRef: Operation [0..1]	Currently not implemented.

Class	BPMN attribute name	Implementation in ARIS
Multiple event		Object type: Event (OT_EVT) Symbols: * Multiple start event (ST_BPMN_MULTIPLE_START_EVENT) * Multiple start event (non-interrupting) (ST_BPMN_MULTIPLE_START_NI) * Multiple intermediate event (catch) (ST_BPMN_MULTIPLE_INTERMEDIATE_CATCH) * Multiple intermediate event (non-interrupting)(ST_BPMN_MULTIPLE_INTERMEDIATE_NI) * Multiple intermediate event (throw) (ST_BPMN_MULTIPLE_INTERMEDIATE_THROW) * Multiple end event (ST_BPMN_MULTIPLE_END_EVENT)
None event		Object type: Event (OT_EVT) Symbols: * Start event (ST_BPMN_SE) * Intermediate event (ST_BPMN_IE) * End event (ST_BPMN_EE) These symbols are available in the Symbols bar.

Class	BPMN attribute name	Implementation in ARIS
Parallel multiple event		Object type: Event (OT_EVT) Symbols: * Parallel multiple start event (ST_BPMN_PARALLEL_MULTIPLE_START) * Parallel multiple start event (non-interrupting) (ST_BPMN_PARALLEL_MULTIPLE_START_NI) * Parallel multiple intermediate event (ST_BPMN_PARALLEL_MULTIPLE_INTERMEDIATE) * Parallel multiple intermediate event (non-interrupting) (ST_BPMN_PARALLEL_MULTIPLE_INTERMEDIATE_NI)
SignalEventDefinition	inherits from BaseElement	Object type: Event (OT_EVT) Symbols: * Signal start event (ST_BPMN_SIGNAL_START_EVENT) * Signal start event (non-interrupting) (ST_BPMN_SIGNAL_START_NI) * Signal intermediate event (catch) (ST_BPMN_SIGNAL_INTERMEDIATE_EVENT) * Signal intermediate event (non-interrupting) (ST_BPMN_SIGNAL_INTERMEDIATE_NI) * Signal intermediate event (throw) (ST_BPMN_SIGNAL_INTERMEDIATE_THROW) * Signal end event (ST_BPMN_SIGNAL_END_EVENT)
	signalRef: Signal	Currently not implemented.

Class	BPMN attribute name	Implementation in ARIS
TerminateEvent Definition	inherits from BaseElement	Object type: Event (OT_EVT) Symbol: * Terminate end event (ST_BPMN_TERMINATE_END_EVENT)
TimerEventDefinition	inherits from BaseElement	Object type: Event (OT_EVT) Symbols: * Timer start event (ST_BPMN_TIMER_START_EVENT) * Timer start event (non-interrupting) (ST_BPMN_TIMER_START_NI) * Timer intermediate event (ST_BPMN_TIMER_INTERMEDIATE_EVENT) * Timer intermediate event (non-interrupting) (ST_BPMN_TIMER_INTERMEDIATE_NI)
	timeDate: Expression [0..1]	Attribute type **Time date** (AT_BPMN_TIMEDATE) in the attribute type group **BPMN 2.0 attributes/Timer event attributes** of object type **Event** (OT_EVT).
	timeCycle: Expression [0..1]	Attribute type **Time cycle** (AT_BPMN_TIMECYCLE) in the attribute type group **BPMN 2.0 attributes/Timer event attributes** of object type **Event** (OT_EVT)

10.4.4 Gateways

The ARIS object type **Rule** depicts BPMN gateways. Although BPMN 2.0 knows five different gateway types, only one symbol is available in the **Symbols** bar:

The remaining gateway symbols are handled by the program. The following figure depicts all (basic) gateway symbols.

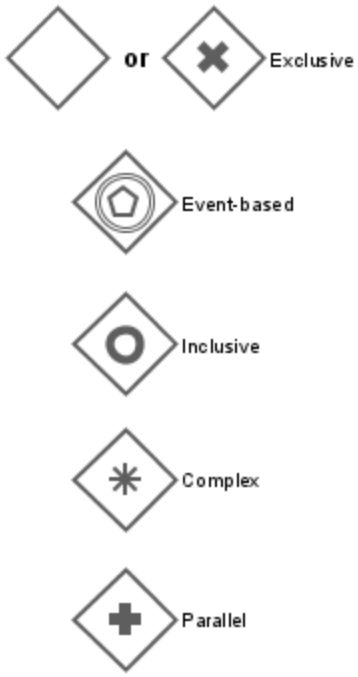
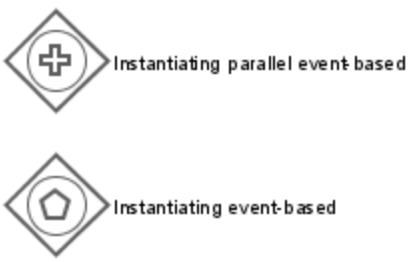


Illustration 170: BPMN gateway types

For event-based gateways there are two additional symbols which are used to start a process:



All in all the Méthode ARIS will provide eight gateway symbols. Contrary to events, an ARIS attribute recording the gateway type is not required. It is up to the modeler to ensure that gateways are used in a semantically correct way. The modeler should not reuse gateways.

10.4.4.1 Exclusive gateway

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Exclusive gateway	inherits from Gateway	Object type: Rule (OT_RULE) Symbols: * Gateway (ST_BPMN_RULE_1) * Exclusive gateway (ST_BPMN_RULE_XOR_3) * Event-based gateway (ST_BPMN_RULE_XOR_4)
	default: SequenceFlow [0..1]	Attribute type **Sequence flow condition** (AT_BPMN_SEQ_FLOW_CONDITION) in the attribute type group **BPMN 2.0 attributes** of the following connection types: * Rule leads to (CT_LEADS_TO_2) event * Rule activates (CT_ACTIV_1) function * Rule links (CT_LNK_2) rule The attribute value must be set to **Default sequence flow**. The symbol (slash) is automatically set by the software.

10.4.4.2 Inclusive gateway

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Inclusive gateway	inherits from Gateway	Object type: Rule (OT_RULE) Symbol: * Inclusive gateway (ST_BPMN_RULE_OR_1)
	default: SequenceFlow [0..1]	See: exclusive gateway

10.4.4.3 Parallel gateway

Parallel gateways have no specific attributes.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Parallel gateway	inherits from Gateway	Object type: Rule (OT_RULE) Symbol: * Parallel gateway (ST_BPMN_RULE_AND_1)

10.4.4.4 Complex gateway

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Complex gateway	inherits from Gateway	Object type: Rule (OT_RULE) Symbol: * Complex gateway (ST_BPMN_RULE_COMPLEX_1)
	activationCondition: Expression [0..1]	Attribute type **Activation condition** (AT_ACTIVATION_CONDITION) in the attribute type group **BPMN 2.0 attributes/Complex gateway attributes** of object type Rule (OT_RULE)

10.4.4.5 Event-based gateways

All attributes are represented by specific gateway symbols.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Event-based gateway	inherits from Gateway	Object type: Rule (OT_RULE) Symbols: * Event-based gateway (ST_BPMN_RULE_XOR_4) * Instantiating event-based gateway (ST_BPMN_RULE_XOR_START) * Instantiating parallel event-based gateway (ST_BPMN_RULE_XOR_PARALLEL)
	instantiate: boolean = False	Represented by symbols. **True** if: * Instantiating event-based gateway (ST_BPMN_RULE_XOR_START) * Instantiating parallel event-based gateway (ST_BPMN_RULE_XOR_PARALLEL) **False** if: * Event-based gateway (ST_BPMN_RULE_XOR_4)
	eventGatewayType: EventGatewayType = Exclusive { Exclusive | Parallel }	Represented by symbols: **Exclusive** if: * Event-based gateway (ST_BPMN_RULE_XOR_4) * Instantiating event-based gateway (ST_BPMN_RULE_XOR_START) **Parallel** if: * Instantiating parallel event-based gateway (ST_BPMN_RULE_XOR_PARALLEL)

10.4.5 Lanes

A lane is a subdivision of a process or a pool. Lanes have no semantics in BPMN. BPMN 2.0 uses lanes as a way to categorizes Flow Elements. Most often lanes represent organizational elements, but in principle any categorization may be used for lanes. Lanes may contain nested sub-lanes. A lane set specifies the categorization represented by the lanes.

See: Business Process Model and Notation (BPMN), version 2.0.

Like a pool a lane is drawn as a rectangular box, its label is not boxed off.

Illustration 171: Nested Lanes

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
LaneSet	inherits from BaseElement	Object type: Lane (OT_BPMN_LANE) Symbol: Lane (ST_BPMN_LANE_1)
	process: Process	The BPMN process model that contains the lane(s).
	lanes: Lane [0..*]	Object type: Lane (OT_BPMN_LANE) Symbol: Lane (ST_BPMN_LANE_1) The source objects in the connection type: Lane belongs to (CT_BELONGS_TO_1) lane
	parentLane: Lane [0..1]	The target object in the connection type: Lane belongs to (CT_BELONGS_TO_1) lane CT: Lane belongs to lane
Lane	inherits from BaseElement	Object type: Lane (OT_BPMN_LANE) Symbol: Lane (ST_BPMN_LANE_1)

Class	BPMN attribute name	Implementation in ARIS
	name: string	Attribute type **Name** (AT_NAME) of object type Lane (OT_BPMN_LANE)
	partitionElement: BaseElement [0..1]	Currently not implemented.
	partitionElementRef: BaseElement [0..1]	Currently not implemented.
	childLaneSet: LaneSet [0..1]	The source objects in the connection type: Lane belongs to (CT_BELONGS_TO_1) lane
	flowElementRefs: FlowElement [0..*]	The source objects in the following **belongs to** (CT_BELONGS_TO_1) connection types: * Function belongs to lane * Event belongs to lane * Rule belongs to lane * Cluster/data model belongs to lane * Information carrier belongs to lane

10.5 Collaboration

See: Business Process Model and Notation (BPMN), version 2.0.

A collaboration shows message exchanges between participants. A collaboration contains at least two pools representing the participants. A pool may include a process (white box) or may be shown as a black box with all details hidden. The message exchanges between the participants are represented by message flows that connect two pools (or the objects within the pools). Only one pool may be represented without a boundary.

The model type **BPMN collaboration diagram (BPMN 2.0)** has been introduced to model collaborations.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Collaboration	inherits from BaseElement and InteractionSpecification	Model type: BPMN collaboration diagram (BPMN 2.0) (MT_BPMN_COLLABORATION_DIAGRAM)
	name: string	Attribute type **Name** of the BPMN collaboration diagram (BPMN 2.0)
	choreographyRef: Choreography [0..1]	Currently not implemented.
	conversationAssociations: ConversationAssociation [0..*]	The relationships to conversations are represented by occurrence copies of participants (OT_BPMN_POOL; ST_BPMN_POOL_1), occurrence copies of message flow connections and the assignment of a BPMN collaboration model (BPMN 2.0) to the object type **Conversation** (OT_BPMN_CONVERSATION).
	conversations: Conversation [0..*]	BPMN collaboration diagram (BPMN 2.0) assigned to the object type **Conversation** (OT_BPMN_CONVERSATION)
	artifacts: Artifact [0..*]	Artifacts (page 196)

Class	BPMN attribute name	Implementation in ARIS
	participantAssociations : ParticipantAssociations [0..*]	The relationships to participants are represented by occurrence copies of participants (OT_BPMN_POOL; ST_BPMN_POOL_1)
	messageFlowAssociations: Message flow association [0..*]	The relationships to message flows are represented by occurrence copies of message flow connections (and the involved participants).
	IsClosed: boolean = false	Attribute type **Is closed** in the attribute type group **BPMN 2.0 attributes** of model type the BPMN Collaboration diagram (BPMN 2.0).

The object types and connection types of the BPMN collaboration diagram are detailed in the following chapters.

10.5.1 Pool and participant

Pools and participants play a central role in collaborations. They are described in detail in chapter Participant (page 209).

10.5.2 Object types and connection types reused from a process

As a pool may show a process (white box) all object types and connection types that are allowed in the BPMN process diagram (BPMN 2.0) are also available in the BPMN collaboration diagram (BPMN 2.0).

The object types and connection types taken over from the BPMN process diagram (BPMN 2.0) are described in detail in chapter Process (page 219).

The connection type **belongs to** is used to embed the object types of a visible process into a pool.

Source object type	Connection type	Target object type
Event (OT_EVT)	belongs to (CT_BELONGS_TO_1)	Participant (OT_BPMN_POOL)
Function (OT_FUNC)	belongs to (CT_BELONGS_TO_1)	Participant (OT_BPMN_POOL)
Rule (OT_RULE)	belongs to (CT_BELONGS_TO_1)	Participant (OT_BPMN_POOL)
Lane (OT_BPMN_LANE)	belongs to (CT_BELONGS_TO_1)	Participant (OT_BPMN_POOL)
Cluster/data model (OT_CLST)	belongs to (CT_BELONGS_TO_1)	Participant (OT_BPMN_POOL)
Information carrier (OT_INFO_CARR)	belongs to (CT_BELONGS_TO_1)	Participant (OT_BPMN_POOL)

10.5.3 Message flow

The message flow between different participants is represented by an ARIS connection type of the same name. It connects two pools or the objects within a pool. The attributes and model associations of message flow are described in chapter Message flow (page 207).

To show the messages being exchanged in message flows the ARIS object type **message** represented by a message symbol is used. The message flow connection type is replaced by

two connection types: **sends** and **is received from**. This work around is required due to the fact that it is not possible in ARIS to assign object types to connection types.

The program will display the **sends** and **is received from** connection types like a normal message flow.

10.6 Conversation

See: Business Process Modeling Notation (BPMN), version 2.0, page 124 and the following.

The BPMN conversation diagram has been introduced with BPMN 2.0 to provide a big picture of the interactions (in terms of related message exchanges) between collaborating participants.

The BPMN conversation diagram is similar to the BPMN collaboration diagram, but its pools are not allowed to contain a process and a choreography is not allowed between the pools.

The BPMN conversation diagram differentiates three basic elements.

- Conversation nodes (Communication, Sub-conversation)
- Participants (Pools)
- Conversation links (message flow, participates in)

They are described in the next chapters.

10.6.1 Conversation container

The attributes and model associations of a conversation and conversation container are summarized in the table below.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Conversation	inherits from CallableElement, InteractionSpecification, ConversationContainer	
	correlationKeys: CorrelationKey [0..*]	Currently not implemented.

Class	BPMN attribute name	Implementation in ARIS
	messageFlowRefs: MessageFlow [0..*]	Occurrence copies of message flows (and the involved participants)
Conversation container	inherits from BaseElement	Model type: BPMN conversation diagram (BPMN 2.0) MT_BPMN_CONVERSATION_DIAGRAM
	conversationNodes: ConversationNode [0..*]	see below
	artifacts: Artifact [0..*]	see chapter Artifacts (page 196)

10.6.2 Conversation nodes

BPMN 2.0 distinguishes three sub-types of conversation nodes.

- Communication
- Sub-conversation
- Call conversation
  (The concept of call conversation is not clear, thus, Call conversations are ignored in the current implementation of BPMN 2.0 in ARIS. However, in ARIS Call conversations can be particularly distinguished. See description below.)

Conversation nodes are represented in ARIS by the object type **Conversation**.

A **communication** is an atomic conversation element in a BPMN conversation diagram, it represents a set of message flow grouped together based on a single correlation key. A communication will involve at least two participants.

The symbol for a communication is available in the **Symbols** bar of the BPMN conversation diagram (BPMN 2.0).

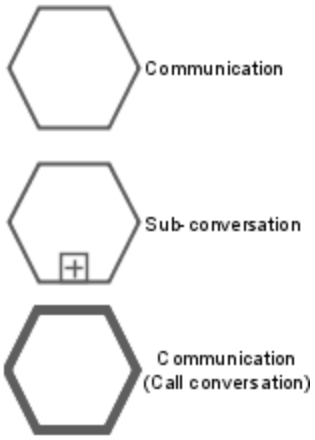


Illustration 172: Symbols of Conversation nodes

A **Sub conversation** is a conversation which consists of lower-level conversations which are modeled in a separate BPMN conversation diagram assigned to the sub-conversation. A sub-conversation shares the participants of its parent conversation.

The ARIS method provides a **Sub conversation** symbol, it also shown in the **Symbols** bar.

A **Call conversation** identifies a place in a conversation where a global conversation or a global communication is used. A **global communication** is a reusable atomic communication definition that can be called from within any conversation by a Call conversation.

The concepts of Call conversations and global communication are very vague. Thus, the ARIS method does not provide specific symbols. But there is the Boolean ARIS attribute type **Call conversation** which allows the modeler to flag Call conversations. If the value is **true**, the **Call conversation** symbol is rendered by the software automatically.

10.6.3 Participant

Participants are represented by the ARIS object type **Participant**. The **Pool** symbol is available in the **Symbols** bar. If the ARIS attribute type **Multi-instance participant** is set to **true** the program will render the symbol: three vertical lines are displayed at the bottom of the pool symbol.

Participants/pools are described in detail in chapter Participant (page 209).

10.6.4 Artifacts

According to the metamodel artifacts are allowed in a conversation diagram. However, the relevance of groupings in a conversation diagram is not evident. For that reason only text annotations are implemented in the current version of the BPMN conversation diagram.

The symbol **Text annotation** is available in the **Symbols** bar. Artifacts and their usage are described in detail in chapter Artifacts (page 196).

10.6.5 Conversation link

A conversation link is used to link participants with conversation nodes. A conversation node has at least two participants.

There is an inconsistency in the specification: Sometimes the name **Communication links** is used, sometimes the name **Conversation link**.

N-ary (n > 2) conversations are allowed.

In ARIS the connection type **participates in** (Participant **participates in** Conversation) has been introduced. The passive name of the connection type is **has conversation link to** (Conversation **has conversation link to** Participant). Specific attributes are not required.

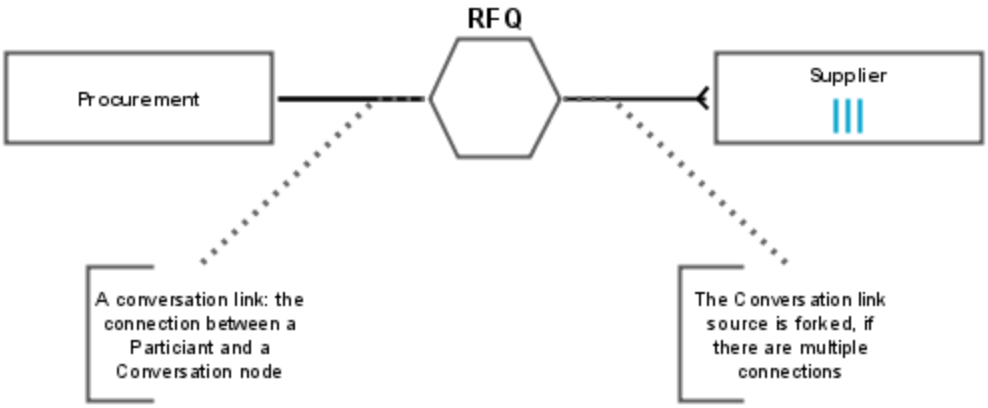


Illustration 173: Conversation link with Participant multiplicity

The fork shown at the source of a conversation link must be manually set by the modeler using the property dialog of the relevant connection type.

10.6.6 Message flow in a conversation

According to the specification, it is allowed in the BPMN conversation diagram to model message exchanges between participants using message flows. (See: Business Process Model and Notation (BPMN), version 2.0.)

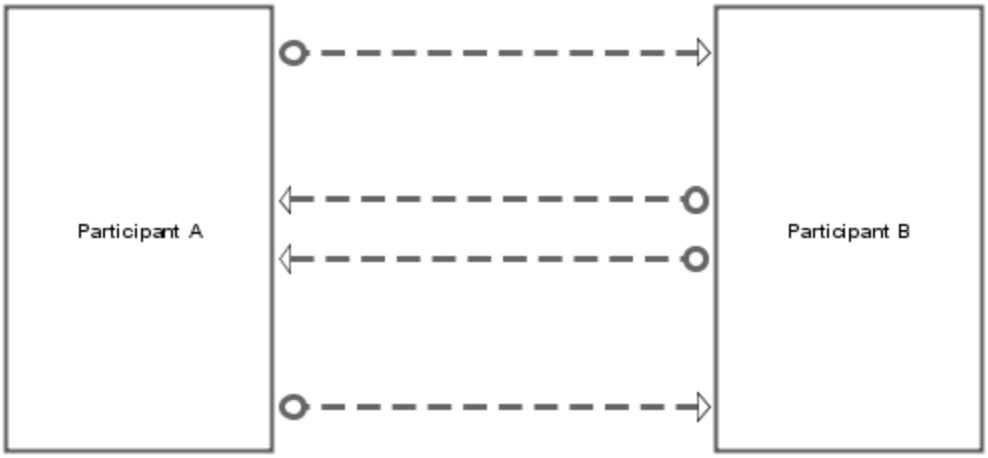


Illustration 174: Message flow between Participants in a BPMN conversation diagram (BPMN 2.0)

Thus, the ARIS connection type **message flow** (Participant **message flow** Participant) is available in the BPMN conversation diagram (BPMN 2.0).

10.6.7 Model assignments

The object type **Conversation** has the following assignments:

- BPMN conversation diagram
- BPMN collaboration diagram.

Only one model of each type can be assigned to a conversation object.

10.7 Enterprise BPMN collaboration diagram

The model type **Enterprise BPMN Collaboration Diagram** is based on the BPMN Collaboration Diagram (BPMN 2.0).

It extents the **BPMN Collaboration Diagram (BPMN 2.0)** model type by ARIS constructs that are already available in the EPC, but that are out of scope in the BPMN specification. Thus, for example, the following object types can be (re-)used as lanes:

- Application system type
- Role
- Organizational unit
- Position
- Group

The **supports** connection is used to nest tasks in a lane object of the object type **Applications system type**. The **carries out** connection is used to nest tasks in the lane objects of the **Organizational** object types.

For all other nestings known from the BPMN specification the **belongs to** connection is used.

Similar to the EPC, an assignment relationship of the connection type **is process-oriented superior** is available between a function assigned to an Enterprise BPMN collaboration diagram and the tasks occurring in the assigned model.

11 Gestion de l'expérience client (CXM)

En raison de la numérisation, le consommateur passe de plus en plus de temps en ligne et les processus de clientèle contiennent de plus en plus souvent aussi des interactions numériques. De ce fait, la gestion de l'expérience client (CXM) devient un des plus importants moteurs d'innovation et de fidélité de la clientèle.

CXM vise à influencer positivement le comportement de consommation du client de même qu'à fournir et à proposer en tout temps à chaque client l'information qui répond à ses besoins, et ce, par le biais des canaux appropriés.

ARIS permet de réaliser des projets CXM selon deux approches différentes. Il est d'une part possible de réaliser un projet CXM selon une approche du haut vers le bas, c'est-à-dire que le projet CXM commence par la définition du paysage du parcours client suivi de la création des cartographies du parcours client. D'autre part, il est aussi possible d'ajouter des points de contact client aux processus internes et de les spécifier plus précisément par exemple dans des diagrammes d'affectation des points de contact client. Le projet CXM ainsi organisé suit une approche du bas vers le haut.

11.1 Paysage du parcours client

Le modèle de type **Paysage du parcours client** permet la description du cycle de vie du client, avec les différentes étapes du cycle de vie et les parcours client. En l'occurrence, on peut utiliser le modèle soit pour modéliser exclusivement les étapes du cycle de vie du client et les affecter à des parcours client, soit pour illustrer dans un modèle les étapes du cycle de vie du client de même que les parcours client s'y rapportant.

Illustration 175: Paysage du parcours client

Si l'on adopte l'approche du haut vers le bas, on commence avec le paysage du parcours client, qui donne un aperçu clair de toutes les étapes du cycle de vie du client et des parcours client s'y rapportant.

Le type d'objet **Parcours client** donne également la possibilité de représenter à l'aide d'attributs le propriétaire du parcours client, l'axe stratégique et l'impact des moteurs d'activité sur la transformation de même que l'expérience client globale.

L'activation des modèles CXM correspondants permet de visualiser par signalisation multicolore les différentes mesures en fonction des diverses valeurs d'attributs définies, de sorte que chaque personne responsable peut voir directement quelles mesures sont nécessaires pour quels parcours client.

11.2 Cartographie du parcours client

Le modèle **Cartographie du parcours client** est un diagramme de colonnes et symbolise un des parcours client du modèle **Paysage de parcours client**. La cartographie du parcours client permet de décrire le « périple » qu'entreprend le client avec l'organisation et qui caractérise ses interactions avec l'entreprise, en représentant les objets **Étape du parcours client** et **Point de contact client**. Le modèle autorise ainsi une description détaillée des points de contact client, entre autres avec les KPI correspondants, les responsabilités organisationnelles, les mesures et les risques.

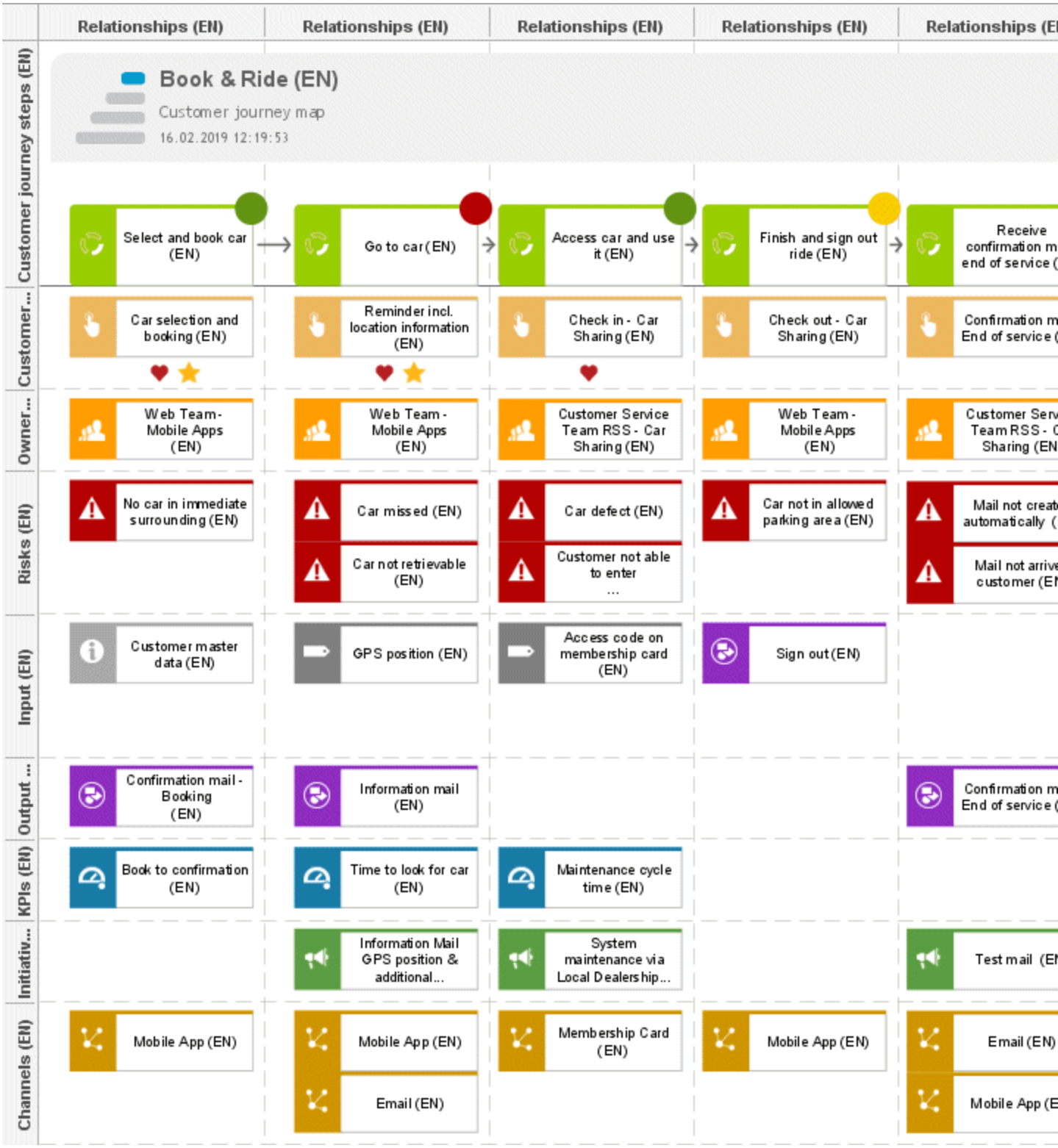


Illustration 176: Cartographie du parcours client

L'objet central du type de modèle est cependant le **point de contact client**, qui peut être décrit avec une multitude d'attributs différents, notamment :

- les objectifs du client
- les attentes du client et

- le ressenti du client

Il est en outre possible d'indiquer pour le point de contact client s'il s'agit d'un moment de vérité, d'une préoccupation ou d'une meilleure pratique. Du fait de son importance, cette information apparaît avec des icônes spéciales dans le modèle.

Illustration 177: Symboles CXM

Le **moment de vérité** indique qu'il s'agit d'un point de contact très important et décisif pour l'entreprise et le processus. Un moment de vérité peut déterminer si une relation entre le client et l'entreprise s'approfondit ou se termine. L'identification de moments de vérité doit donc revêtir une grande priorité, puisqu'ils ont une influence directe sur les affaires.

Pour organiser la modélisation de la cartographie du parcours client de façon aussi simple que possible, ARIS permet de placer les objets avec la fonction glisser-déplacer. Il n'est pas non plus nécessaire de tirer des liaisons de relations dans ce type de modèle, à l'exception près des objets de type **Étape du parcours client**, puisque celles-ci sont automatiquement mises en relation. Tous les objets d'une colonne appartiennent à l'objet **Point de contact client**, celui-ci étant à son tour en relation avec l'objet **Étape du parcours client**.

Au cas où une étape du parcours client possèderait plus d'un point de contact client en raison de canaux multiples, il est possible d'affecter et de décrire de façon détaillée les différents points de contact avec le diagramme d'affectation des points de contact du client. Si vous n'affectez pas les points de contact, tous les objets sous ce point de contact client sont utilisés pour tous les points de contact de cette étape. Vous ne représentez ainsi qu'une seule spécification générale.

11.3 Diagramme d'affectation des points de contact du client

Le modèle **Diagramme d'affectation des points de contact du client** sert à décrire de façon détaillée un point de contact client, avec les KPI correspondants, les unités organisationnelles, les mesures, les risques, etc.

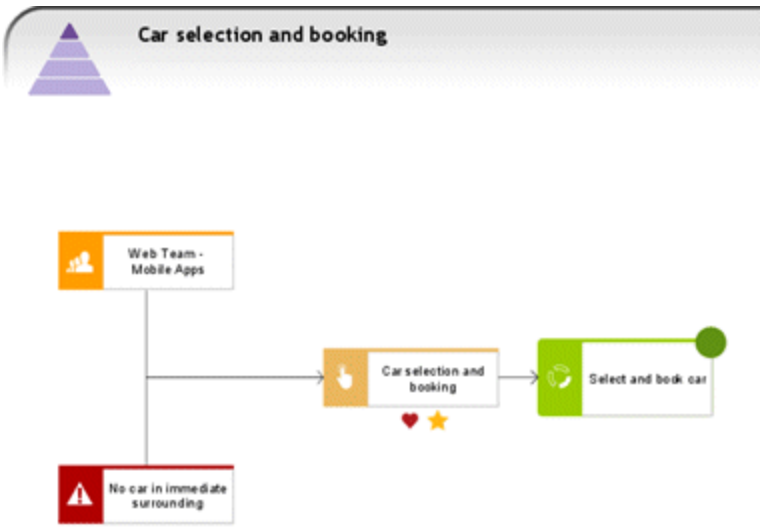


Illustration 178: Diagramme d'affectation des points de contact du client

En plus de l'affectation de l'objet point de contact client dans la cartographie du parcours client, ce modèle s'avère particulièrement utile lorsque les entreprises utilisent déjà ARIS et souhaitent représenter exclusivement les points de contact client dans leurs processus internes mais pas dans des modèles de cartographie du parcours client. Dans ce modèle, les types d'objets offerts pour la représentation détaillée des points de contact client sont les mêmes que dans le modèle **Cartographie du parcours client**.

11.4 Cartographie des points de contact du client

Le modèle **Cartographie des points de contact du client** répertorie tous les points de contact client et peut être utilisé par exemple comme point de départ d'un projet d'expérience client pour identifier les points d'interaction avec le client. Les points de contact client sont regroupés en fonction d'un critère qui peut être important pour l'analyse, par exemple l'unité organisationnelle, le canal ou le risque.

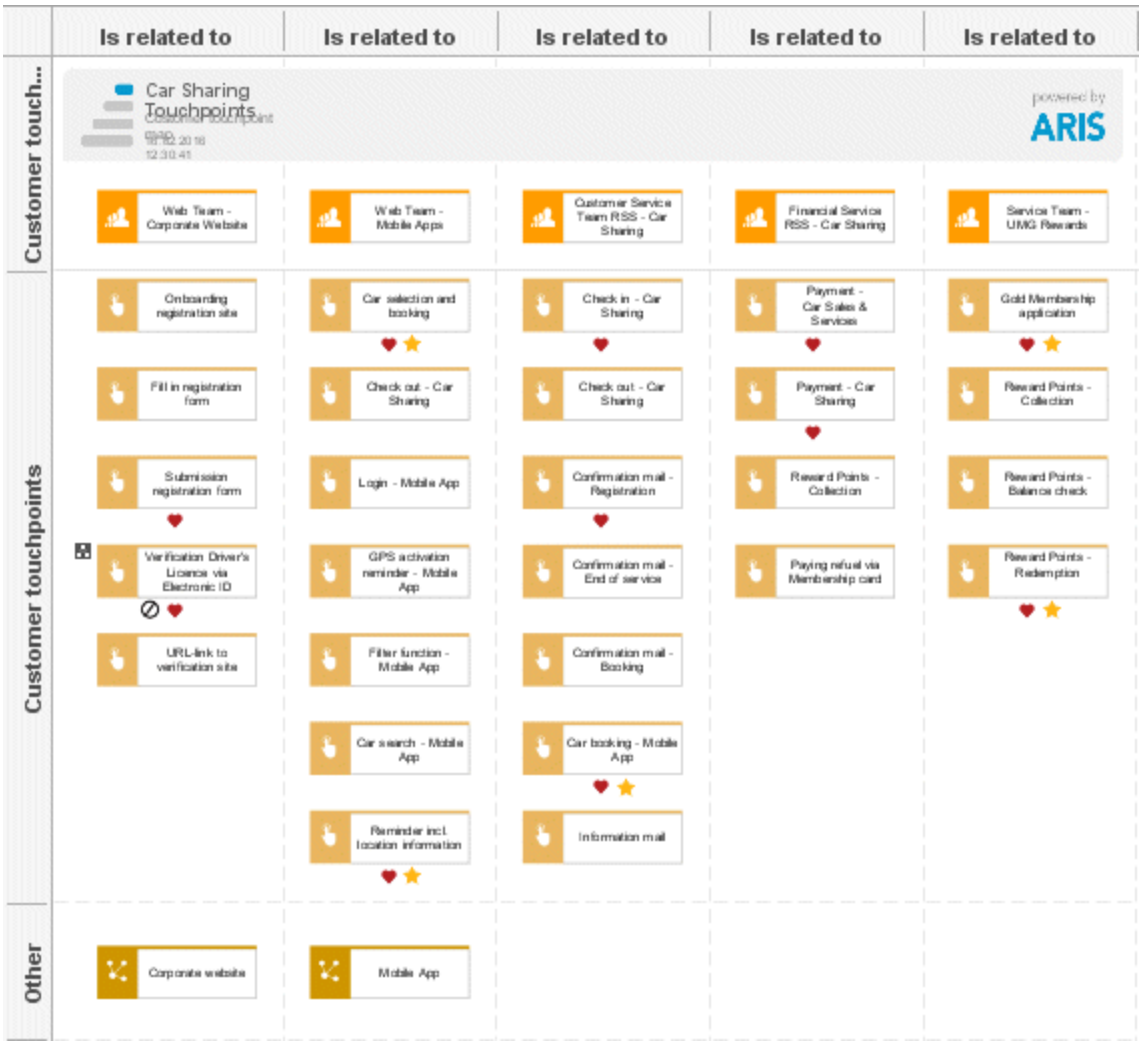


Illustration 179: Cartographie des points de contact du client

La cartographie des points de contact client peut toutefois être aussi utilisée pour donner un aperçu de tous les points de contact d'un projet d'expérience client existant et pour gérer ces points de contact. Elle permet de reconnaître d'un seul coup d'œil quel point de contact client représente un moment de vérité, une préoccupation et/ou une meilleure pratique.

11.5 Connexion entre CXM et BPM

ARIS permet de représenter les perspectives et les processus internes comme externes et de les relier entre eux au moyen de l'objet **Point de contact client**. Si l'entreprise a par exemple un point de contact client défini dans une cartographie du parcours client, elle peut lui donner une valeur et le réutiliser dans un processus interne, par exemple dans un CPE important pour le parcours client. De par cette possibilité, les deux modèles sont mis en relation, ce qui donne un aperçu, par exemple au moyen d'analyses, des processus internes qui seront influencés par des modifications au point de contact.

Illustration 180: Connexion entre CXM et BPM

11.5.1 Possibilités d'analyse

En plus du type de modèle, ARIS donne la possibilité d'analyser les données CXM par rapport ou par requête.

11.5.1.1 Rapport

Le rapport **Analyser satisfaction client** permet de visualiser sous forme d'infographique de quelle manière le client vit ses interactions avec l'entreprise durant un parcours client et aide donc à améliorer la satisfaction du client ainsi qu'à faire face à ses problèmes.

Les informations suivantes sont évaluées et présentées :

- **Étapes du parcours client** avec les **points de contact client**
- **Moment de vérité** et **préoccupations** avec description
- **Meilleure pratique**
- **Importance pour le client** (Importance) & **ressenti du client** (Satisfaction)
- **Part proportionnelle des préoccupations** dans la **cartographie du parcours client**
- Nombre de **processus internes** concernés

- Proportion **des clients satisfaits**

Illustration 181: Infographique CXM

11.5.1.2 Requêtes

Les requêtes vous permettent de visualiser en quelques clics des relations complexes au sein d'une base de données ARIS. Les données et les relations sont représentées graphiquement, mais peuvent aussi être visualisées sous forme de tableaux.

11.5.1.3 Get full customer journey overview

La requête **Get full customer journey overview** est lancée par objet de type **Parcours client**.

La requête recueille toutes les cartographies de parcours client associées des objets de parcours client sélectionnés et présente toutes les informations disponibles sur les étapes de parcours client et les points de contact client associés. Les informations relatives à un point de contact client décrivent le point de contact client en détail, par exemple les risques qui y sont associés.

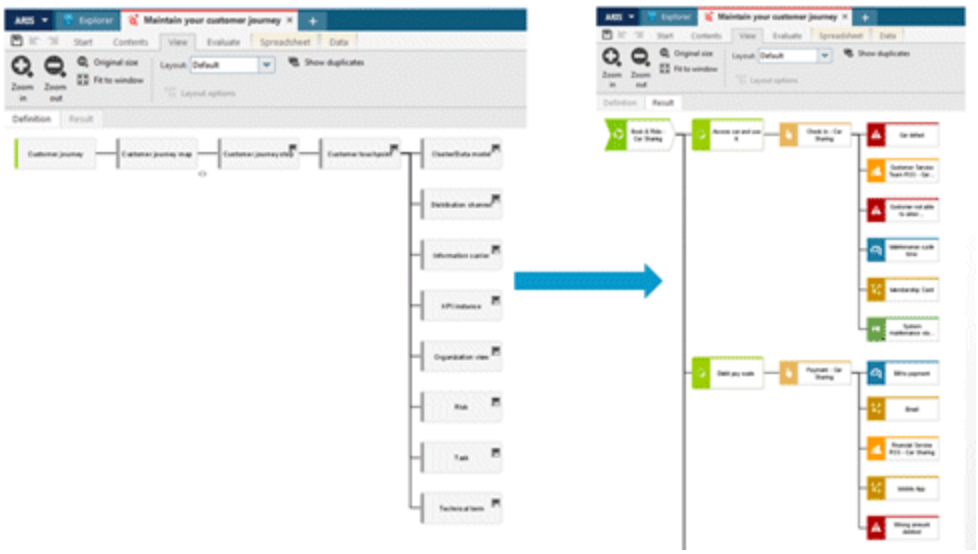


Illustration 182: Cartographie du parcours client-Aperçu

En plus de la représentation graphique, cette requête présente aussi un tableau avec l'aperçu complet du parcours client qui peut être utilisé, entre autres, pour la gestion des données.

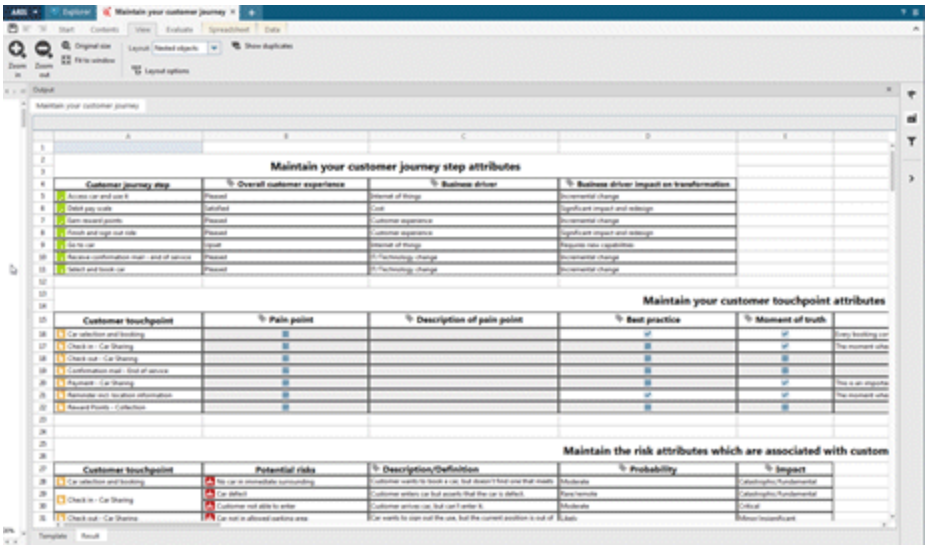


Illustration 183: Vue d'ensemble: Cartographie du parcours client - Tableau

Le tableau comprend des informations sur les différentes étapes du parcours du client, comme l'expérience globale du client, de même que tous les détails importants sur les points de contact client du parcours client sélectionné, par exemple les points de contact constituant une préoccupation ou non.

Le tableau permet non seulement d'obtenir un aperçu, mais aussi de gérer les attributs.

11.5.1.4 Find customer touchpoints clustered by associated risk

Cette requête est lancée par objet de type **Risque**. Elle indique tous les points de contact reliés à un risque afin d'obtenir un aperçu rapide.

Illustration 184: Points de contact client en fonction du risque (requête)

11.5.1.5 Find customer touchpoints clustered by associated ownership

Cette requête est lancée par objet de type **Unité organisationnelle** et indique tous les points de contact reliés à un domaine responsable afin d'obtenir un aperçu rapide.

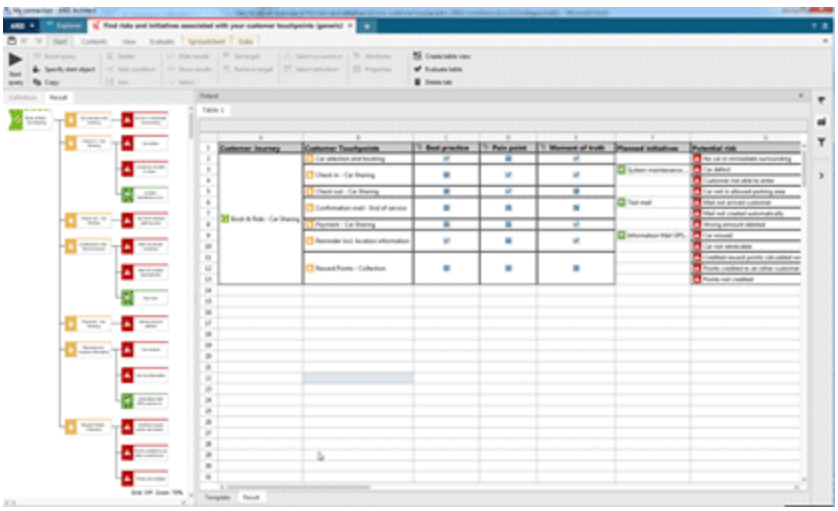


Illustration 185: Risques et initiatives à tous les points de contact client (requête)

11.5.1.6 Find customer touchpoints clustered by associated channel

Cette requête est lancée par objet de type **Canal de distribution** et indique tous les points de contact reliés à un canal de distribution afin d'obtenir un aperçu rapide.

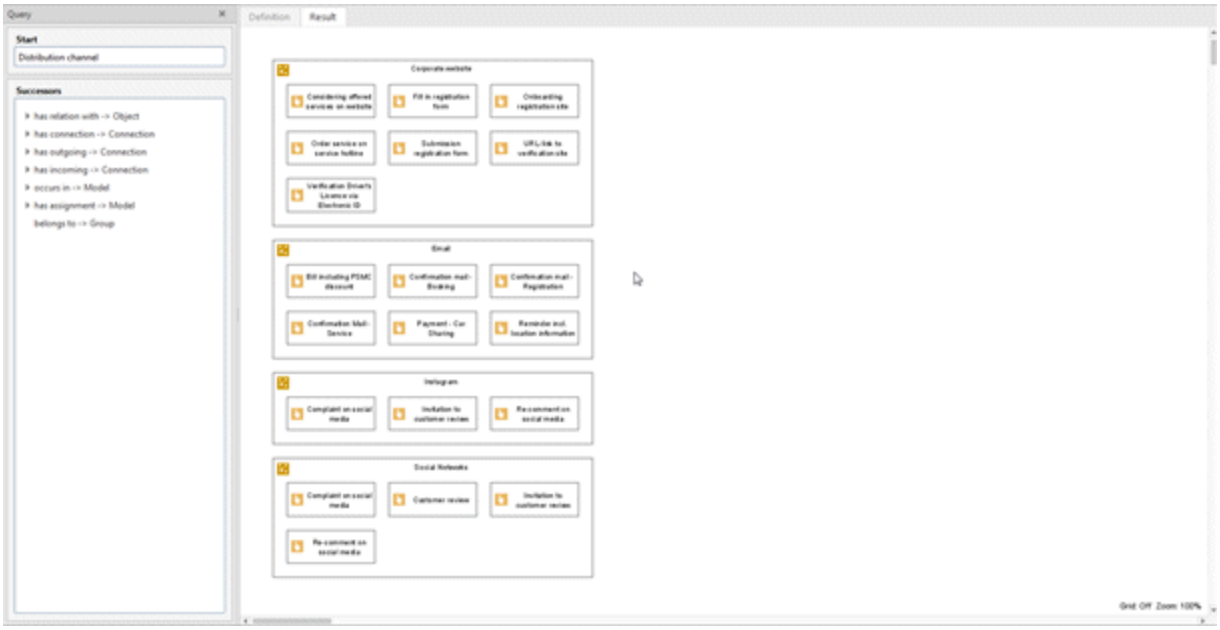


Illustration 186: Points de contact client associés au canal (requête)

11.5.1.7 Find risks and initiatives for all customer touchpoints

La requête **Find risks and initiatives for all customer touchpoints** est lancée par objet de type Parcours client et répertorie toutes les cartographies de parcours du client associées des objets Parcours du client sélectionnés. Elle indique les points de contact client, leurs risques associés et les mesures prévues.

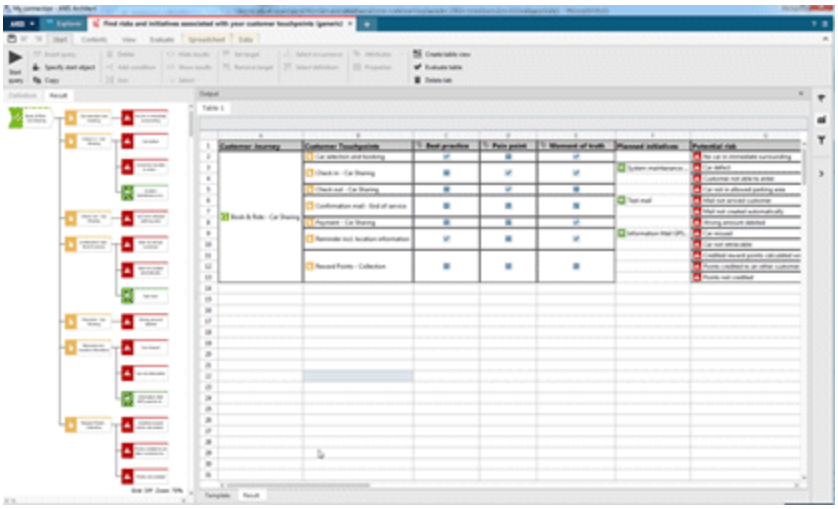


Illustration 187: Risques et initiatives à tous les points de contact client (requête)

Le tableau connexe donne des informations sur les objets Parcours du client et les points de contact client s'y rapportant. Les autres informations sur les points de contact sont les valeurs des attributs Préoccupation, Moment de vérité et Meilleure pratique. Le tableau indique en outre les risques associés aux différents points de contact, de même que les mesures prévues.

11.5.1.8 Find risks and initiatives for bad customer touchpoints only

La requête **Find risks and initiatives for bad customer touchpoints only** est lancée par objet de type Parcours client. Elle comprend des informations semblables à celle de la requête **Find risks and initiatives for all customer touchpoints**, à la différence près que tous les points de contact client ne sont pas évalués, mais seulement ceux qui sont gérés comme pertinents pour l'attribut **Préoccupation** et qui font que le point de contact est négatif pour le client. La requête se concentre aussi sur les risques des différents points de contact et donne des informations sur les risques, par exemple la probabilité de survenue.

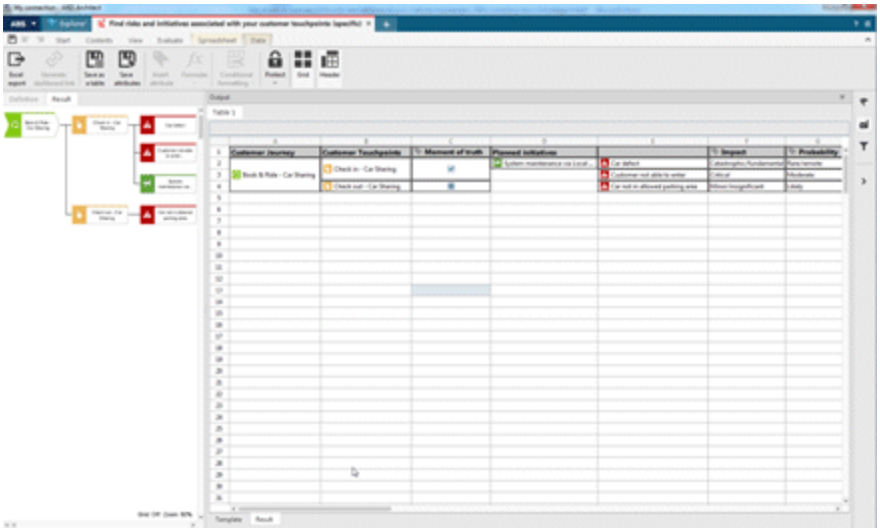


Illustration 188: Risques et initiatives aux points de contact client déficients (requête)

11.5.1.9 Find all processes related to customer journeys

La requête est lancée par fonction d'un diagramme de chaînes de plus-value (DCPV).

Elle permet de trouver des points de contact client dans des processus internes et de déterminer quelle cartographie du parcours client est en relation avec quel processus interne.

La requête indique d'abord tous les DCPV qui sont associés aux fonctions sélectionnées. Elle indique ensuite toute les fonctions des DCPV qui ont une relation avec les points de contact client.

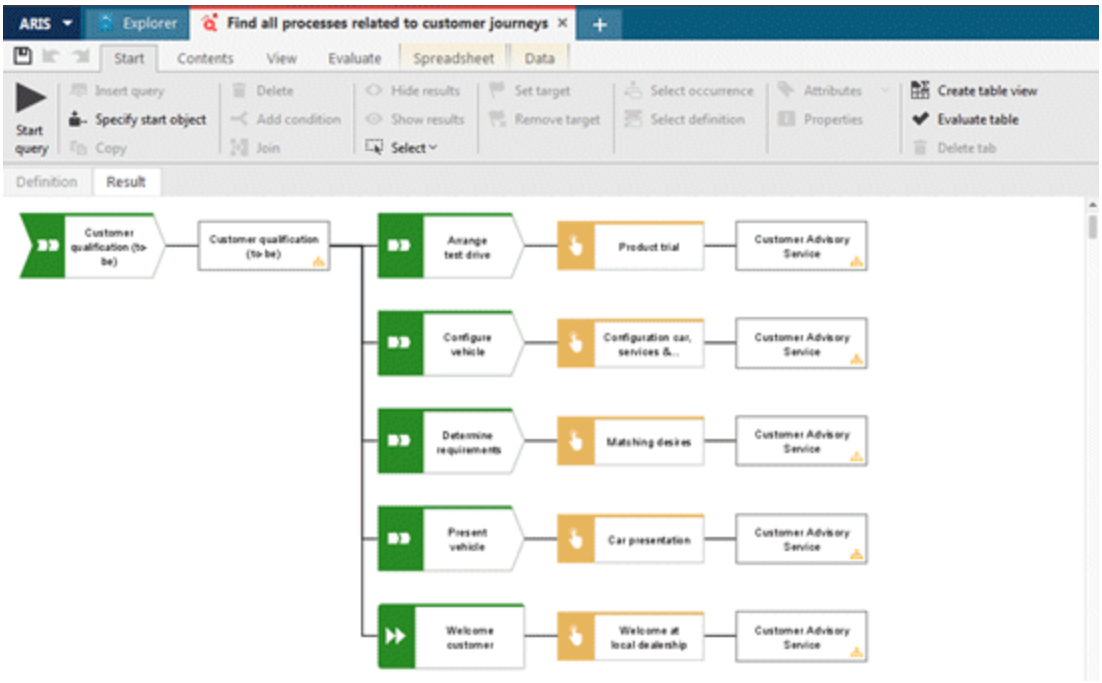


Illustration 189: Parcours client reliés à la fonction

En plus de la représentation graphique des relations, cette requête contient un tableau indiquant les informations suivantes :

- Les DCPV pertinentes
- Le processus qui est en relation avec un point de contact client
- Le point de contact client
- L'information indiquant si le point de contact client est une préoccupation ou un moment de vérité
- Le nom du parcours client dans lequel le point de contact client a une valeur

12 Cas d'application

Ce chapitre doit vous permettre de trouver l'aide d'ARIS adaptée à certains énoncés de problèmes économiques spécifiques. A cette fin, ce chapitre a été divisé en scénarios d'application (sous-chapitres).

Pour chaque scénario d'application, une brève description permet de comprendre le scénario et d'appréhender les opérations réalisées habituellement dans un tel cas. L'utilisateur trouvera par la suite la description de certaines tâches typiques apparaissant dans le scénario. La manière d'utiliser ARIS pour trouver une solution est présentée pour chaque tâche.

La table suivante propose un aperçu des cas d'application décrits et des types de modèles à utiliser.

Scénario	Tâches de scénario	Types de modèles
Documentation générale de l'entreprise (page 300)	Documentation des objectifs de l'entreprise Documentation de la plus-value de l'entreprise Documentation de la structure d'organisation Documentation des fonctions de l'entreprise Documentation de processus Process warehousing	Diagramme des objectifs Diagramme de chaînes de plus-value Organigramme Arbre de fonctions CPE
Gestion de base de données/Data Warehousing (page 302)	Structure des données/Élaboration de bases de données Gestion de base de données/d'accès	MER Modèle de données IEF Diagramme de table Diagramme de classes
Gestion de réseau et de matériel informatique (page 303)	Recherche des besoins en infrastructures d'information Documentation de l'infrastructure d'information Droits d'accès	Topologie de réseau Diagramme de réseau
Calcul de coûts de processus (page 304)	Description des structures de processus et organisationnelles Analyse de centre de coûts	CPE Organigramme

Scénario	Tâches de scénario	Types de modèles
	Calcul de processus	
Gestion de la qualité (page 305)	Élaboration de documentations GQ Processus de certification Documents de certification	Arbre de produits Matrice de choix des produits CPE Diagramme de chaînes de plus-value Modèle de composition Organigramme
Mesures de réorganisation (page 306)	Documentation du projet Réalisation du projet de réorganisation	Diagramme de chaînes de plus-value CPE Organigramme Modèle de produit Modèle de prestation Diagramme des objectifs
Introduction à SAP-R/3 (page 308)	Analyse de phase, spécification (préparation du projet) Conception de phase (Business Blueprint) ; 3 cas de tâches disponibles	CPE Organigramme
Développement et introduction de logiciels (page 309)	Documentation du projet Détermination de systèmes d'application et de modules Description des processus informatiques Développement de l'interface système	Diagramme de chaînes de plus-value Organigramme CPE Use case diagram Diagramme de type d'application Diagramme de déroulement de programme Diagramme de masque
Gestion des connaissances (page 311)	Cartes des connaissances ou pages jaunes Catégorisation des connaissances	Carte des connaissances Diagramme de structure des connaissances CPE

Scénario	Tâches de scénario	Types de modèles
	Traitement des connaissances dans les processus d'entreprise	Diagramme de rattachement de fonctions
Gestion du flux de travail (page 313)	Personnalisation du processus de systèmes de gestion Workflow	CPE Diagramme de rattachement de fonctions Diagramme d'applications Diagramme de type d'application

12.1 Documentation générale de l'entreprise

Les données caractéristiques d'une entreprise, telles que les déroulements et structures, peuvent être documentées de manière adéquate à des fins de formation, présentation et évaluation. Les principales tâches de la documentation d'entreprise sont décrites ci-après.

TÂCHE : DOCUMENTATION DES OBJECTIFS DE L'ENTREPRISE

Support ARIS : pour hiérarchiser les objectifs de l'entreprise et les facteurs de succès correspondants, vous pouvez utiliser les diagrammes des objectifs.

TÂCHE : DOCUMENTATION DE LA PLUS-VALUE DE L'ENTREPRISE

Beaucoup de décisions au sein d'une entreprise sont basées sur l'identification des fonctions qui participent à la création de plus-value.

Support ARIS : les fonctions de l'entreprise qui participent à la création de plus-value peuvent être représentées par un diagramme de chaînes de plus-value. Ce modèle illustre l'ordre des fonctions successives et les fonctions hiérarchisées.

TÂCHE : DOCUMENTATION DE LA STRUCTURE D'ORGANISATION

Support ARIS : la structure d'une entreprise peut être documentée dans des organigrammes qui représentent la hiérarchie et les relations des unités organisationnelles.

TÂCHE : DOCUMENTATION DES FONCTIONS DE L'ENTREPRISE

Support ARIS : dans un arbre de fonctions, les différentes fonctions d'une entreprise peuvent être affichées sous forme d'aperçu. La disposition des fonctions peut être orientée objet, orientée processus ou orientée traitement.

TÂCHE : DOCUMENTATION DE PROCESSUS

Support ARIS : si l'utilisation de modèles de processus par des applications SAP®, des simulations, des workflows, etc. est nécessaire, nous vous recommandons d'utiliser des CPE à des fins de modélisation.

TÂCHE : PROCESS WAREHOUSING

Par 'process warehouse', on entend la saisie, l'enregistrement et la gestion systématiques des connaissances de processus d'entreprise dans un référentiel.

Support ARIS : les CPE sont utiles pour modéliser des connaissances de processus, s'appuyant sur des documents, images et vidéos, de sorte que les modèles puissent être utilisés pour des évaluations complexes, comme une simulation ou un calcul de coûts de processus.

12.2 Gestion de base de données/Data Warehousing

L'enregistrement de données d'entreprise dans des bases de données réduit la gestion redondante de données et permet un accès autonome à des données disponibles dans le monde entier. Le data warehousing garantit la qualité, l'intégrité et la cohérence des données de base. En général, le terme 'Data Warehouse' désigne une base de données isolée des systèmes informatiques opérationnelles qui est mise à disposition de tous les systèmes de l'entreprise qui soutiennent la gestion et qui est caractérisée par une séparation stricte entre les données et systèmes opérationnelles et ceux qui sont utilisés dans les prises de décisions. L'approche Data Warehouse est centrée sur la mise à disposition et le traitement efficaces de grandes quantités de données pour effectuer des évaluations et analyses dans des processus pertinents pour les prises de décisions.

TÂCHE : STRUCTURE DES DONNÉES/ÉLABORATION DE BASES DE DONNÉES

La structure des bases de données est déterminée par les modèles de données sur lesquels elles se fondent.

Support ARIS : la méthode la plus usuelle pour modéliser des données est le modèle entité-relation (MER), qui est le point de départ de l'implémentation d'une base de données relationnelle.

Les tables et champs d'un système de base de données sont décrits dans le diagramme de table.

Les systèmes de bases de données orientés objets peuvent être conçus à l'aide de l'Unified Modeling Language (UML). Dans UML, un diagramme de classes peut être utilisé pour représenter les relations de données statiques.

TÂCHE : GESTION DE BASE DE DONNÉES/D'ACCÈS

Affectation d'utilisateurs et de responsables système aux systèmes de bases de données.

ARIS support: le diagramme d'accès permet de définir, en combinaison avec des relations et composants système, les droits d'accès au système de base de données des unités organisationnelles, postes de travail et personnes.

12.3 Gestion de réseau et de matériel informatique

La gestion de réseau comprend la gestion, le contrôle et la coordination de toutes les ressources (réparties), réseaux de données, ordinateurs, données et applications, permettant de communiquer au sein d'un réseau d'ordinateurs.

TÂCHE : RECHERCHE DES BESOINS EN INFRASTRUCTURES D'INFORMATION

Une organisation structurelle existante doit permettre de discerner les systèmes de communication et d'information pouvant lui offrir un support efficace.

Support ARIS : le type de modèle Topologie de réseau illustre les exigences posées à la structure des systèmes d'information. La représentation des systèmes d'applications, des types de réseaux et des composants matériel ne couvre pas des exemplaires identifiables individuellement (p. ex. PC avec numéro d'inventaire 3423), mais des assimilations sur une base technologique identique.

TÂCHE : DOCUMENTATION DE L'INFRASTRUCTURE D'INFORMATION

La tâche est ici d'illustrer une infrastructure informatique existante ou devant être installée avec des composants matériel concrets, des réseaux et des systèmes d'application.

Support ARIS : il est possible de représenter, à l'aide du diagramme de réseau, une infrastructure informatique comme conversion concrète d'une topologie de réseau.

TÂCHE : AUTORISATIONS D'ACCÈS

La tâche présente consiste à préciser les applications et les utilisateurs demandant l'accès, le type d'accès requis ainsi que les données auxquelles on accède.

Support ARIS : le diagramme d'accès permet de décrire les applications ou les modules d'application qui demandent un accès ainsi que la forme d'accès (écriture, lecture, modification) aux stocks de données et aux supports d'informations.Il indique en outre si les données agissent comme Input ou comme Output. Il est également possible d'illustrer les autorisations d'utilisateurs et les vues que possèdent certains utilisateurs ou groupes d'utilisateurs pour les applications ou les modules d'application.

12.4 Calcul de coûts de processus

Grâce à la saisie et à l'imputation des coûts résultant de la création de prestation de l'entreprise, le calcul des coûts met à la disposition de l'utilisateur une base de disposition ainsi qu'un instrument de contrôle. La modification des structures de coûts et, notamment, la hausse des coûts généraux, permettent de remplacer les méthodes traditionnelles de calcul des coûts par le calcul des coûts de processus. Le calcul des coûts de processus définit les coûts de processus pour l'ensemble des centres de coûts. La planification budgétaire, la transparence des coûts dans les secteurs de prestation indirects, la recherche des prix et le soutien dans les décisions Make-or-Buy sont les atouts majeurs du calcul des coûts de processus.

TÂCHE : DESCRIPTION DES STRUCTURES DES PROCESSUS ET DE L'ORGANISATION

Définition des processus auxquels le calcul des coûts de processus peut être appliqué et description des centres de coûts.

Support ARIS : les processus sont représentés à l'aide des types de modèles standard (CPE par exemple). Dans le cadre du calcul des coûts de processus, il est important de gérer les attributs de temps et d'affecter des unités organisationnelles.

La structure de l'entreprise est décrite par le biais d'un organigramme dans lequel les unités organisationnelles correspondent à des centres de coûts (entre autres avec les attributs **Taux de coûts** et **Prestation**).

TÂCHE : CALCUL DE PROCESSUS

Support ARIS : vous devez disposer au préalable d'une analyse de centres de coûts complète avec détermination des taux de coûts de processus. Il n'est pas nécessaire de gérer d'autres modèles pour effectuer le calcul des processus. Les résultats obtenus sont affichés dans une table de calcul.

12.5 Gestion de la qualité

La gestion de la qualité (GQ) comprend toutes les opérations définissant la politique de la qualité, les objectifs et les responsabilités d'une entreprise. Ces opérations peuvent se concrétiser par la planification de la qualité, l'orientation de la qualité (gestion des processus), le contrôle de la qualité et l'amélioration de la qualité (promotion de la qualité).

TÂCHE : ÉLABORATION DE DOCUMENTATION GQ

Pour garantir la qualité des produits et des processus au sein d'une entreprise, il est nécessaire de créer une documentation appropriée permettant d'évaluer, de comparer et ainsi d'améliorer les produits et les processus.

Support ARIS : la documentation des produits peut s'effectuer grâce à arborescences permettant de classifier ces produits. Cette forme de représentation est de plus en plus utilisée dans le secteur des services et notamment dans l'administration publique. La matrice de choix des produits permet en outre d'illustrer les fonctions de l'entreprise nécessaires à la fabrication de certains produits et de représenter ces mêmes produits ; elle indique également les unités organisationnelles responsables de la fabrication des produits.

Un autre objectif principal de la documentation GQ reste la documentation de déroulements définis à l'aide de CPE, évalués par des rapports et pouvant renvoyer à des documents ou des applications au sein de l'entreprise.

TÂCHE : PROCÉDURE DE CERTIFICATION

Support par des modèles de procédés de la gestion de projet lors de la certification selon des normes nationales et internationales reconnues (p. ex. ISO ou VDA).

Support ARIS : le diagramme de chaîne de plus-value permet de représenter un modèle de procédé destiné à la certification (p. ex. le modèle de procédé ARIS). Les différentes étapes peuvent être décrites plus en détail par l'association d'autres modèles de procédés.

TÂCHE : DOCUMENTS DE CERTIFICATION

Elaboration des documents de qualité nécessaires à la certification.

Support ARIS : le modèle de composition permet de structurer les différentes normes de certification par leurs composants. Pour orienter la qualité, il est possible d'associer des modèles d'entreprise aux différents éléments du modèle de composition. Ces modèles peuvent être, par exemple, des modèles de processus tels que des CPE, des organigrammes ou encore des diagrammes de chaînes de création de plus-value.

Il est possible, à partir d' ARIS Architect, de générer des rapports reconnus comme manuel GQ pour la certification.

12.6 Mesures de réorganisation

Les mesures de réorganisation réalisées pour réduire les coûts ou les temps des processus et améliorer la qualité des résultats ou du travail servent à modifier les processus d'entreprise (process redesign) ou à les reconfigurer complètement (process reengineering).

TÂCHE : DOCUMENTATION DU PROJET

Documentation de la planification, de la procédure et des résultats des mesures de réorganisation.

Support ARIS : les principales étapes du projet de réorganisation peuvent être décrites comme modèle de procédé grâce à un diagramme de chaînes de plus-value.

Chaque opération du projet de réorganisation peut être documentée dans son déroulement à l'aide de CPE.

La structure des personnes et des unités participant au projet peut être représentée par un organigramme.

TÂCHE : RÉALISATION D'UNE RÉORGANISATION

Dans le cadre d'une réorganisation, la préparation du projet et la planification stratégique sont suivies de l'analyse de la situation réelle, du développement du plan cible et de l'implémentation des solutions.

Support ARIS : pour documenter les conditions de base stratégiques, on utilise des modèles de produits et prestations ainsi que des diagrammes des objectifs pour que les principaux champs d'activité de l'entreprise et leurs produits, prestations et groupes de clients soient enregistrés et les facteurs de succès critiques et la hiérarchie des objectifs de l'entreprise représentés.

Pendant l'analyse de la situation réelle, des diagrammes de chaînes de plus-value sont utilisés pour développer d'abord un cadre contenant les principaux processus d'entreprise. Ensuite, ces processus d'entreprise sont enregistrés sous forme de CPE qui sont basées sur des enquêtes effectuées auprès des employés.

Après l'analyse des points faibles, qui considère les temps de cycle, coûts de processus, changements de l'organisation, de systèmes et de médias, les redondances de données, etc., des processus cibles alternatifs sont définis. De manière analogue à l'analyse de la situation réelle, ils sont modélisés à l'aide de CPE.

Pour soutenir l'implémentation, les composants de système, d'organisation et de données sont représentés de manière plus concrète à la fin de la planification des processus cibles. Par exemple, l'information **Traitement de texte** peut, en tant que composant d'application, maintentant être spécifiée comme Microsoft® Word.

Remarque : la phase de l'analyse des points faibles peut être soutenue pour des évaluations réalisées à l'aide d'Simulation.

12.7 Introduction à SAP-R/3

Les potentiels de support d'ARIS pour l'introduction des logiciels standard R/3 de la société SAP AG sont orientés sur le cycle de vie de l'approche introductive ASAP. Outre ASAP, d'autres outils sont également pris en charge, qui s'orientent davantage sur l'optimisation des processus d'entreprise au sens large. Une partie du support ARIS décrit ci-après n'est disponible qu'avec le pack d'extension SAP®.

TÂCHE : ANALYSE DE PHASE, SPÉCIFICATION (PRÉPARATION DU PROJET)

Il s'agit ici d'identifier le degré de couverture des processus d'entreprise grâce au système SAP® et d'identifier en temps voulu les éventuels points faibles.

Support ARIS : si l'analyse n'est pas effectuée directement par le modèle de référence SAP R/3, les processus "optimaux du point de vue économique" à supporter peuvent être modélisés avec ARIS (cf. documentation générale d'entreprise). Des rapports offrent ensuite une première estimation du degré de couvrement en mappant le modèle d'entreprise avec le modèle de référence SAP R/3.

TÂCHE : CONCEPTION DE PHASE (BUSINESS BLUEPRINT)

Des déficits de processus et/ou de fonctions ont été identifiés dans le modèle de référence SAP R/3.

Support ARIS : les composants issus du modèle de référence SAP R/3 permettent de déduire de nouvelles variantes de processus et de scénario dans ARIS. Par ailleurs, il est possible d'ajouter des fonctions, des événements et des règles aux composants des processus et des scénarios (en développant, le cas échéant, de nouvelles fonctions SAP® ABAP).

TÂCHE : CONCEPTION DE PHASE (BUSINESS BLUEPRINT)

Élaboration d'interfaces entre SAP® R/3 et des applications non SAP®.

Support ARIS : la documentation des attributs à échanger peut être décrite plus amplement dans des modèles de processus d'ARIS Repository par des affectations de données ou d'objets à des fonctions et des modèles de données ou d'objets. Ces informations peuvent être transmises sous forme de rapport et constituent la base de la conception d'interfaces.

TÂCHE : CONCEPTION DE PHASE (BUSINESS BLUEPRINT)

Développement de la conception organisationnelle des processus d'entreprise et du système SAP®.

Support ARIS : l'utilisation d'organigrammes permet de représenter la hiérarchie sociale de l'entreprise et la structure ® et de les comparer entre elles.

12.8 Développement et introduction de logiciels

TÂCHE : DOCUMENTATION DU PROJET

Documentation de la planification, du procédé et des résultats du développement et de l'introduction de logiciels.

Support ARIS : les principales étapes du projet peuvent être décrites comme un modèle de procédé grâce à un diagramme de chaînes de plus-value.

Chaque opération du projet de développement et d'introduction peut être documentée dans son déroulement à l'aide de CPE.

La structure des personnes et des unités participant au projet peut être représentée par un organigramme.

TÂCHE : DÉTERMINATION DE SYSTÈMES D'APPLICATION ET DE MODULES

La tâche consiste ici à représenter la configuration structurelle d'un système d'informations en se fondant sur les exigences du système.

Support ARIS : le diagramme Use case peut servir à identifier les cas d'application du système de logiciel à développer. Les utilisateurs système continuent à être définis et affectés à différents cas d'application. Le diagramme Use Case constitue souvent le point de départ d'une modélisation détaillée du processus. Des modèles de processus peuvent être associés aux différents use cases.

Le diagramme de types de systèmes d'application permet de décrire la structure hiérarchique de systèmes d'application par des types de modules et des types de fonctions informatiques.

Il est également possible d'utiliser le diagramme des systèmes d'application pour une description spécifique des différents types à travers des valeurs concrètes.

TÂCHE : DESCRIPTION DES PROCESSUS INFORMATIQUES

Cette tâche consiste à décrire le déroulement chronologique au sein de modules ou pour l'ensemble des modules.

Support ARIS : le diagramme de déroulement de programmes permet de modéliser des processus informatiques.

TÂCHE : DÉVELOPPEMENT DE L'INTERFACE SYSTÈME

Cette tâche consiste à développer et à documenter l'interface utilisateur.

Support ARIS : la configuration géographique et fonctionnelle d'un masque d'écran (fenêtre) peut être décrite à l'aide d'un diagramme de masque. Le diagramme de masque est la base de la dérivation du code de programme.

12.9 Gestion des connaissances

L'hypothèse à la base de la constitution d'une gestion complète des connaissances suppose que ces connaissance sont devenues, ou deviennent, le facteur de production dominant dans l'entreprise. Il en résulte la nécessité d'aborder ces connaissances de la même façon que les facteurs de production économiques classiques.

La gestion des connaissances est donc responsable du rassemblement, de la représentation et de la distribution des connaissances. La gestion des connaissances est la somme des méthodes, des mesures et des systèmes utilisés par l'entreprise pour développer ses connaissances, en augmenter la transparence et en assurer la disponibilité indépendamment des horaires, des personnes et du lieu. L'objectif de la gestion des connaissances est d'augmenter le capital en connaissances et d'utiliser au mieux cette base cognitive au sein de l'entreprise.

TÂCHE : CARTE DES CONNAISSANCES OU PAGES JAUNES

La tâche est ici d'illustrer la localisation et le type de connaissances dans l'entreprise.

Support ARIS : le type de modèle **Carte des connaissances** permet de représenter la répartition organisationnelle des différentes catégories des connaissances. Ce diagramme peut représenter l'unité organisationnelle, le service ou le collaborateur qui dispose du savoir-faire nécessaire dans des catégories de connaissances précises ainsi que l'importance de cette compétence.

TÂCHE : CATÉGORISATION DES CONNAISSANCES

Cette tâche consiste à classifier le capital intellectuel d'une organisation de façon analytique, c'est-à-dire à représenter les divers groupes et catégories de connaissances pour concevoir, p. ex. une structure de classement des connaissances.

Support ARIS : le diagramme de structure des connaissances permet de représenter la répartition de la base des connaissances d'une organisation en différentes catégories de connaissances et la façon dont celles-ci se divisent à leur tour en catégories de connaissances et connaissances documentées. Pour les connaissances documentées, cette méthode permet également d'indiquer les supports d'informations sur lesquels ces connaissances sont enregistrées.

TÂCHE : TRAITEMENT DES CONNAISSANCES DANS LES PROCESSUS D'ENTREPRISE

Cette tâche consiste à représenter l'emplacement des processus d'entreprise dans lequel les connaissances sont créées, modifiées et requises pour garantir une utilisation aussi efficace que possible de la ressource Connaissances.

Support ARIS : dans les types de modèles **CPE**, **Diagramme de chaînes de processus** et **Diagramme de rattachement de fonctions**, il est possible d'utiliser les objets **Catégorie de connaissances** et **Connaissances documentées**. La structure et la répartition organisationnelle des connaissances peuvent être décrites séparément par le diagramme de structure des connaissances et la carte des connaissances.

12.10 Gestion du flux de travail

Un flux de travail (Workflow) peut être interprété au sens large comme un processus d'entreprise. Le terme Workflow (flux de travail) symbolise des processus de répartition du travail initiés pour le déroulement d'opérations commerciales. Il peut s'agir de processus d'entreprise très simples ainsi que de processus complexes exécutés au sein des divers services de l'organisation. L'analyse des flux de travail est centrée sur le déroulement dynamique des processus, du début à la fin. La gestion du flux de travail est l'ensemble des méthodes, mesures et systèmes qui sont utilisés pour développer, contrôler et optimiser des workflows.

Un système de gestion de flux de travail est un logiciel librement configurable, basé sur un ensemble de règles organisationnelles et opérant de façon active, qui gère une procédure s'étendant sur plusieurs postes de travail et qui inclut des composants de base techniques existants. Des systèmes de contrôle d'opération permettent de soutenir des tâches complexes qui sont traitées par un grand nombre de collaborateurs et postes.

TÂCHE : PERSONNALISATION DU PROCESSUS DE SYSTÈMES DE GESTION DU FLUX DE TRAVAIL

ARIS soutient, en particulier, la transformation de modèles de processus d'entreprise généraux en des modèles de workflow, qui permettent de configurer de façon (semi-)automatique différents systèmes de gestion du flux de travail.

Support ARIS :la CPE présente le flux des activités de manière analogue à la modélisation de processus. Tenez compte du fait que la modélisation doit être strictement conforme à la méthode ! Pour chaque fonction, il faut créer un diagramme de rattachement de fonctions qui lui affecte un éditeur et les données d'entrée et de sortie (si cela n'a pas déjà été modélisé dans la CPE).

Pour que les applications qui appartiennent aux données soient automatiquement lancées pendant l'exécution, il faut modéliser une affectation des fichiers aux applications dans un diagramme d'applications ou un diagramme de type d'application.

13 Bibliographie

13.1 Bibliographie générale

- Brombacher, R.; Bungert, W.: Praxis der Unternehmensmodellierung 1992

  Praxis der Unternehmensmodellierung, Seminar der IDS Prof. Scheer GmbH, Bad Soden/Taunus, 12/13 novembre 1992.
- Chen, P. P.: Entity-Relationship Model 1976

  The Entity-Relationship model: Toward a Unified View of Data, in: ACM Transactions on Database-Systems, Vol. 1 (1976), No. 1, pages 9 - 36. 1 (1976), No. 1, P. 9 - 36.
- Hoffmann, W.; Kirsch, J.; Scheer, A.-W.: Modellierung mit Ereignisgesteuerten Prozeßketten 1993

  Modellierung mit Ereignisgesteuerten Prozeßketten (manuel Méthode, version : décembre 1992), dans : Scheer, A.-W. (Ed.), Veröffentlichungen des Instituts für Wirtschaftsinformatik (publications de l'institut d'informatique de gestion), fascicule 101, Sarrebruck, janvier 1993.
- Keller, G.; Hechler, H.-J.: Informationsmodell 1991

  Konzeption eines integrierten Informationsmodells für die Kostenrechnung des SAP®-Systems, dans : Scheer, A.-W. (Ed.): 12. Saarbrücker Arbeitstagung Rechnungswesen und EDV: Kritische Erfolgsfaktoren in Rechnungswesen und Controlling, Heidelberg 1991, pages 67 - 106.
- Scheer, A.-W.: Architektur integrierter Informationssysteme 1992

  Architektur integrierter Informationssysteme - Grundlagen der Unternehmensmodellierung, 2ème édition, Berlin et al. 1992.
- Scheer, A.-W.: EDV-orientierte Betriebswirtschaftslehre 1990

  EDV-orientierte Betriebswirtschaftslehre - Grundlagen für ein effizientes Informationsmanagement, 4ème édition, Berlin et al. 1990.
- Scheer, A.-W.: Wirtschaftsinformatik 1994

  Wirtschaftsinformatik - Referenzmodelle für industrielle Geschäftsprozesse, 5ème édition, Berlin et al. 1994.
- Schlageter, G.; Stucky, W.: Datenbanksysteme 1983

  Datenbanksysteme: Konzepte und Modelle, 2ème édition, Stuttgart 1983.
- Seubert, M.: SAP®-Datenmodell 1991

  Entwicklungsstand und Konzeption des SAP®-Datenmodells, in: Scheer, A.-W. (Hrsg.): Datenbanken 1991 - Praxis relationaler Datenbanken: Vom Datenmodell zur Implementierung, congrès du 04.-05.06.1991 à Sarrebruck, Sarrebruck 1991, pages 87 - 109.
- Sinz, E. J.: Entity-Relationship-Modell 1990

Das Entity-Relationship-Modell (ERM) und seine Erweiterungen, dans : HMD Theorie und Praxis der Wirtschaftsinformatik, 27 (1990), fascicule 152, pages 17 - 29.

- Scheer, A.-W.: ARIS - Vom Geschäftsprozeß zum Anwendungssystem. 3. édition et al. 1998.
- Scheer, A.-W.: ARIS - Modellierungsmethoden, Metamodelle, Anwendungen. 3. édition et al. 1998.
- Scheer, A.-W., Jost, W.: ARIS in der Praxis 2002

  Gestaltung, Implementierung und Optimierung von Geschäftsprozessen, Berlin, Heidelberg, New York 2002.
- Scheer, A.-W., Abolhassan, F., Jost, W., Kirschmer, M.: Business Process Excellence 2002 ARIS in Practice , Berlin, Heidelberg, New York 2002.

13.2 Bibliographie relative aux sujets

13.2.1 Unified Modeling Language dans ARIS

13.2.1.1 Spécification UML

Spécification UML : https://www.uml.org.

13.2.1.2 Utilisation d'UML

- Burkhardt, R.: UML Unified Modeling Language, Objektorientierte Modellierung für die Praxis, Bonn 1997.
- Fowler, M.; Scott, K.: UML Distilled - Applying the Standard Object Modeling Language, Reading et al. 1997. 1997.
- Oesterreich, B.: Objektorientierte Softwareentwicklung mit der UML, 3ème édition, Munich-Vienne 1997.

13.2.1.3 UML et la modélisation de processus d'entreprise

- Ambler, S. W.: What's Missing from the UML? Techniques that can help model effective business applications, Object Magazine 7(1997)8
- Loos, P.; Allweyer, Th.: Process Orientation and Object-Orientation - An Approach for Integrating UML and Event-Driven Process Chains (EPC), Publication of the Institut für Wirtschaftsinformatik, Paper 144, Sarrebruck 1998.

13.2.2 Méthodes de gestion des connaissances

13.2.2.1 Gestion des connaissances en général

- Probst, G.; Raub, S.; Romhardt, K.: Wissen managen. Wie Unternehmen ihre wertvollste Ressource optimal nutzen, Francfort Wiesbaden 1998. Frankfurt Wiesbaden 1998.
- Bürgel, H. D. (Ed.): Wissensmanagement. Schritte zum intelligenten Unternehmen. Berlin et al. 1998.

13.2.2.2 Utilisation d'ARIS pour la gestion des connaissances

- Allweyer, Th.: Modellbasiertes Wissensmanagement. Dans : IM Information Management & Consulting 13 (1998) 1, pages 37-45.
- Allweyer, Th.: Wissensmanagement mit ARIS-Modellen. Dans : Scheer, A.-W.: ARIS - Vom Geschäftsprozeß zum Anwendungssystem. 3. édition Berlin et al. 1998, pages 162-168.

13.2.3 Méthode du tableau de bord prospectif

Kaplan, Robert/Norton, David: The Balanced Scorecard-Measures that drive Performance, Harvard Business Review janvier/février 1992.

13.2.4 IT City Planning

- Schulman, Jeff: A New View of Architectures Needed for New Business Drivers, Gartner Briefing Presentations.
- Longépé, Christophe : Le projet d'urbanisation du système d'information, Dunod, Paris, 2001

13.2.5 Business process modeling

Business Process Modeling Notation, Working Draft (1.0) ; BPMI.org ; 25 août, 2003.

14 Informations légales et d'assistance

Ce chapitre donne des informations générales concernant le support sur les produits et les informations légales.

14.1 Étendue de la documentation

Les informations fournies sont le reflet des configurations et fonctionnalités en leur état à la date de la publication. Vu que le processus de réalisation de la documentation et celui du logiciel sont soumis à des cycles de production différents, il est possible que la description des configurations et des fonctionnalités diffèrent du contexte effectif. Les notes de publication qui accompagnent ce produit donnent les informations requises quant à ces différences. Veuillez lire les notes de publication et vous informer concernant l'installation, la configuration et l'utilisation du produit.

Si vous souhaitez installer des fonctionnalités système techniques et/ou commerciales sans recourir aux services de consulting fournis par SAG ARIS GmbH, vous devez disposer de connaissances approfondies sur le système à installer, le but à atteindre, ainsi que sur les systèmes cibles et leur interopérabilité. Face à la diversité des plateformes et des configurations matérielles et logicielles interdépendantes, nous ne pouvons décrire que certaines installations spécifiques. Il n'est pas possible de traiter de toutes les configurations et interdépendances.

Veuillez tenir compte des instructions des fabricants en cas de combinaison de différentes technologies, notamment des communiqués récents sur leurs sites Internet concernant les autorisations. Nous ne pouvons dès lors pas garantir l'installation et le fonctionnement parfait de systèmes tiers validés et nous ne fournissons pas d'assistance dans ce cadre. Veuillez toujours respecter les consignes mentionnées dans les manuels d'installation des différents fabricants. En cas de problème, veuillez vous adresser au fabricant et éditeur concerné.

Si vous avez besoin d'aide pour installer des systèmes tiers, contactez votre organisation de vente ARIS (https://aris.com) locale. Veuillez noter que ce type de personnalisation spécifique à l'éditeur ou au client n'est pas couvert par le contrat de maintenance logicielle ARIS standard et ne peut être réalisé que sur demande spéciale et après accord.

14.2 Support

SUPPORT PRODUIT

Nous proposons un support concernant les produits ARIS à tous les clients bénéficiant d'un contrat de support valide.

Vous trouverez des informations sur la disponibilité et la prise en charge des versions des produits, ainsi qu'aux dates de fin de prise en charge dans l'ARIS Community (https://ariscommunity.com/support/product-version-availability).

Contactez nos services d'assistance sous ARIS Community pour soumettre et mettre à jour des incidents de support (https://ariscommunity.com/support) ou publier des idées et des demandes de fonctionnalité (https://aris.ideas.aha.io).

COMMUNAUTÉ

Enregistrez-vous sur ARIS Community pour télécharger des produits, des mises à jour et des correctifs, trouver des informations pertinentes et interagir avec d'autres utilisateurs.

DOCUMENTATION

Pour obtenir des informations sur l'activation et l'utilisation des produits, et poser d'autres questions techniques au support, visitez le site web de documentation ARIS (https://docs.aris.com/).

Si vous souhaitez transmettre un feed-back sur un sujet abordé dans la documentation, écrivez à l'adresse documentation@aris.com.