

Généralités

Définitions

Intervenants

Cycles de vie du logiciel

Quelques méthodes

Exemple « jouet »

Des modèles

Organigramme - Fiche de description de fonction - Modèle fonctionnel - Arbre de décomposition fonctionnel - Diagramme de flots de données - Graphe acteurs-flux - Modèle de flux (modèle de contexte, modèle de flux conceptuel, modèle de flux organisationnel)

Réseaux de Pétri - Diagramme de circulation de l'information (des documents) - Modèle conceptuel des traitements (analytique) - Modèle organisationnel des traitements (analytique) - Modèle logique des traitements (répartis) - Schéma d'architecture logique des moyens informatiques - Actigramme - Modèle dynamique - Diagramme de suivi d'événements - Diagramme de flots d'événements entre objets - Diagramme de flots d'événements entre classes - Diagramme d'états (structuré) - Cycle de vie d'un objet - Diagramme de structure Table de décision -

Modèle entité-association - Modèle organisationnel des données - Modèle logique des données (réparties) - Modèle navigationnel - Modèle relationnel - Modèle objet - Datagramme - Grille d'analyse des rubriques - Fiches de description de rubrique/fichier/document

...

Une démarche

Étude préalable

Analyse fonctionnelle

Analyse organique

Des méthodes

MERISE - MERISE/2 - SSADM - SADT - HOOD - OMT - Z

ACSIOME

Modélisation dans la conception des systèmes d'information
MASSON, 1989

M. ADIBA, C. DELOBEL

Bases de données et systèmes relationnels
DUNOD, 1983

G. BOOCH

Analyse et Conception orientées objets
Addison Wesley, 1994

M. BOUZEGHOUB, G. GARDARIN, P. VALDURIEZ

Les Objets : concepts, langages, bases de données, méthodes, interfaces
EYROLLES, 1997

CGI (Compagnie Générale d'Informatique)

MERISE ou l'informatique avec méthode
NATHAN, 1986

E. DOWNS, P. CLARE, I. COE

SSADM, application and context
Prentice Hall International, 1992

P. DUMAS, G. CHARBONNEL

La méthode OSSAD, pour maîtriser les technologies de l'information, tome 1 : principes
Les Éditions d'Organisation, 1990

P. DUMAS, G. CHARBONNEL, F. CALMES

La méthode OSSAD, pour maîtriser les technologies de l'information, tome 2 : guide pratique
Les Éditions d'Organisation, 1990

GALACSI

Les systèmes d'information : analyse et conception
DUNOD, 1984

GALACSI

Comprendre les systèmes d'information : exercices corrigés d'analyse et de conception
DUNOD, 1985

G. GARDARIN

Bases de données : les systèmes et leurs langages
EYROLLES, 1983

IGL Technology

SADT, un langage pour communiquer

EYROLLES, 1989

I. JACOBSON

Le génie logiciel orienté objet

Addison Wesley

N. KETTANI, D. MIGNET, P. PARÉ, C. ROSENTHAL-SABROUX

De Merise à UML

EYROLLES, 1998

M. LAI

Conception orientée objet, pratique de la méthode HOOD

DUNOD, 1991

J-L. LE MOIGNE

Les systèmes d'information dans les organisations

Presses Universitaires de France, 1973

J-L. LE MOIGNE

Les systèmes de décision dans les organisations

Presses Universitaires de France, 1974

J-L. LE MOIGNE

La théorie du système général - Théorie de la modélisation

Presses Universitaires de France, 1977

D. LIGHTFOOT

Spécification formelle avec Z

TEKNEA, 1994

J. MELESE

Analyse modulaire des systèmes

Éditions Hommes et Techniques, 1977

P.-A. MULLER

Modélisation objet avec UML

EYROLLES, 1997

G. PANET, R. LETOUCHE

MERISE/2, modèles et techniques MERISE avancés

Les Éditions d'Organisation, 1994

R. REIX

L'analyse en informatique de gestion, tome 1 : principes méthodologiques
DUNOD, 1971

R. REIX

L'analyse en informatique de gestion, tome 2 : éléments techniques et applications
DUNOD, 1971

A. ROCHFELD, J. MOREJON

La méthode MERISE, tome 3 : gamme opératoire
Les Éditions d'Organisation, 1989

J. RUMBAUGH, M. BLAHA, W. PREMERLANI, F. EDDY, W. LORENSEN

Object oriented modeling and design
Prentice Hall, 1991

I. SOMMERVILLE

Le génie logiciel et ses applications
InterÉditions, 1985

H. TARDIEU, A. ROCHFELD, R. COLLETTI

La méthode MERISE, tome 1 : principes et outils
Les Éditions d'Organisation, 1983

H. TARDIEU, A. ROCHFELD, R. COLLETTI, G. PANET, G. VAHEE

La méthode MERISE, tome 2 : démarches et pratiques
Les Éditions d'Organisation, 1985

C. TESSIER

La pratique des méthodes en informatique de gestion
Les Éditions d'Organisation, 1995

P.L. WEAVER

Practical SSADM version 4, a complete tutorial guide
Pitman Publishing, 1993

Analyse et Conception des Systèmes d'Information (ACSI)

Analyse

Processus d'examen de l'existant

Conception

Processus de définition de la future application informatique

Systèmes d'Information

Ensemble des moyens (humains et matériels) et des méthodes se rapportant au traitement de l'information d'une organisation

Informatique

Science du traitement automatique et rationnel de l'information

Informatique de Gestion

Informatisation des systèmes d'information

Logiciel

(selon l'arrêté du 22 décembre 1981)

Ensemble des programmes, procédés et règles, et éventuellement de la documentation, relatifs au fonctionnement d'un ensemble de traitements de l'information

Génie Logiciel (ou l'ingénierie des systèmes d'information)

(selon l'arrêté du 30 décembre 1983)

Ensemble des activités de conception et de mise en œuvre des produits et des procédures tendant à rationaliser la production du logiciel et de son suivi

L'information est indispensable dans le processus de décision d'une organisation

- Diminution de l'incertitude

- Liberté de choix

- Cohésion de l'organisation

- Évolutivité par rapport à l'environnement

Les qualités requises pour une information

- Pertinence

 - Précision

 - Sécurité

 - Intégrité

 - Confidentialité

 - Non redondance

 - Convivialité

 - Âge

 - Fréquence

- Cohérence

- Rentabilité

Les types d'information

- Niveau d'agrégation : brutes ou élaborées

- Flux : logistique, monétaire, de personnel, de l'actif

- Utilisation : planification stratégique, gestion administrative, régulation opérationnelle

- Nature du support : oral, documentaire, informatique

Analyse systémique

Envisage les éléments d'une conformation complexe, les faits, non pas isolément mais globalement, en tant que parties intégrante d'un ensemble dont les différents composants sont dans une relation de dépendance réciproque [P.L.I. 1994]

Neuf niveaux (imbriqués) de complexité selon cette théorie

1. l'objet passif
 2. l'objet actif
 3. l'objet actif régulé
 4. l'objet s'informe
 5. l'objet décide son activité
 6. l'objet actif a une mémoire
 7. l'objet actif se coordonne
 8. l'objet actif imagine (et donc s'auto-organise)
 9. l'objet actif s'auto-finalise
- L'organisation correspond au dernier niveau

Définition

Ensemble d'éléments en interaction dynamique,
dont les éléments sont organisés et coordonnés en vue d'atteindre un objectif,
qui évolue dans un environnement

Description

Éléments

Attributs des éléments

Relations entre éléments

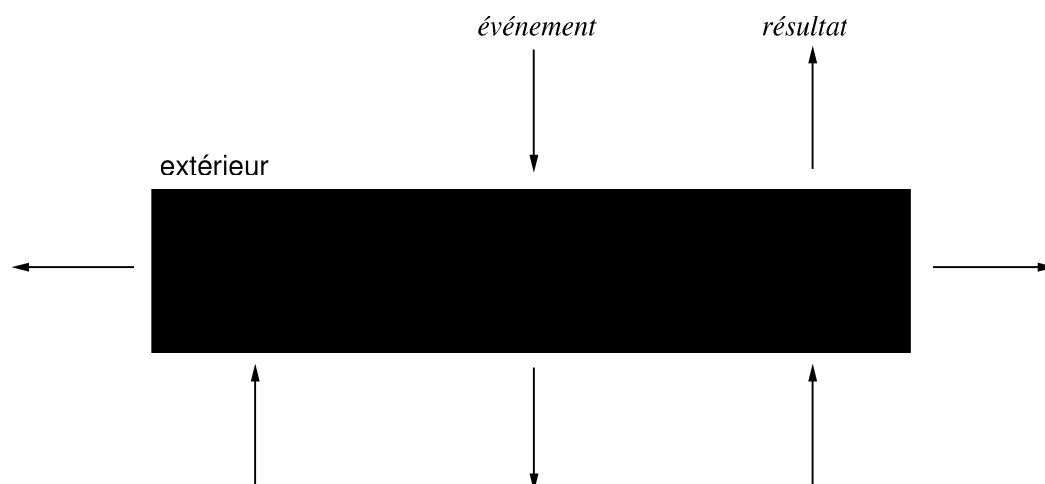
Identité

Objectif

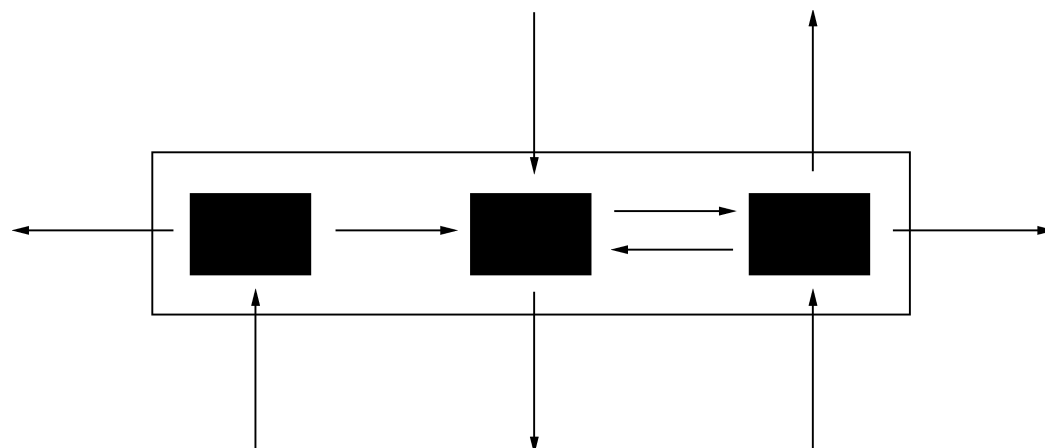
Contrôle

Environnement

Le système vu comme une « boîte noire »



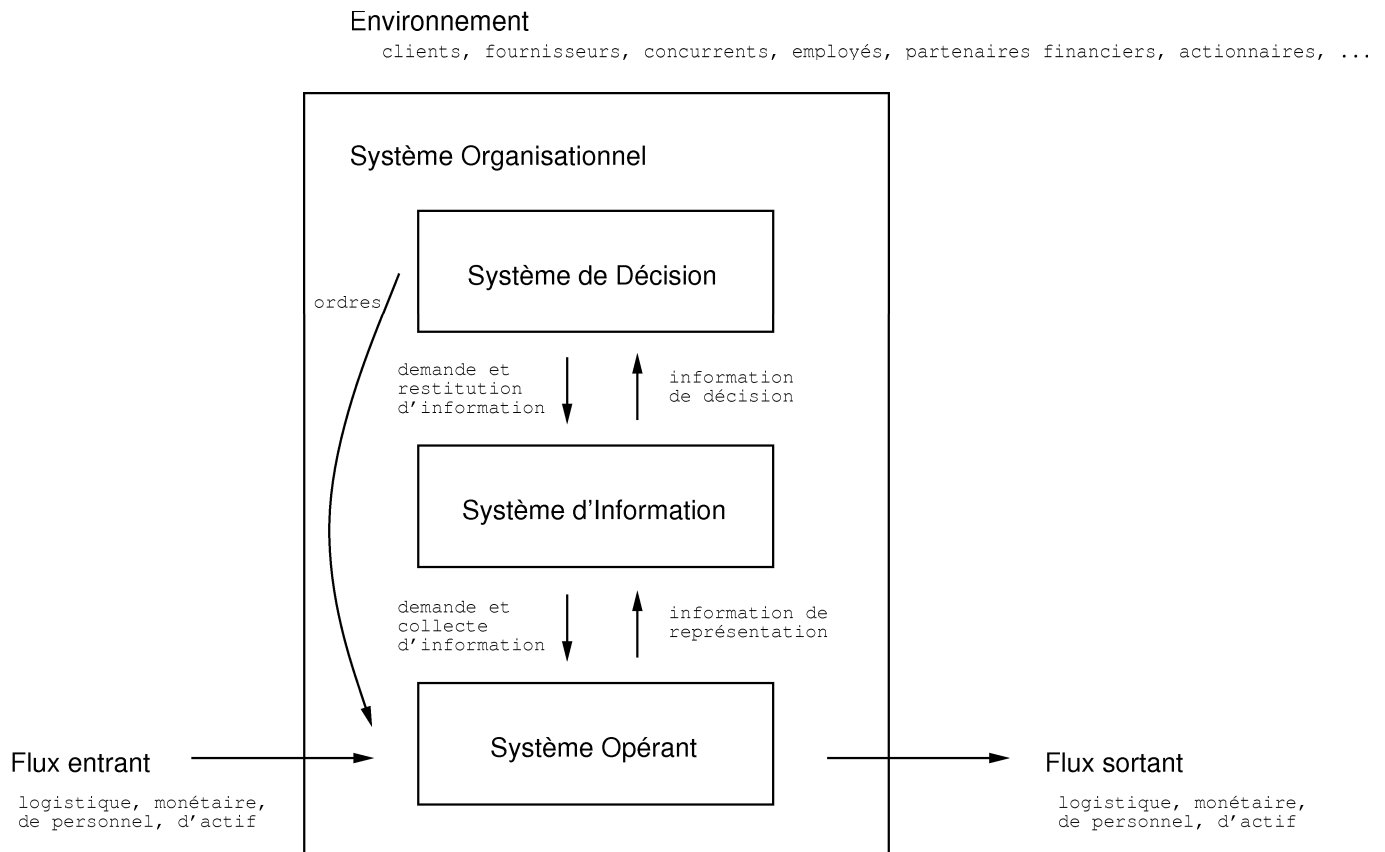
Le système se décompose en sous-systèmes dont on définit les entrées (issues de l'extérieur ou sorties d'autres sous-systèmes) et les sorties (à destination de l'extérieur ou devenant les entrées d'autres sous-systèmes)



Chaque sous-système est lui-même un système :
affinages successifs jusqu'à l'obtention d'une « boîte blanche »

Principale difficulté

Identifier le système, ses limites, ses sous-systèmes, le risque de perte engendrée par la décomposition, etc.



Système de Décision (ou pilotage, management, etc.)

Guide l'organisation vers ses objectifs (activités de planification et de contrôle)

Gérer

Système Opérant (ou logistique, technologique, physique, de production, etc.)

Effectue la transformation

Acheter - Produire - Stocker - Vendre

Système d'Information

Intermédiaire entre les systèmes de décision et opérationnel, par qui transite toute information

Remarque : un même employé peut être un acteur de chacun des trois sous-systèmes

Le système d'information d'une organisation est identifiable comme :

un ensemble de variables (élémentaires ou composées),
muni de règles (pour la saisie, la circulation, le traitement, la mémorisation des informations),
matérialisées par un support (papier ou informatique)

Rôles

Produire les informations légales réclamées par l'environnement

Déclencher les décisions programmées

Fournir des informations aux décideurs pour aider à la prise de décisions non programmées

Coordonner les tâches en assurant les communications au sein du système organisationnel

Fonctions

Circulation de l'information (accès à la mémoire, échange entre acteurs)

Traitement de l'information (rapprochements, calculs, comparaisons)

Mémorisation de l'information (conservation de l'information pour des besoins ultérieurs)

[Collecte et saisie de l'information]

Du système d'information au système informatique



Remarque : le système d'information informatisé fait partie du système organisationnel informatisé

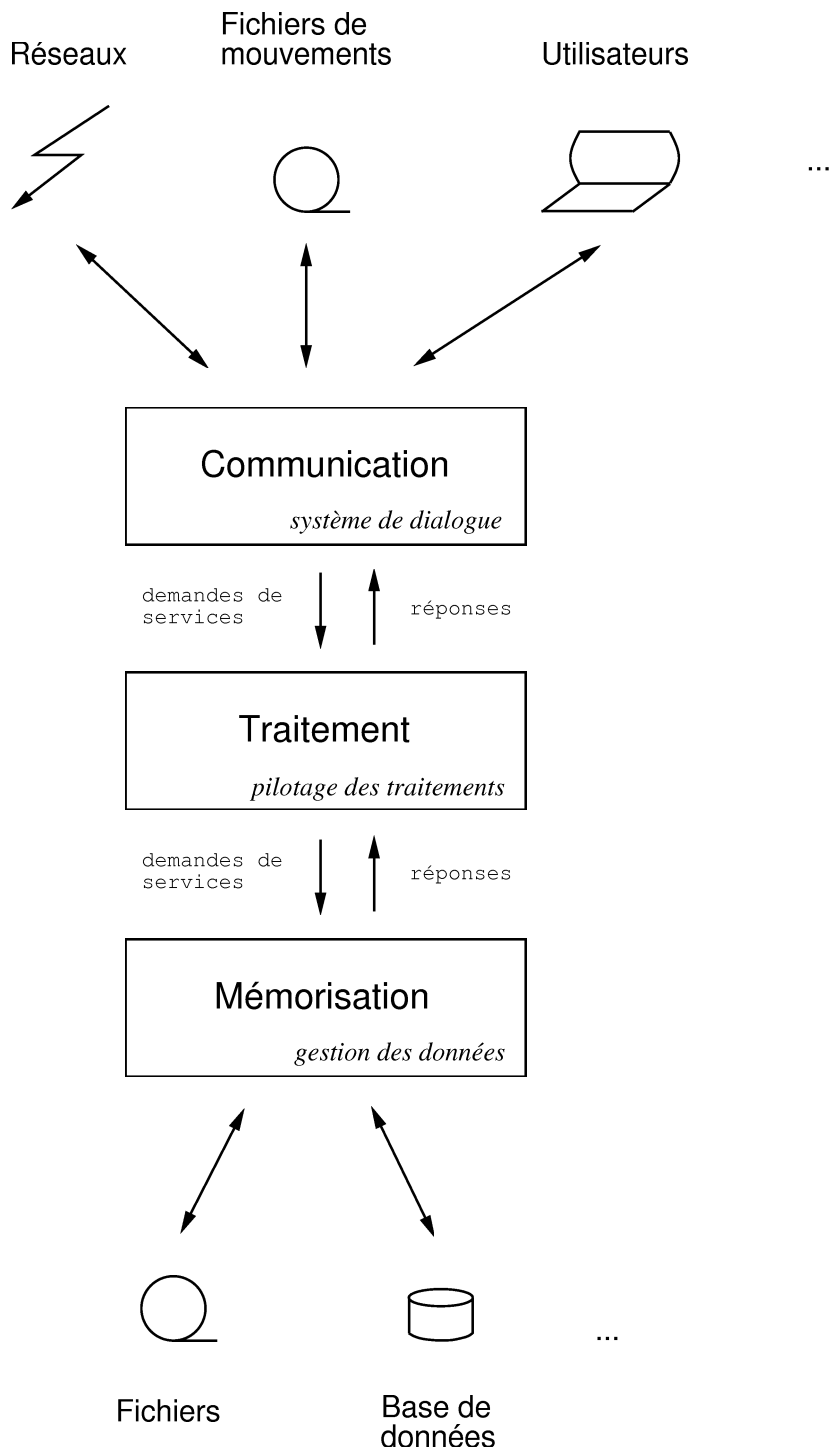
Connaissances nécessaires

Science de gestion

Mise en place du réseau d'information et de communication (conception du système d'information)

Technique informatique

Conception et réalisation du système informatique pour gérer le système d'information (conception du logiciel)



Enjeux pour l'organisation

- Augmenter la productivité

- Améliorer les conditions de travail

- Rendre un meilleur service (de qualité, rapide, etc.) aux partenaires de l'organisation

Critères d'un bon système informatique

- Productivité (en rationalisant le processus d'informatisation)

 - Établissement d'une ligne directrice des informatisations

 - Planification et suivi des performances

 - Efficacité des études informatiques

 - Utilisation judicieuse des technologies

- Qualité

 - Conformité de la réalisation par rapport aux besoins

 - Documentation correcte

 - Adaptabilité

 - Fiabilité

 - Facilité d'utilisation

- Rentabilité

 - Gain pour l'organisation relativement au coût de l'informatisation

Facteurs de la complexité de l'informatisation

- Difficultés techniques de l'informatique

- Constantes novations (matérielle et logicielle)

- Symbiose requise entre l'application informatique et toute l'organisation (et ses partenaires)

- Multiplicité des décisions et nombreux domaines (humain, financier, technique, etc.) de l'organisation concernés

L'informatique remplit maintenant un rôle stratégique dans l'organisation

(automatisation des tâches administratives → système d'information d'aide à la décision)

Le service informatique

Direction informatique

- Responsable du service informatique
- Chef d'un département du service informatique

Expertise

- Administrateur du système (d'exploitation)
- Administrateur du réseau
- Administrateur de la base de données
- Expert en méthodes
- Expert en qualité
- Expert en sécurité
- Expert en technologies diverses

Études - Développement

- Chef de projet
- Analyste
- Concepteur (ou ingénieur d'étude)
- Analyste - Programmeur
- Développeur (ou programmeur ou réalisateur d'application)

Production - Exploitation

- Opérateur - Pupitreux
- Analyste d'exploitation
- Contrôleur réseau
- Technicien (micro-informatique, réseau, messagerie, téléphonie)

Support et assistance

- Assistant technique clientèle

Autre métier : Consultant en systèmes d'information

Quelques chiffres concernant les anciens diplômés du Département Informatique de l'Institut Universitaire de Technologie 'A' de l'Université Bordeaux I (sur la base de 530 réponses reçues au 18/12/1996)

- 54 % en études et développement
- 16 % à la direction informatique
- 10 % d'experts
- 12 % d'autres fonctions d'informaticien
- 8 % n'exercent pas la fonction d'informaticien

Différents types d'organisation du département Études - Développement

Horizontale : découpage fonctionnel des activités

- Un responsable pour chaque phase du cycle de vie de tous les projets
- Seul le chef du département a la responsabilité globale de tous les projets

Verticale : découpage par projets

- Constitution d'une équipe pour toute la durée de vie d'un projet
- Projet sous l'autorité d'un responsable direct et unique

Matricielle : double découpage horizontal et vertical

- Chaque membre du département est sous l'autorité d'un responsable technique et d'un chef de projet

Les rôles des différents intervenants liés à l'informatisation

- Directeur de l'organisation : décide du lancement du projet
- Manager : responsable de l'avancement et de l'aboutissement du projet
- Coordinateur du développement :
assure la liaison entre les différents groupes du projet, rend compte au manager
- Gestionnaire de ressources : met à disposition les ressources nécessaires au projet
- Analyste de l'organisation : analyse l'existant
- Concepteur : conçoit la future application
- Réalisateur : programme
- Réalisateur-valideur : contrôle le travail du réalisateur
- Utilisateur : utilise le logiciel une fois installé
- Utilisateur-valideur : « recette » le logiciel

L'équipe d'analyse

Les informaticiens

- Responsable du service informatique
- Chef de projet
- Analyste
- Développeur
- Personnel de l'exploitation
- Sous-traitant de l'application

Les utilisateurs

- Direction générale
- Responsable du service des utilisateurs
- Personnel
- Autres services
- Clients

La démarche de la méthode d'informatisation traditionnelle

Étude préalable

- Analyse du fonctionnement de l'organisation et diagnostic général de l'existant
- Recensement des critiques et des besoins des utilisateurs
- Opportunité et faisabilité des automatisations
- Rédaction d'un cahier des charges

Analyse fonctionnelle

- Élaboration d'une solution conceptuelle, indépendante de tout moyen informatique
- Rédaction d'un dossier de conception (traitements généraux, nouvelles formes des informations, budgets)

Analyse organique

- Définition des structures de données et de leur enregistrement
- Détermination des unités de traitement
- Choix des matériels
- Établissement du planning et des budgets de réalisation

Programmation

- Réalisation effective du système informatique
- Tests unitaires et d'intégration

Mise en service

- Démarrage du système
- Formation des utilisateurs
- Exploitation et maintenance du système

Les différentes phases du cycle de vie (et de développement) du logiciel

Analyse

- Analyse de l'existant et définition des besoins, du système d'information et du logiciel

Conception

- Conception du système d'information et du logiciel

Réalisation (ou codage, programmation)

- Traduction des algorithmes dans un langage compréhensible par un ordinateur

Tests du logiciel et du système d'information

- Vérification et validation du logiciel
- Vérification et validation du système d'information

Exploitation

(hors développement)

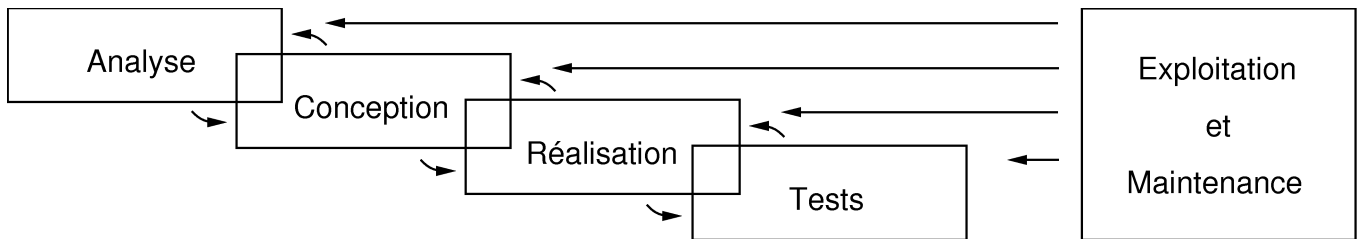
- Utilisation du logiciel une fois installé (et « recetté »)

Maintenance

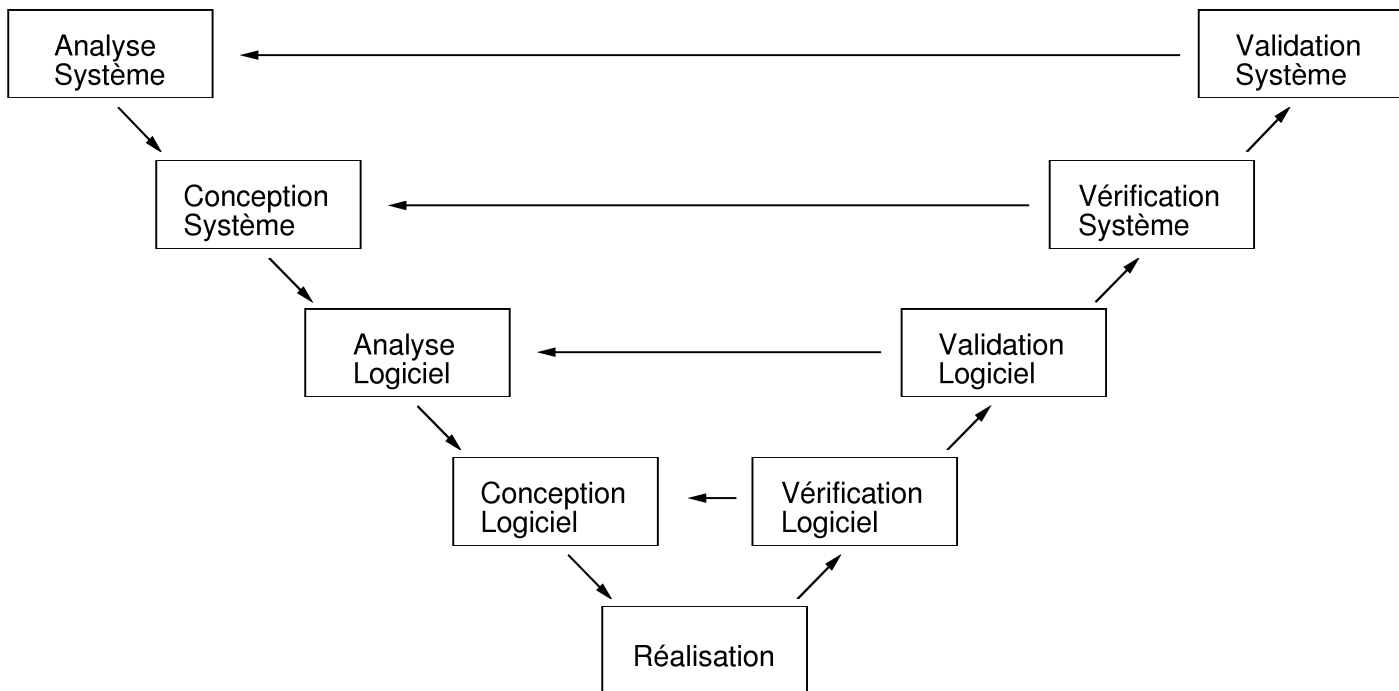
(hors développement)

- Correction des erreurs
- Amélioration des fonctions existantes
- Ajout de nouvelles fonctionnalités

Cycle de vie en cascade (ou en chute d'eau)



Cycle de développement en V



Système signifie ici système d'information (manuel et informatisé)

Vérification : le produit en cours d'élaboration répond à la définition des besoins (est-ce bien le produit ?)

Validation : le produit en cours d'élaboration remplit les fonctionnalités désirées par l'utilisateur (est-ce le bon produit ?)

C'est le modèle de l'Association Française pour le Contrôle Industriel de Qualité (AFCIQ) qui utilise les phases suivantes :

Spécification fonctionnelle

Conception préliminaire

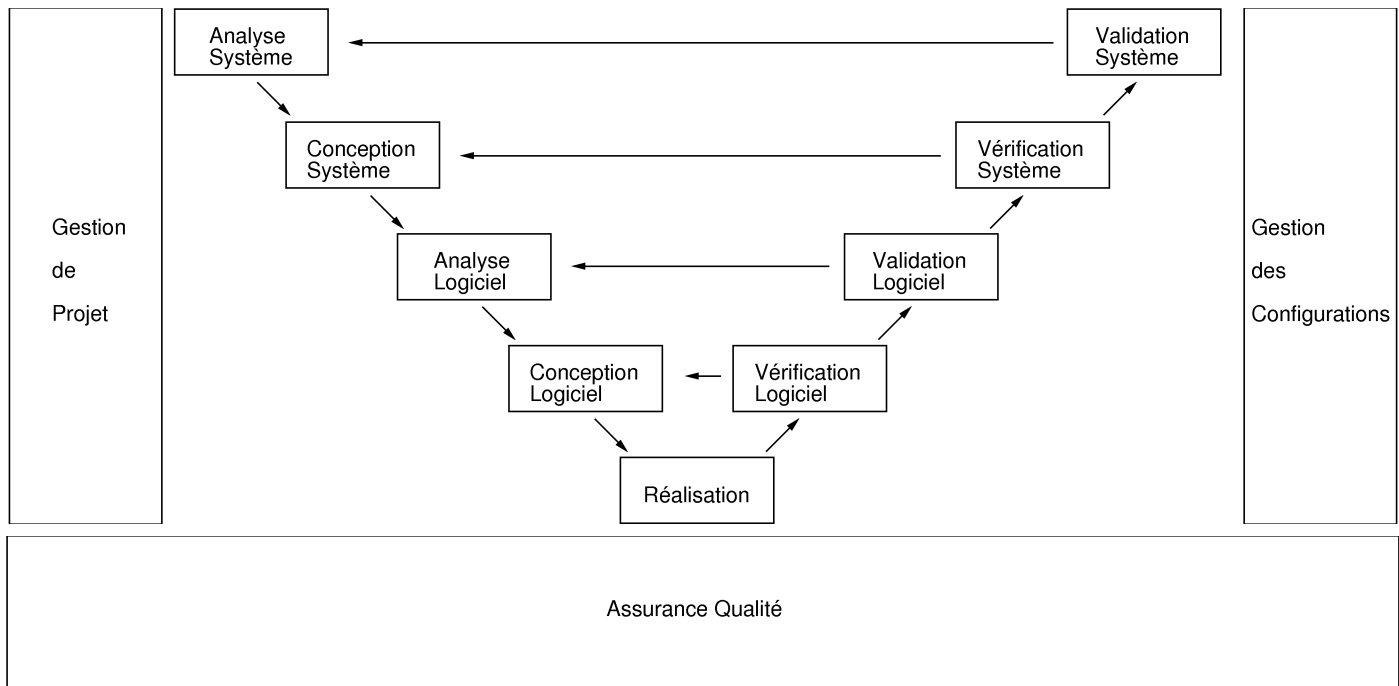
Conception détaillée

Codage

Tests d'intégration
Tests unitaires

Recette

Cycle de développement en M

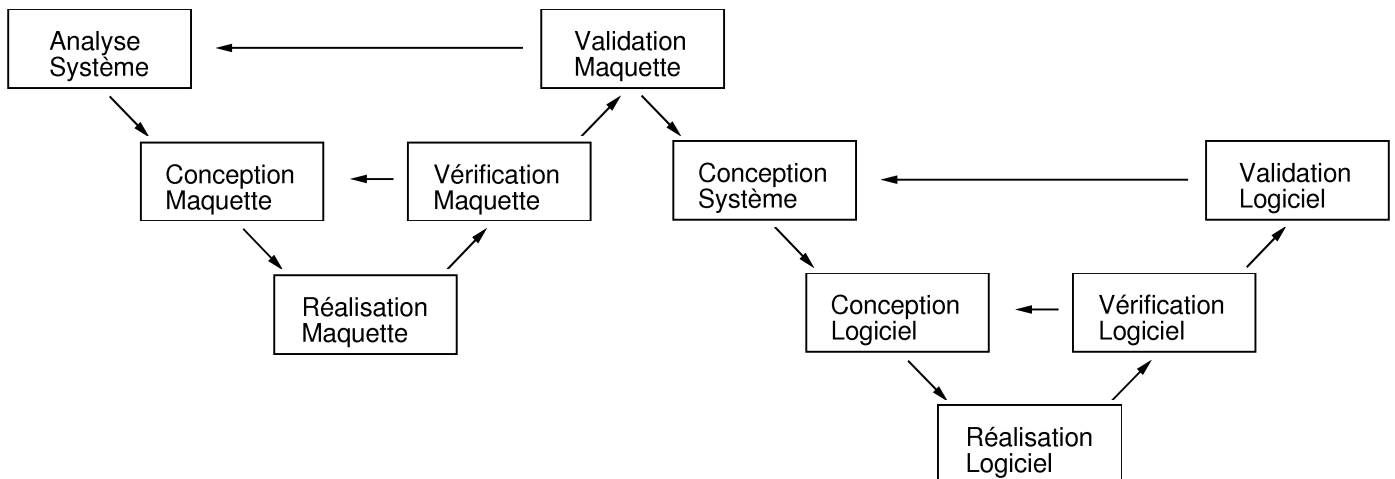


Gestion de projet : pilotage durant toute la durée de développement du produit

Gestion des configurations : gestion des différentes versions du produit

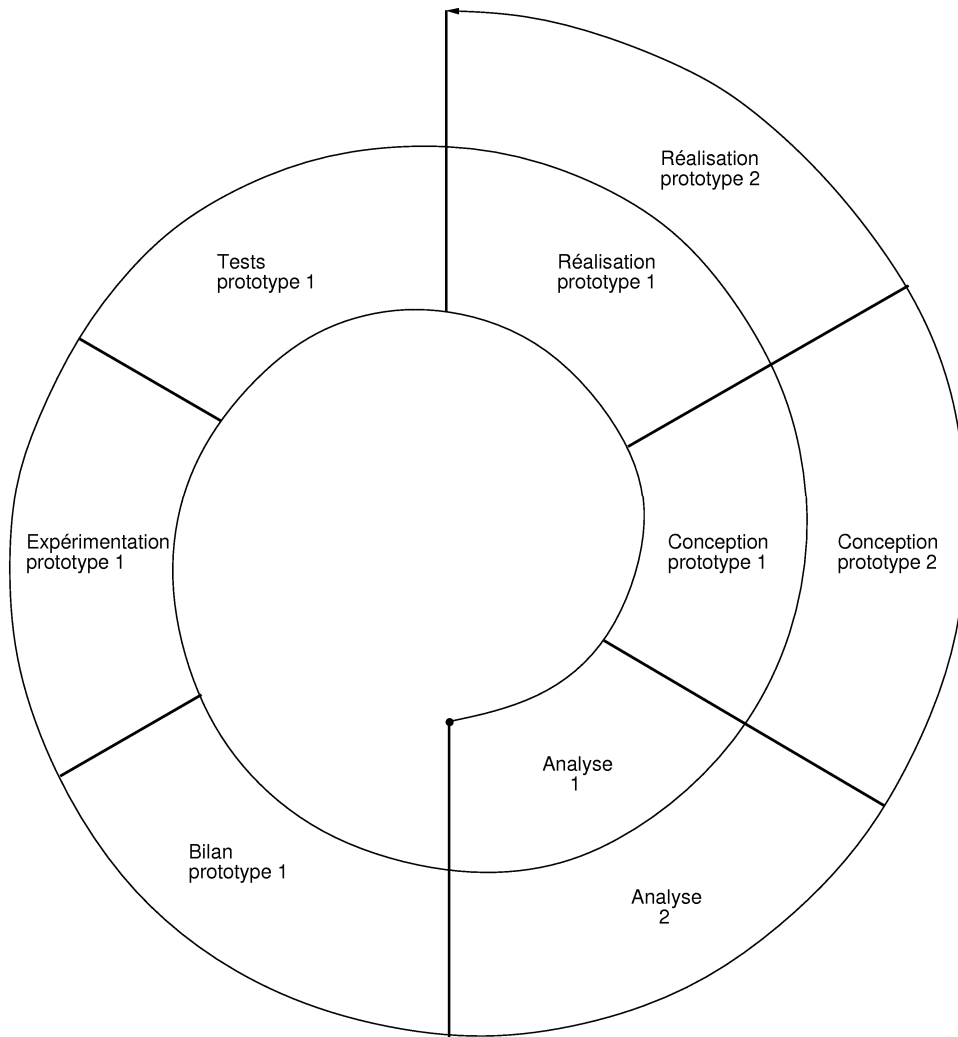
Assurance qualité : contrôle systématiquement que le produit en cours est cohérent et complet, en le confrontant à des normes préétablies si elles existent

Cycle de développement en W



Maquette : défilement d'écrans donnant une idée de ce que sera la future application (sans accès aux données)

Cycle de développement en spirale

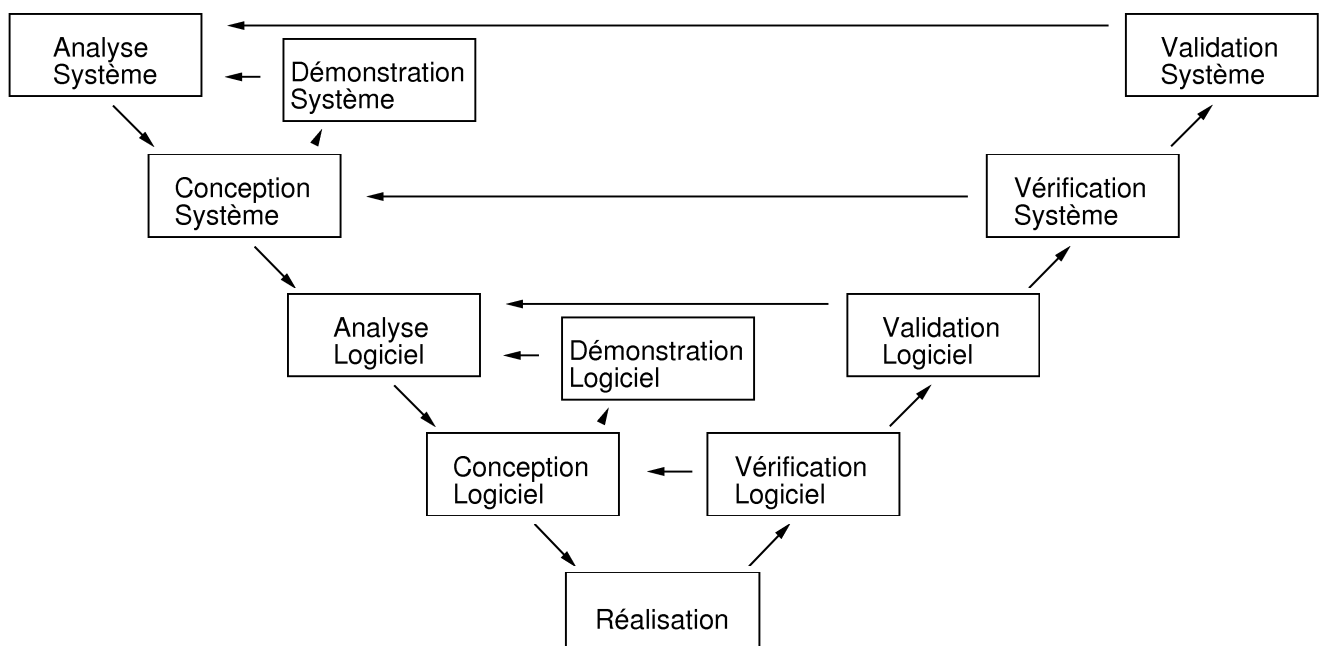


Prototype : application en réduction (avec accès aux données)

Expérimentation : tests de la part des utilisateurs du produit dans sa version actuelle (éventuellement définitive)

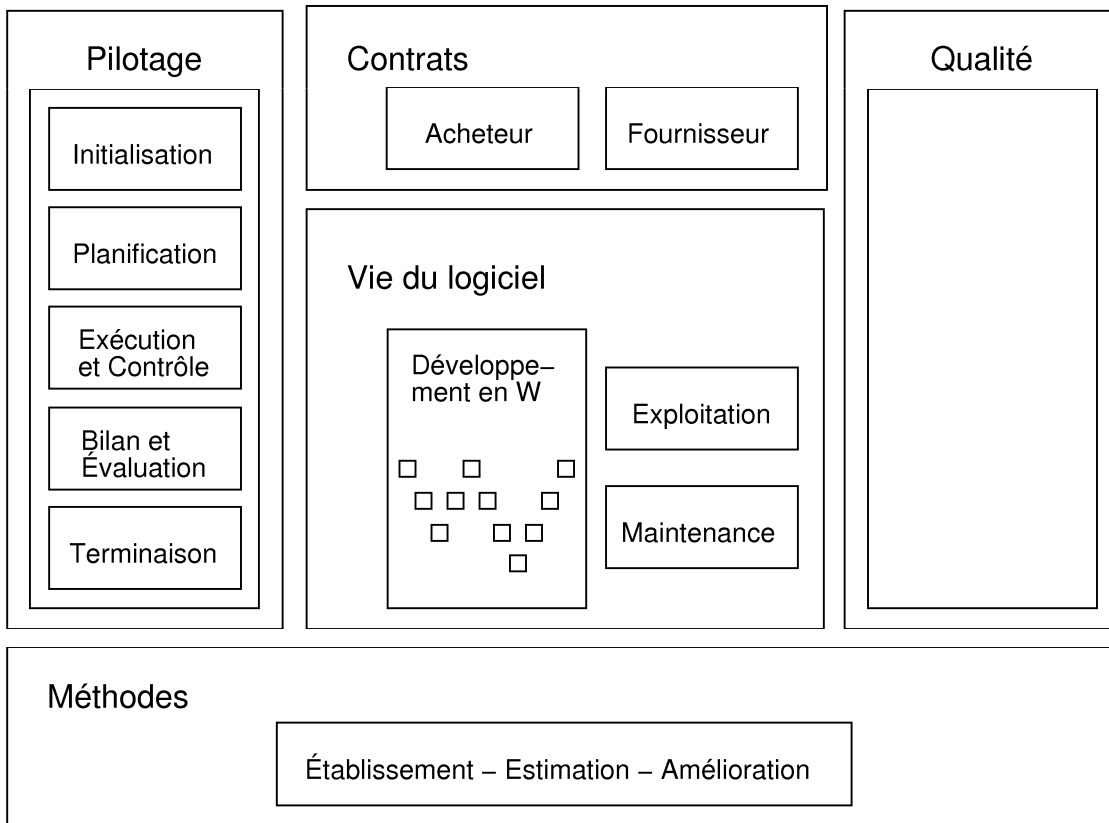
Bilan : critique de l'expérimentation

Exemple de cycle de développement composite

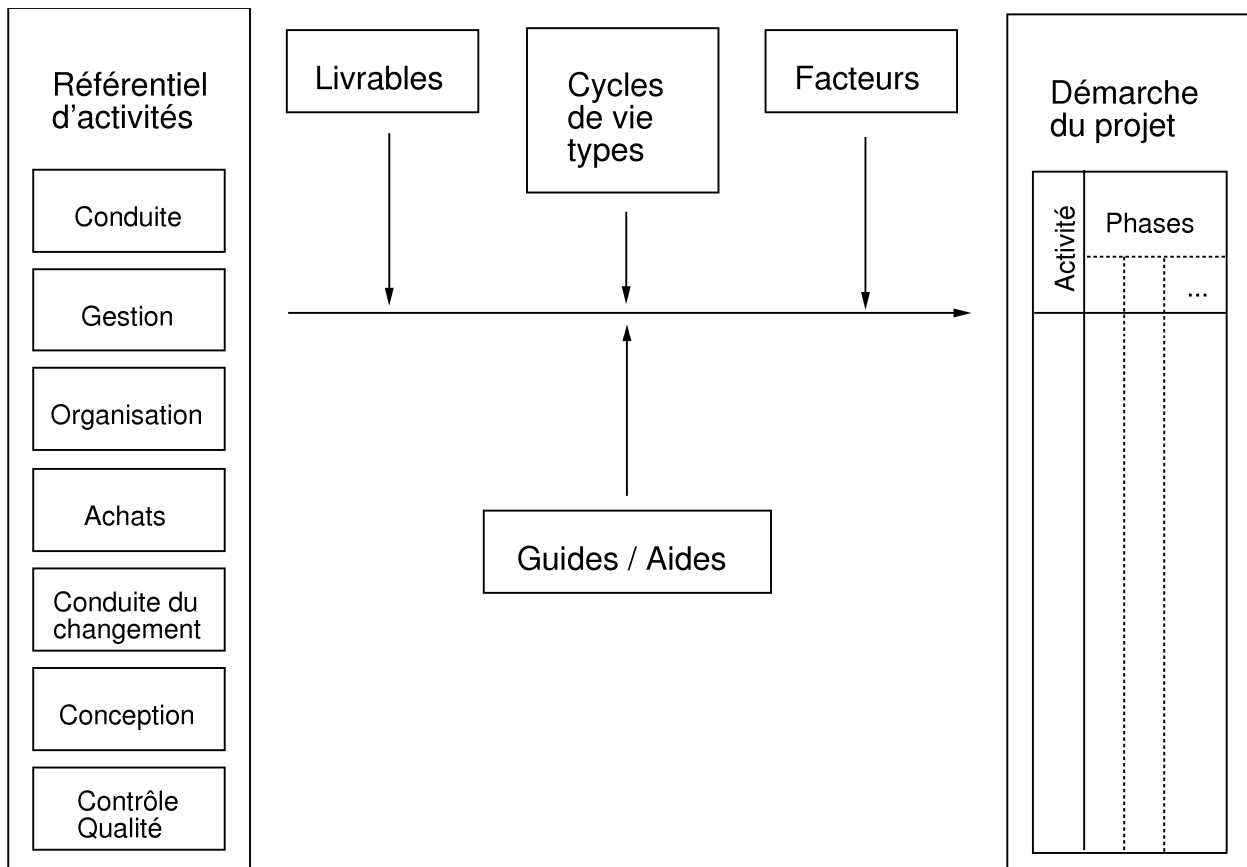


Démonstration : présentation du produit aux utilisateurs

Cycle de vie ISO (en cours de normalisation)



Cycle de vie EuroMethode (en cours de normalisation)



Coût moyen relatif de chaque phase (du cycle de développement du logiciel) pour une application de gestion

Analyse et Conception	:	44 %
Réalisation	:	28 %
Tests	:	28 %

Coût relatif de correction d'une erreur selon la phase (du cycle de vie du logiciel) au cours de laquelle elle a été détectée

Analyse	:	1
Conception	:	2
Réalisation	:	5
Tests	:	10
Exploitation et Maintenance	:	≥ 100

Remarque : plus de 80 % des erreurs sont introduites durant les phases d'analyse et conception

Les coûts de la maintenance corrective (ni adaptative, ni évolutive) peuvent aller jusqu'à 2 fois ceux du développement

Exemple pathologique (système avionique) :

coût de développement	:	30 \$ par instruction
coût de maintenance	:	4000 \$ par instruction

Productivité moyenne d'un programmeur d'une application de gestion simple :

moins de 600 lignes de code par mois

Application moyenne (en 1985)

100 000 lignes de code pour 4 000 000 de francs

Exemples : suivi de production pour 3000 personnes, entreprise commerciale de 2 milliards de chiffre d'affaires

Définition

Une méthode d'informatisation en informatique de gestion définit un processus d'informatisation du système d'information (partiellement ou totalement), possède une portée (champ d'étude), et décrit un ensemble de travaux en les ordonnant (démarche)

Règles

S'appuyer sur des concepts théoriques

Proposer une démarche

Permettre sa mise en œuvre par des outils

Atteindre un but

Remarque : une méthode ne remplace ni l'expérience, ni la connaissance, ni le talent

Composants

Modèles

Ensemble de concepts et de règles destiné à expliquer et construire la représentation de phénomènes organisationnels

Langages

Destinés à l'élaboration des spécifications, à faciliter la communication

Démarche

Processus pour effectuer les travaux préconisés, découpée en étapes

Outils et techniques

Aides à la mise en œuvre des modèles, langages, démarche

Objectifs

Réduire la complexité des informatisations

Rendre cohérents tous les projets

Capitaliser les expériences

Augmenter la qualité des travaux d'informatisation

Augmenter la productivité des travaux d'informatisation

Améliorer les communications entre intervenants (utilisateurs et informaticiens)

Les solutions empiriques

Avantage : répondent à l'urgence

Inconvénient : génèrent des applications provisoires (car complexes, non fiables, coûteuses, etc.)

Fondements théoriques (différents types d'approche des problèmes d'informatisation)

Cartésienne (démarche dite analytique)

Approche fonctionnelle (analyse et conception des systèmes d'information par rapport à la définition des besoins) et descendante (du général au particulier)

Exemples : SADT, CORIG

Systémique (démarche dite globalisante)

Approche conceptuelle (processus de modélisation par niveaux d'abstraction successifs)

Repose sur l'identification de projets qui structurent l'organisation (sans qu'il y ait obligatoirement un besoin)

Exemples : MERISE, AXIAL, IA-NIAM

À objet (application du paradigme objet à tout le processus)

Les objets (de l'application, de services distribués) et les utilitaires communs échangent des informations (demandes et réponses de services) à l'aide de messages

Exemples : OOA, OMT, MCO

Générations

Première génération

Des années 60 au début des années 70

Automatisation des procédures administratives

Problèmes de programmation

Approche analytique (par les données) ou synthétique (par les fonctions)

Exemple : CORIG

Deuxième génération

Années 70

Généralisation des champs d'étude au système d'information et à l'organisation en entier

Préconisations (en analyse, conception, programmation) et démarche d'informatisation (schéma directeur, plan d'informatisation, conduite de projet)

Prise en compte de nouvelles techniques (temps réel, bases de données, ergonomie), nouvelles formalisations (entités-associations), évolution des sciences de gestion

Exemples : IA-NIAM, SADT

Troisième génération

Depuis la fin des années 70

Informatisation globale (cohérence, complétude)

Innovations technologiques (matérielles et logicielles)

Démarche de synthèse, davantage de modélisation, introduction d'outils logiciels associés

Inclut la famille des méthodes à objets (années 80)

Exemples : MERISE, AXIAL, SSADM, OOA

Quatrième génération ?

Intégration des technologies orientées objets, client/serveur

Domaines d'application

Particulier

Application à un travail précis et indépendant de toute démarche

Exemple : RACINES pour l'élaboration d'un schéma directeur

Partiel

Description et ordonnancement de travaux relativement à une démarche d'informatisation partielle

Exemples : CORIG pour la conception et la réalisation du système informatique, SADT et IA-NIAM pour la conception du système d'information et du système informatique

Global

Processus d'informatisation complet (de l'introduction de l'informatique dans une organisation à la maintenance des applications) : description et ordonnancement de tous les travaux

Exemples : MERISE, AXIAL, SSADM, OOA

Démarche

Linéaire

Succession linéaire des travaux (démarche découpée en étapes découpées en phases découpées en tâches découpées en opérations)

Analyse descendante (des problèmes généraux aux problèmes particuliers) par décomposition hiérarchique des travaux

Itération et condition possibles

Exemples : MERISE, AXIAL

Non linéaire

Analyse ascendante par intégration progressive des résultats

Approche

Ascendante

Recensement et analyse des sorties (papier ou écran) puis établissement des entrées nécessaires et suffisantes

La liste des informations obtenue est ainsi l'ensemble minimal nécessaire pour obtenir les résultats, ce qui permet difficilement de prendre en compte l'évolution des besoins de l'organisation

Exemple : MINOS

Descendante

Recensement des informations du système d'information existant (sans oubli ni répétition) et des nouvelles fonctionnalités des utilisateurs

Exemples : CORIG, MERISE, etc.

MERISE (et MERISE/2)

Méthode d'étude et de réalisation informatique pour les systèmes d'entreprise

H. TARDIEU pour Séma-Matra et Gamma International, France, 1983

Informatisation complète

AXIAL

Analyse et conception de systèmes d'information assistés par logiciels

IBM, France, 1986

Informatisation complète

CORIG

Compagnie Générale d'Informatique, France, 1966

Conception et réalisation du système informatique

SSADM

Structured systems analysis and design method

LBMS pour CCTA, Grande-Bretagne, 1986

Informatisation complète

IA-NIAM

Nijssen's information analysis method

M. NIJSSEN pour Control Data, Belgique, 1982

Conception du système d'information et du système informatique

SADT

Structured analysis and design techniques

D.T. ROSS pour SofTech, USA, 1976 (et IGL Technology, France, 1977)

Conception du système d'information et du système informatique

OOA

Object-oriented analysis

P. COAD et E. YOURDON, 1991

MCX (et MCO)

Méthode générale d'analyse des applications informatiques

X. CASTELLANI, France, 1983

Informatisation complète

SDM/S

System development methodology

AGS Management Systems, USA

MCP

Méthode de conduite de projets informatiques

RATP et AFCET, France, 1978

Conduite de projets

JSD

Jackson system development

M. JACKSON, Systems Ltd, Royaume-Uni, 1983

Conception du système d'information et du système informatique

REMORA

C. ROLLAND de l'Université Paris 1 (Sorbonne), France, 1988

Conception du système d'information et du système informatique

OOD (et HOOD)

(Hierarchical) object oriented design

R. ABBOTT en 1980, G. BOOCH en 1983 (CISI & MATRA & CRI pour l'Agence Spatiale Européenne en 1987)

Conception et réalisation du système informatique

OMT

Object modeling technique

J. RUMBAUGHT

Conception du système d'information et du système informatique

Z

D. LIGHTFOOT en 1994

Conception du système d'information et du système informatique

<i>Numéro INE de l'étudiant : 2</i> <i>Nom : LEROI</i> <i>Département de naissance : 40</i>		<i>Voitures possédées par l'étudiant :</i>										
		<table border="1"> <tr> <th>Numéro d'immatriculation</th> <th>Couleur</th> </tr> <tr> <td></td> <td></td> </tr> </table>	Numéro d'immatriculation	Couleur								
Numéro d'immatriculation	Couleur											
<i>Diplômes obtenus par l'étudiant :</i>												
<table border="1"> <tr> <th>Intitulé abrégé</th> <th>Intitulé complet</th> <th>Année</th> </tr> <tr> <td>BAC</td> <td>Baccalauréat</td> <td>1980</td> </tr> <tr> <td>DEUG</td> <td>Diplôme d'Études Universitaires Générales</td> <td>1982</td> </tr> </table>	Intitulé abrégé	Intitulé complet	Année	BAC	Baccalauréat	1980	DEUG	Diplôme d'Études Universitaires Générales	1982			
Intitulé abrégé	Intitulé complet	Année										
BAC	Baccalauréat	1980										
DEUG	Diplôme d'Études Universitaires Générales	1982										

<i>Numéro INE de l'étudiant : 3</i> <i>Nom : DUPOND</i> <i>Département de naissance : 17</i>		<i>Voitures possédées par l'étudiant :</i>													
		<table border="1"> <tr> <th>Numéro d'immatriculation</th> <th>Couleur</th> </tr> <tr> <td></td> <td></td> </tr> </table>	Numéro d'immatriculation	Couleur											
Numéro d'immatriculation	Couleur														
<i>Diplômes obtenus par l'étudiant :</i>															
<table border="1"> <tr> <th>Intitulé abrégé</th> <th>Intitulé complet</th> <th>Année</th> </tr> <tr> <td>BAC</td> <td>Baccalauréat</td> <td>1981</td> </tr> <tr> <td>DUT</td> <td>Diplôme Universitaire de Technologie</td> <td>1983</td> </tr> <tr> <td>MIAGe</td> <td>Maîtrise des Méthodes Informatiques Appliquées à la Gestion des Entreprises</td> <td>1985</td> </tr> </table>	Intitulé abrégé	Intitulé complet	Année	BAC	Baccalauréat	1981	DUT	Diplôme Universitaire de Technologie	1983	MIAGe	Maîtrise des Méthodes Informatiques Appliquées à la Gestion des Entreprises	1985			
Intitulé abrégé	Intitulé complet	Année													
BAC	Baccalauréat	1981													
DUT	Diplôme Universitaire de Technologie	1983													
MIAGe	Maîtrise des Méthodes Informatiques Appliquées à la Gestion des Entreprises	1985													

<i>Numéro INE de l'étudiant : 4</i> <i>Nom : MARTIN</i> <i>Département de naissance : 47</i>		<i>Voitures possédées par l'étudiant :</i>													
		<table border="1"> <tr> <th>Numéro d'immatriculation</th> <th>Couleur</th> </tr> <tr> <td>4747 LA 47</td> <td>rouge</td> </tr> </table>	Numéro d'immatriculation	Couleur	4747 LA 47	rouge									
Numéro d'immatriculation	Couleur														
4747 LA 47	rouge														
<i>Diplômes obtenus par l'étudiant :</i>															
<table border="1"> <tr> <th>Intitulé abrégé</th> <th>Intitulé complet</th> <th>Année</th> </tr> <tr> <td>BAC</td> <td>Baccalauréat</td> <td>1977</td> </tr> <tr> <td>DEUG</td> <td>Diplôme d'Études Universitaires Générales</td> <td>1980</td> </tr> <tr> <td>MIAGe</td> <td>Maîtrise des Méthodes Informatiques Appliquées à la Gestion des Entreprises</td> <td>1982</td> </tr> </table>	Intitulé abrégé	Intitulé complet	Année	BAC	Baccalauréat	1977	DEUG	Diplôme d'Études Universitaires Générales	1980	MIAGe	Maîtrise des Méthodes Informatiques Appliquées à la Gestion des Entreprises	1982			
Intitulé abrégé	Intitulé complet	Année													
BAC	Baccalauréat	1977													
DEUG	Diplôme d'Études Universitaires Générales	1980													
MIAGe	Maîtrise des Méthodes Informatiques Appliquées à la Gestion des Entreprises	1982													

<i>Numéro INE de l'étudiant : 5</i> <i>Nom : DURAND</i> <i>Département de naissance : 33</i>		<i>Voitures possédées par l'étudiant :</i>										
		<table border="1"> <tr> <th>Numéro d'immatriculation</th> <th>Couleur</th> </tr> <tr> <td>3333 BX 33</td> <td>rouge</td> </tr> <tr> <td>4040 NT 40</td> <td>jaune</td> </tr> </table>	Numéro d'immatriculation	Couleur	3333 BX 33	rouge	4040 NT 40	jaune				
Numéro d'immatriculation	Couleur											
3333 BX 33	rouge											
4040 NT 40	jaune											
<i>Diplômes obtenus par l'étudiant :</i>												
<table border="1"> <tr> <th>Intitulé abrégé</th> <th>Intitulé complet</th> <th>Année</th> </tr> <tr> <td>BAC</td> <td>Baccalauréat</td> <td>1981</td> </tr> <tr> <td>DUT</td> <td>Diplôme Universitaire de Technologie</td> <td>1983</td> </tr> </table>	Intitulé abrégé	Intitulé complet	Année	BAC	Baccalauréat	1981	DUT	Diplôme Universitaire de Technologie	1983			
Intitulé abrégé	Intitulé complet	Année										
BAC	Baccalauréat	1981										
DUT	Diplôme Universitaire de Technologie	1983										

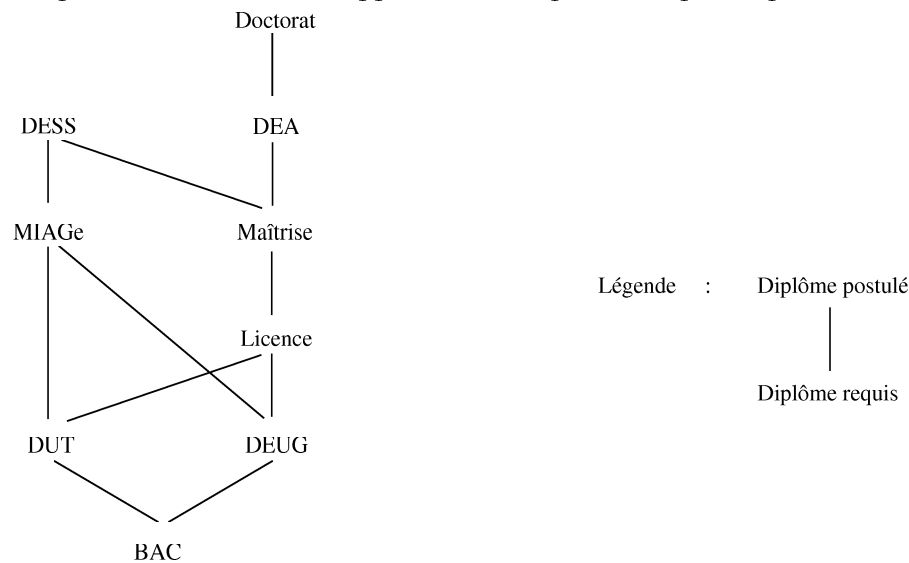
<i>Numéro INE de l'étudiant : 7</i> <i>Nom : LEROI</i> <i>Département de naissance : 33</i>		<i>Voitures possédées par l'étudiant :</i>							
		<table border="1"> <tr> <th>Numéro d'immatriculation</th> <th>Couleur</th> </tr> <tr> <td></td> <td></td> </tr> </table>	Numéro d'immatriculation	Couleur					
Numéro d'immatriculation	Couleur								
<i>Diplômes obtenus par l'étudiant :</i>									
<table border="1"> <tr> <th>Intitulé abrégé</th> <th>Intitulé complet</th> <th>Année</th> </tr> <tr> <td></td> <td></td> <td></td> </tr> </table>	Intitulé abrégé	Intitulé complet	Année						
Intitulé abrégé	Intitulé complet	Année							

Le jour de la rentrée, le secrétariat de l'établissement avise les étudiants qu'ils ont jusqu'à la fin de la semaine pour amener les originaux des diplômes qu'ils ont obtenus, ceci permettant de compléter les fiches des étudiants.

Conception (données)

On considère les informations suivantes destinées à compléter l'exemple « jouet »

L'étudiant PICARD a acheté pour 28000 francs une Twingo immatriculée 1717 LR 17 le 27 juin 1989
 On désire savoir quels sont les étudiants qui peuvent conduire les voitures des étudiants
 L'information sur les diplômes est insuffisante : il faut également considérer les établissements (avec notamment leurs adresses) qui ont délivré ces diplômes aux étudiants
 On veut également pouvoir considérer les adresses des logements des étudiants. Toutefois, tous les étudiants n'ont pas forcément un tel logement et certains habitent chez leurs parents. Par contre, on désire connaître l'adresse des parents de tous les étudiants
 On se propose de constituer une classification de poursuites d'études ; pour cela, nous disposons de l'esquisse suivante faisant apparaître les diplômes requis et postulés



Exercice 1 : déterminer les nouvelles dépendances fonctionnelles, construire le nouveau hyper-graphe des dépendances fonctionnelles, et en déduire le nouveau schéma relationnel

Exercice 2 : établir le nouveau schéma entité-association

Exercice 3

Passer du nouveau schéma relationnel issu de l'exercice 1 au schéma entité-association ; comparer ce dernier à celui issu de l'exercice 2

Passer du nouveau schéma entité-association issu de l'exercice 2 au schéma relationnel ; comparer ce dernier à celui issu de l'exercice 1

Algèbre relationnelle et SQL (traitements)

Exercice 4 : écrire les algorithmes de toutes les opérations de l'algèbre relationnelle

Exercice 5 : écrire en algèbre relationnelle et en SQL les requête suivantes

Les libellés complets des diplômes

Les voitures immatriculées en Gironde (33) et à Paris (75)

Les étudiants nés en Gironde et dont le numéro est au plus égal à 9

Les années d'obtention du DUT qui ne soient pas des années d'obtention du BAC

Les étudiants et leurs diplômes

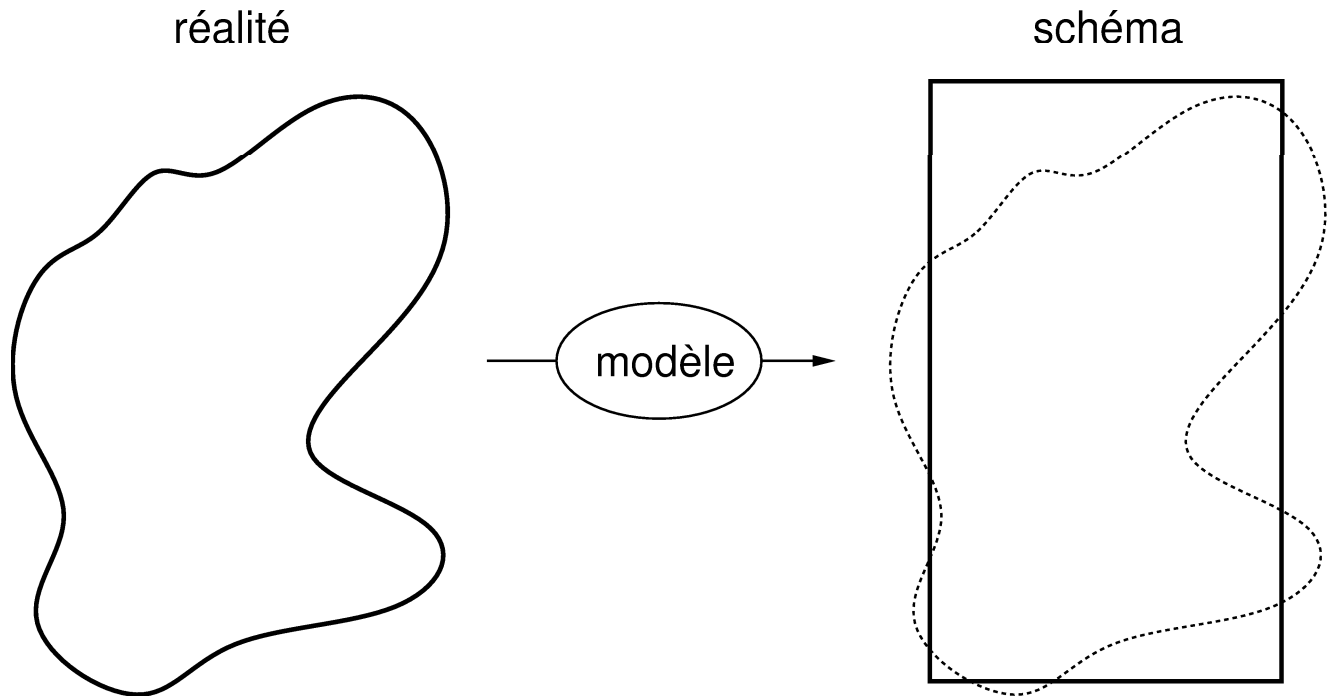
Les étudiants sans diplôme

Tous les diplômes auxquels peuvent prétendre chaque étudiant

Les étudiants ayant obtenu tous les diplômes

Pour chaque étudiant, les diplômes qu'il n'a pas obtenus

La relation universelle



Objectifs

Rendre compte de la réalité

Conforme - Complet - Réalisable - Plausible

Communiquer

Standard - Précis - Simple - Cohérent

Outils

Langage naturel

Représentation graphique

Mathématiques

Le système d'information vu à travers des modèles :

de communication :

organigramme, fiche de description de fonction, modèle fonctionnel, arbre de décomposition fonctionnel, diagramme de flots de données, graphe acteurs-flux, modèle de flux, etc.

des traitements :

réseaux de Pétri, diagramme de circulation de l'information, modèle conceptuel des traitements (analytique), modèle organisationnel des traitements (analytique), modèle logique des traitements (répartis), schéma d'architecture logique des moyens informatiques, actigramme, modèle dynamique, diagramme de suivi d'événements, diagramme de flots d'événements entre objets, diagramme de flots d'événements entre classes, diagramme d'états (structuré), cycle de vie d'un objet, diagramme de structure, table de décision etc.

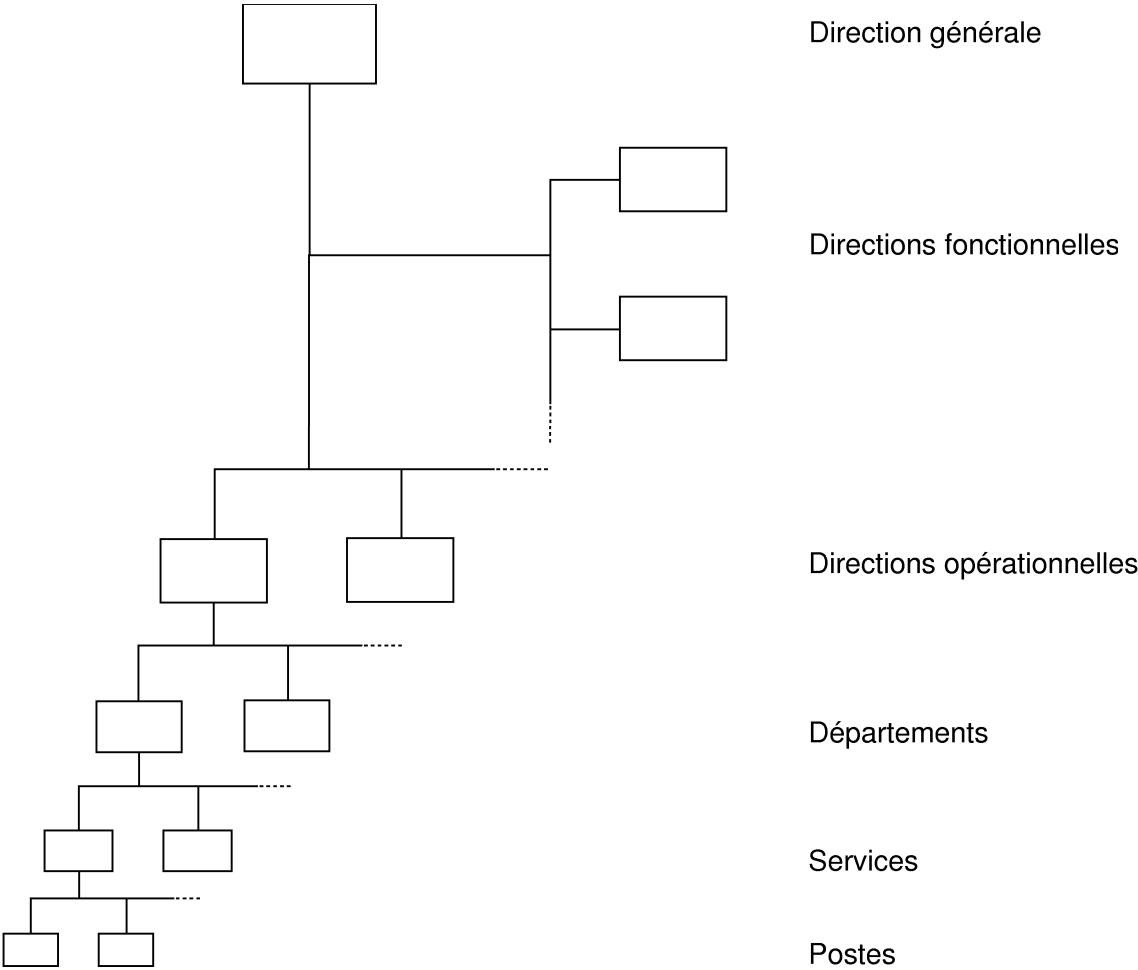
des données :

modèle entité-association, modèle organisationnel des données, modèle logique des données (réparties), modèle navigationnel, modèle relationnel, modèle objet, datagramme, grille d'analyse des rubriques, fiches de description de rubrique/fichier/document, etc.

Objectif

Représentation de la hiérarchie de l'organisation

Représentation graphique



Objectif

Description d'un poste de travail (ou fonction)

Remarque

Modèle de l'étude de l'existant de la méthode d'informatisation traditionnelle

Représentation graphique

Identification du poste de travail :

Personnel :

Fonction principale :

Documents entrants

Identification du document	Intitulé du document	Provenance du document	Volume / Période	Commentaires

Documents sortants

Identification du document	Intitulé du document	Destination du document	Volume / Période	Commentaires

Fichiers utilisés

Identification du fichier	Intitulé du fichier	Support	Volumes moyen / maxi.	Commentaires

Traitements effectués

Identification du traitement	Intitulé du traitement	Chrono.	Périodicité / Date de réf.	Algorithme général	Commentaires

Objectif

Description des calculs (des sorties à partir des entrées et des transformations) à l'intérieur du système

Remarque

Modèle de la méthode OMT

Composants

Un arbre de décomposition fonctionnel

Un ensemble de diagrammes de flots de données (et de contraintes)

Objectif

Spécification des fonctionnalités du système

Remarque

Modèle composant, avec le diagramme de flots de données, le modèle fonctionnel de la méthode OMT

Démarche

Regrouper les fonctionnalités du système en catégories

Créer un arbre fonctionnel de haut niveau pour chaque catégorie

Décomposer chaque arbre fonctionnel de haut niveau

Représentation graphique

Illustration des liens entre les fonctionnalités

Représentation textuelle

Spécification des fonctionnalités de haut niveau

Description

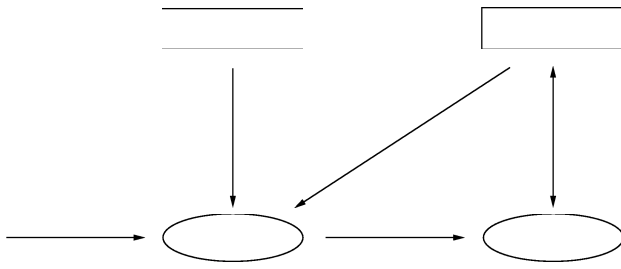
Contraintes pré-conditions et post-conditions

Spécification des fonctionnalités de bas niveau (opérations)

Opérations attachées à des classes

Arguments en entrée et en sortie, objets concernés

Contraintes sur les entrées



Objectifs

Description des relations fonctionnelles entre les valeurs du système (entrantes, calculées, stockées, sortantes)

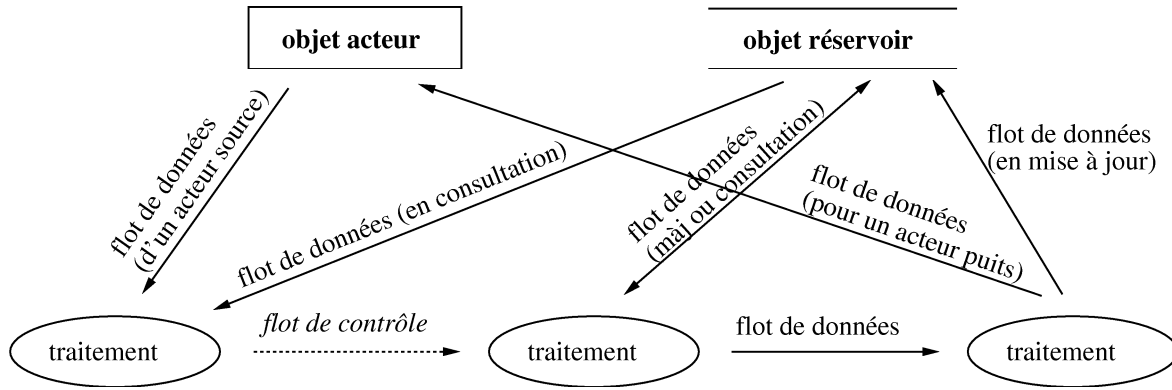
Description du flot des valeurs des données à partir de leur source dans les objets, passant par les traitements qui les transforment, vers leur destination dans les objets

Remarques

Modèle très prisé par les anglo-saxons

Modèle composant intégralement le modèle fonctionnel de la méthode OMT

Diagramme de flots de données



Graphe orienté où les sommets correspondent aux objets (acteur ou réservoir) et aux traitements, et où les arcs correspondent aux flots (de données ou de contrôle)

Objet acteur (ou borne) : objet actif qui dirige le diagramme de flots de données en produisant ou en consommant des données

Objet réservoir (ou objet de stockage de données) : stockage (passif) de données pour un accès ultérieur

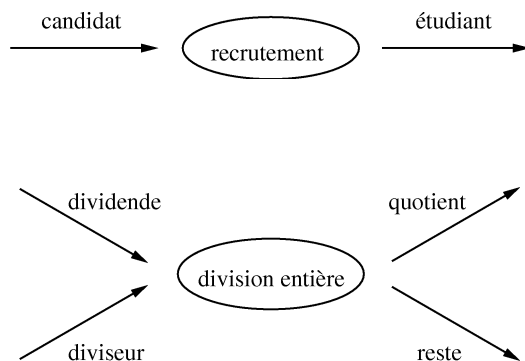
Autorise les opérations de mise à jour (création, modification, suppression) et d'interrogation (accès à la totalité ou à une partie des données, dans un ordre éventuellement différent de l'ordre de stockage)

Traitement : transformation des valeurs des données

Diagrammes de flots de données imbriqués pour différents niveaux d'abstraction des traitements : un traitement de haut niveau peut être étendu en un diagramme de flots de données complet tandis qu'un traitement de bas niveau correspond à une fonction élémentaire

Le nombre de flots de données en entrée et en sortie d'un traitement est fixe

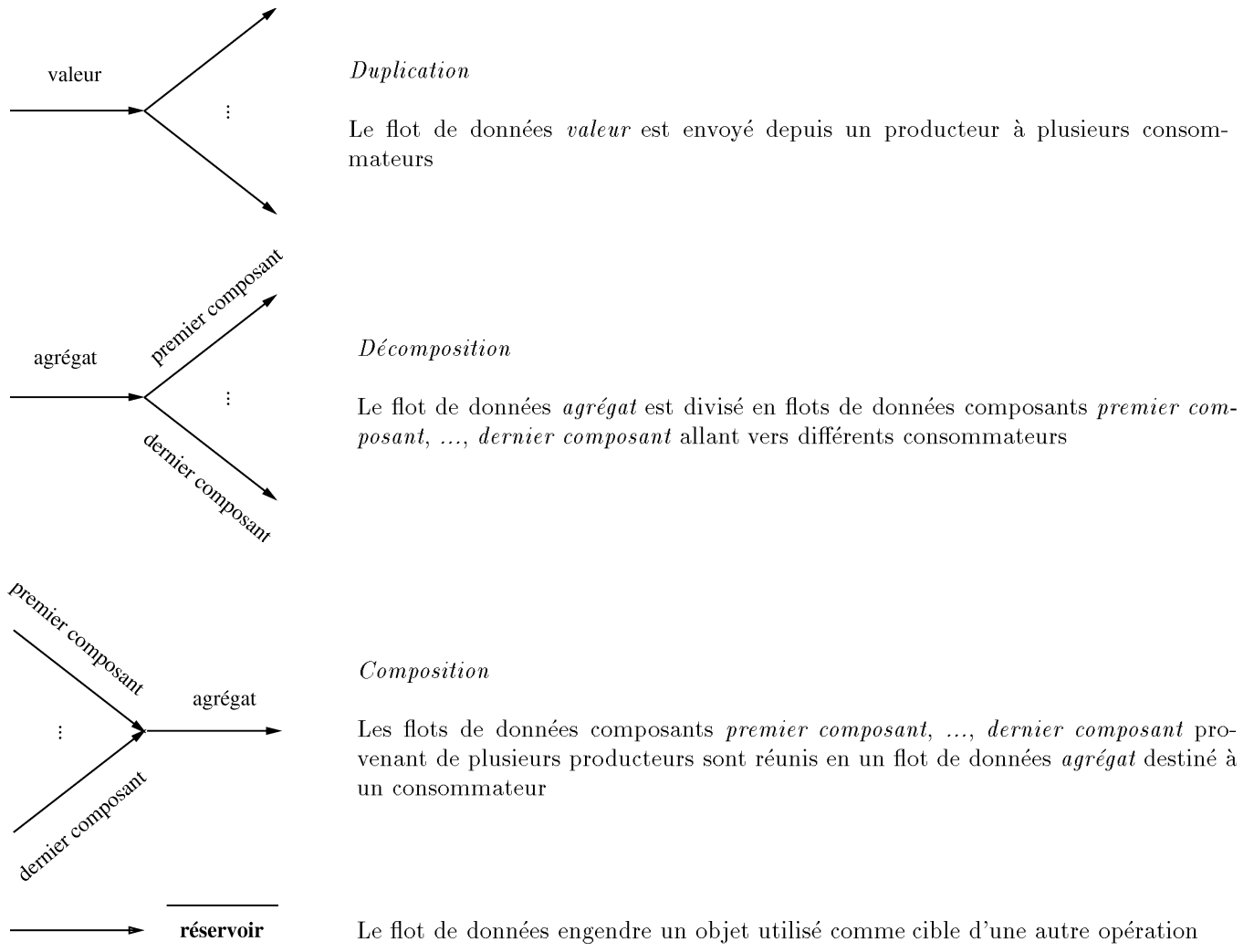
Exemples :



Flot de données : transport des valeurs de données : relie la sortie d'un producteur (un objet ou un traitement) à l'entrée d'un consommateur (un autre objet ou un autre traitement) pour transmettre la valeur de la donnée

Les données manipulées sont des objets (ou certains de leurs attributs) du modèle objet

Les flots de données à la frontière du diagramme de flots de données sont les flots de données en entrée ou en sortie

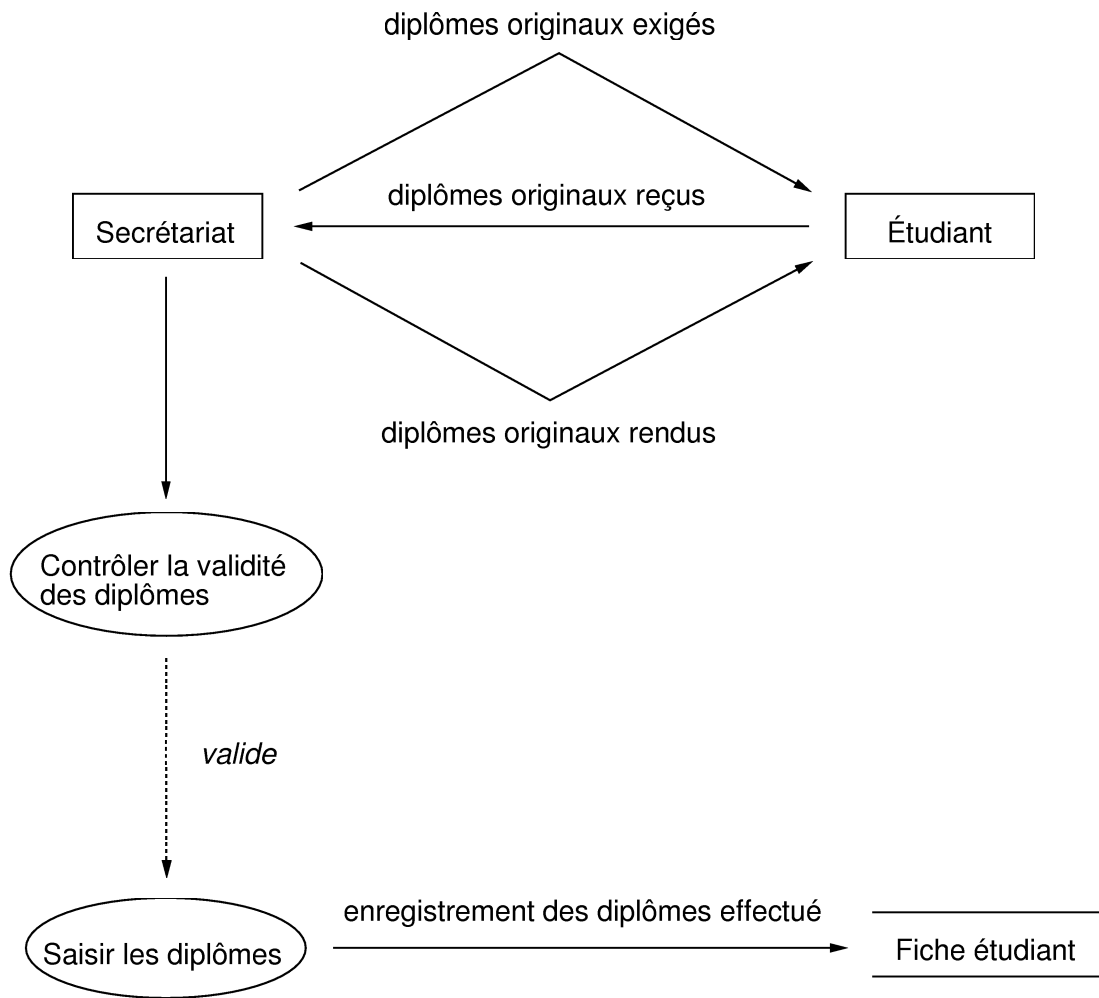


Flot de contrôle : condition booléenne à évaluer sur un traitement (produisant la valeur booléenne) pour que soit déclenché un autre traitement (dit traitement contrôlé)

Contrainte : fonction totale (une valeur est complètement précisée par une autre) ou partielle (valeur bornée par une autre, mais non complètement précisée) faisant apparaître la relation entre deux objets à un même moment

Invariant : contrainte entre les valeurs d'un objet tout au long de sa vie

L'enregistrement des diplômes d'un étudiant





Autres dénominations

Diagramme des flux

Modèle conceptuel de communication d'AMC*Designor

Modèles voisins

Modèle de flux

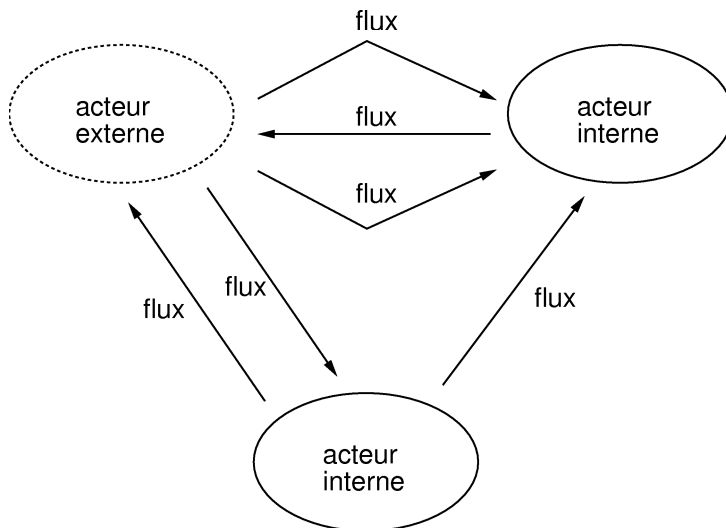
Objectif

Modélisation de la communication entre les acteurs intervenant pour le système d'information étudié

Remarque

Modèle de l'étape d'analyse de l'existant dans certaines variantes de la méthode MERISE

Graphe acteurs-flux



Graphe orienté où les sommets correspondent aux acteurs et où les arcs correspondent aux flux

Acteur (de gestion)

Entité (personne physique ou morale, service, poste de travail, etc.) qui joue un rôle dans le transit d'information et est censé produire un flux ; il s'agit donc d'un lieu de gestion

Exemples : un étudiant, le secrétariat

Interne (appartient au système d'information étudié) ou externe

Flux

Échange entre un acteur émetteur et un autre acteur récepteur, vérifiant qu'au moins un des deux acteurs est interne

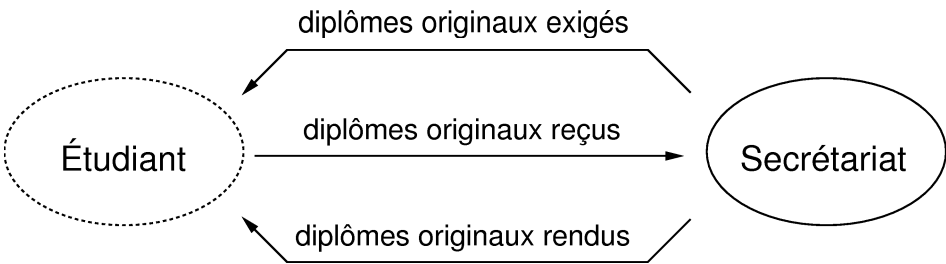
Exemple : livraison de marchandises, paiement de l'inscription à l'université, justificatif d'absence

Les flux peuvent être physiques (biens, marchandises, etc.), monétaires (espèces, chèque, traite, etc.) ou d'information

Remarque : un flux physique ou monétaire s'accompagne toujours d'un flux informationnel

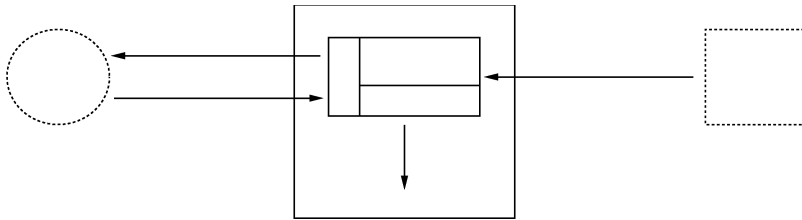
Le support de l'information peut être la voix (information orale), le papier (information documentaire) ou magnétique (information informatique)

La gestion de l'enregistrement des diplômes de tous les étudiants



La matrice des flux correspondante

<i>De</i> \ <i>Vers</i>	Secrétariat	Étudiant
Secrétariat		diplômes originaux exigés diplômes originaux rendus
Étudiant	diplômes originaux reçus	



Modèles voisins

Graphe acteurs-flux

Objectif

Détermination du domaine d'étude et de ses échanges avec l'environnement puis, par affinages successifs, les activités et la composition du domaine d'étude sans en décrire le comportement

Remarque

Modèle très prisé par les anglo-saxons

Modèle de MERISE/2

Type d'acteur

Regroupement d'acteurs exerçant des activités identiques

Type de site

Regroupement géographique et/ou fonctionnel de types d'acteurs

Type de poste

Rapprochement entre un type d'acteur et un type de site

Modèle de contexte

Description des interactions entre le domaine étudié et son environnement ou avec les domaines connexes

Il correspond au modèle de flux de niveau 0

Modèle de flux conceptuel

Description de la composition du domaine étudié, par affinages successifs des activités

Il est constitué des diagrammes de flux de niveau 1, 2, ...

Modèle de flux organisationnel

Description des échanges d'information entre différents types de sites et/ou types de postes, ou avec les acteurs externes et les domaines connexes

Le flux organisationnel précise son support (papier, disquette, téléphone, transfert informatique), son volume et sa périodicité

Modèle de flux

Représenté par un modèle de contexte ou par un modèle de flux conceptuel ou par un modèle de flux organisationnel

Domaine d'étude

Sous-ensemble cohérent du système étudié, bien délimité, formant le contenu du sujet à étudier

Activité (*function*, *process*)

Ensemble de traitements homogènes qui transforment ou manipulent des données

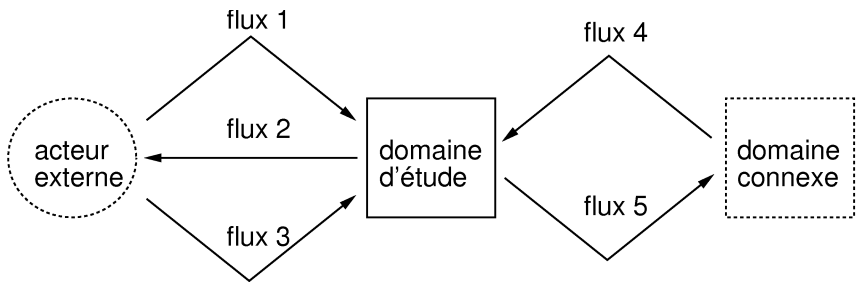
Flots de données (*data flow*)

Échange d'informations entre deux composants du système ou entre un composant du système et un système extérieur

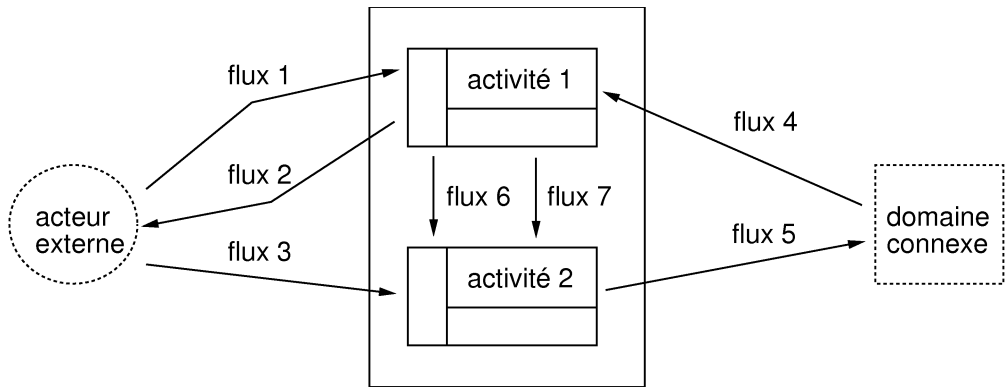
Système extérieur

Acteur externe (source ou destination des données située en dehors du système étudié) ou domaine connexe (composant de l'organisation étudiée interagissant avec le domaine étudié)

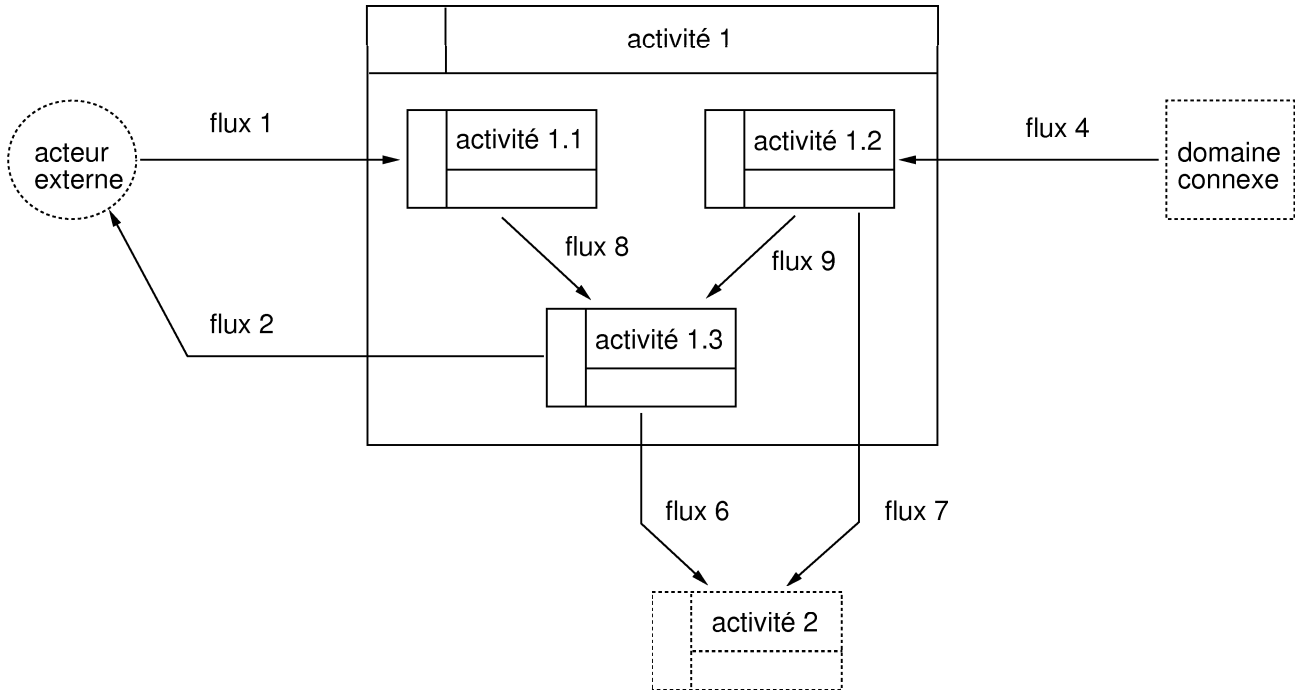
Modèle de contexte



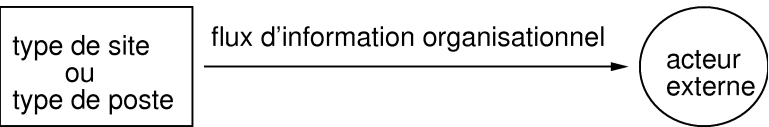
Modèle de flux conceptuel (diagramme de flux de niveau 1)



Modèle de flux conceptuel (diagramme de flux de niveau 2)

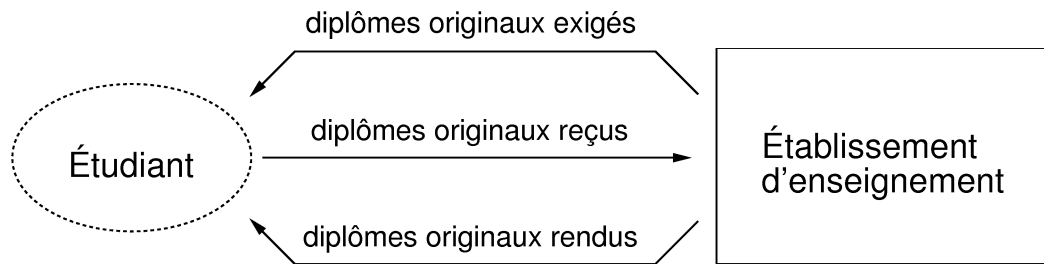


Modèle de flux organisationnel

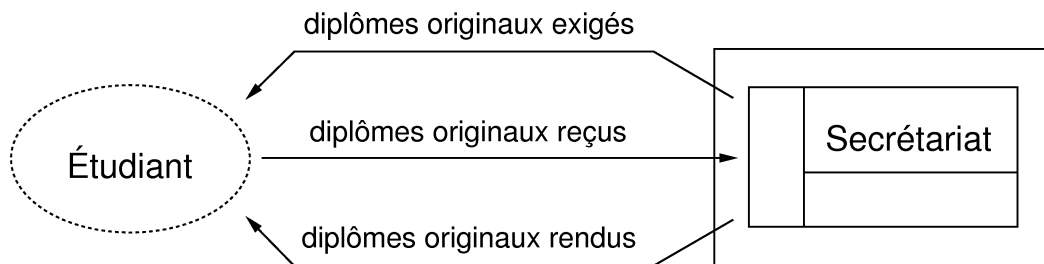


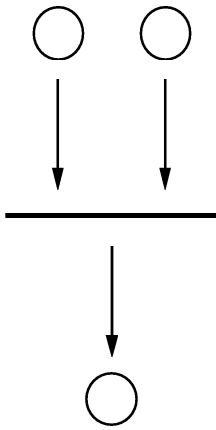
La gestion de l'enregistrement des diplômes de tous les étudiants

Le modèle de contexte



Le diagramme de flux de niveau 1





Origine

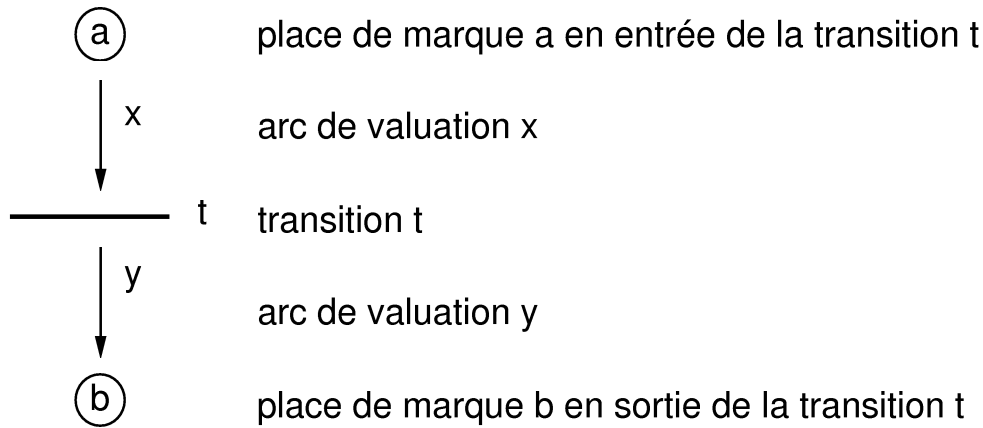
1983 : BRAMS

Remarques

Modélisation de la synchronisation (notamment à la base des modèles de traitements de MERISE)

Modèle dynamique des traitements (fonctionnement exprimé à partir de conditions d'activation et d'enchaînement de séquences élémentaires de traitements)

Définitions



On appelle place en entrée (respectivement en sortie) d'une transition s'il existe un arc valué allant de cette place (respectivement transition) vers cette transition (respectivement place)

La marque d'une place désigne le nombre de jetons de la place

Principe de fonctionnement

Transition franchissable (ou déclenchable) :

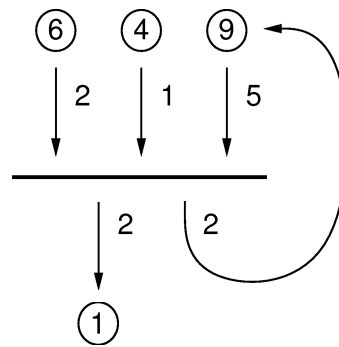
la marque de chaque place en entrée est supérieure ou égale à la valuation de l'arc correspondant

Le franchissement d'une transition induit :

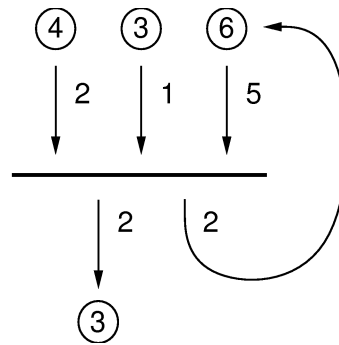
la consommation de jetons de chaque place en entrée (du nombre donné par l'arc valué correspondant)

l'apport de jetons à chaque place en sortie (du nombre donné par l'arc valué correspondant)

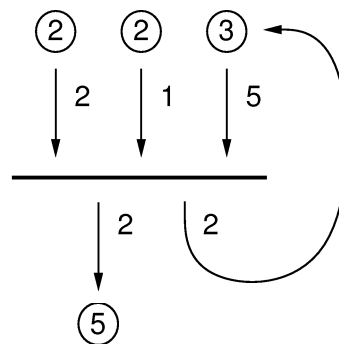
État initial



Après un premier franchissement

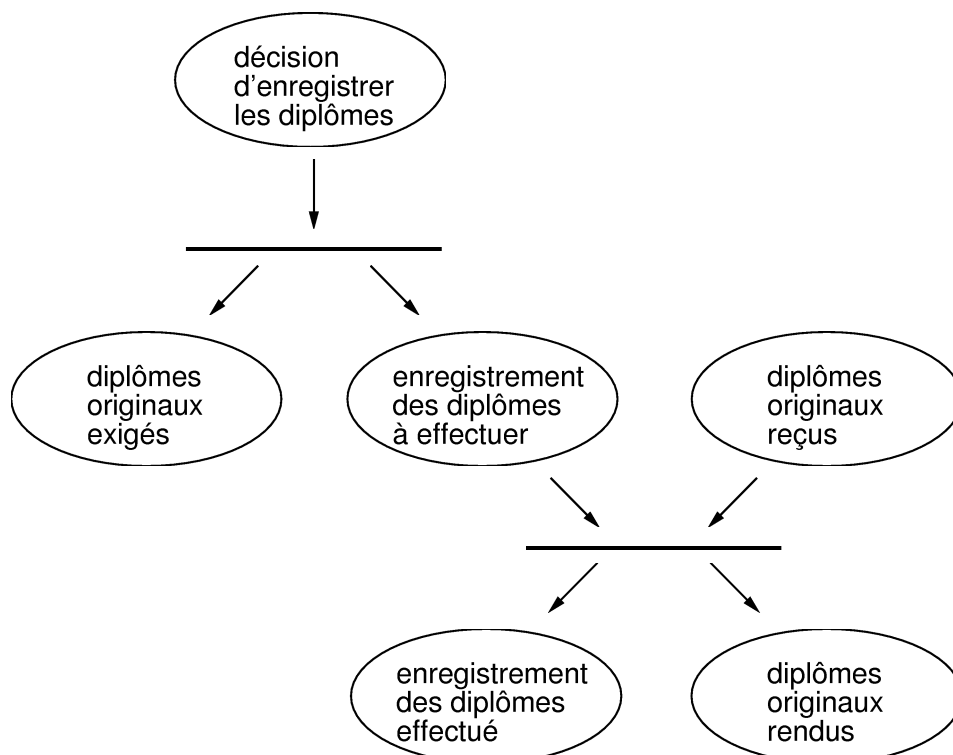


Après un second franchissement



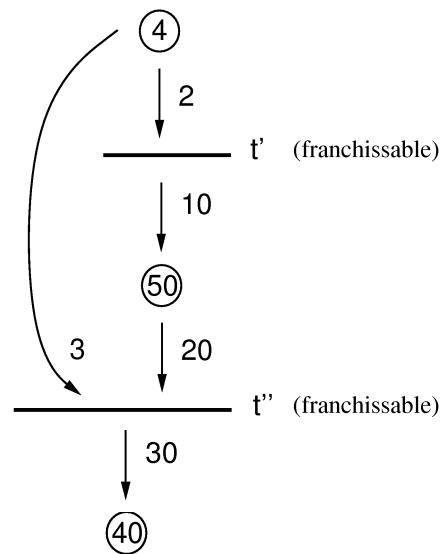
La transition est maintenant infranchissable ($3 < 5$)

La gestion de l'enregistrement des diplômes de tous les étudiants

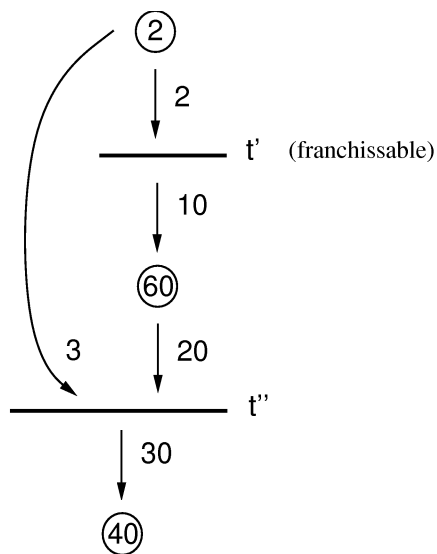


Remarque : l'ordre dans lequel se franchissent les transitions a des conséquences sur les marques des places du réseau de Pétri et sur les possibilités de franchissement des transitions à chaque étape

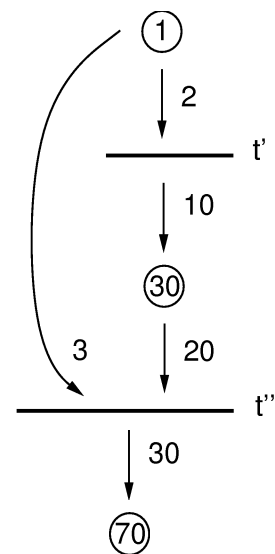
État initial



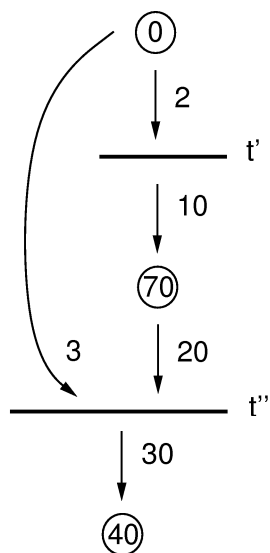
Après un premier franchissement de t'

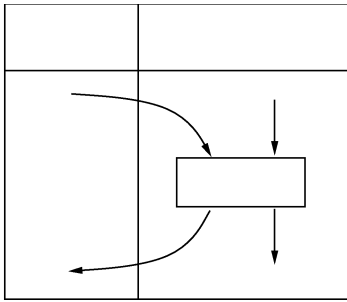


Après le franchissement de t''



Après un second franchissement de t'





Autre dénomination

Graphe de fonctionnement

Modèles voisins

Diagramme de circulation des documents (dans l'espace)

Modèle organisationnel des traitements de MERISE

Objectif

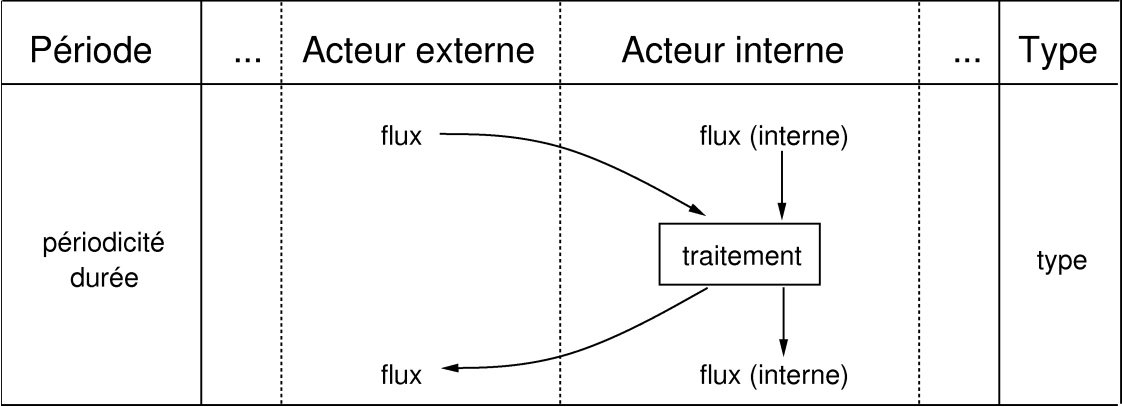
Prise en compte des traitements et de la chronologie des flux en plus de la communication entre les acteurs intervenant pour le système d'information étudié

Par contre, ce modèle ne permet pas de représenter la synchronisation des flux en entrée et les conditions d'émission des flux en sortie pour les traitements ; de même, les données ne sont pas représentées

Remarque

Modèle de l'étape d'analyse de l'existant dans certaines variantes de la méthode MERISE

Diagramme de circulation de l'information : représentation graphique



Acteur

Entité (personne physique ou morale, service, poste de travail, etc.) qui joue un rôle dans le transit d'information et est censé produire un flux

Interne (appartient au système d'information étudié) ou externe

Flux [d'information]

Échange entre deux acteurs (ou un seul pour un flux interne)

Primaire (apparaît en premier dans le système d'information étudié) ou secondaire (son émission est subordonnée à la préexistence d'autres flux)

Le support de l'information peut être la voix (information orale), le papier (information documentaire) ou magnétique (information informatique)

Traitement

Production de flux en sortie en réaction à des flux en entrée

Exemple : l'enregistrement des diplômes (d'un étudiant)

Période

Périodicité et durée du traitement

Exemple : entre 9 et 10 heures durant la semaine de la rentrée, et d'une durée de 20 minutes en moyenne

Type (de traitement)

Précision sur le degré d'automatisation

Manuel (aucune automatisation) , interactif (partiellement automatisé), différé (entièrement automatisé)

Exemple : le traitement de l'enregistrement des diplômes est manuel

Remarques

- Tout traitement doit avoir au moins un flux en entrée et au moins un flux en sortie
- Seuls les traitements des acteurs internes sont représentés
- Tous les acteurs (en colonne) et tous les flux (passant d’une colonne à une autre) du graphe acteurs-flux sont représentés
- De nouveaux flux (flux internes) mettant en jeu un seul acteur apparaissent en plus de ceux provenant du graphe acteurs-flux
- Le temps est implicitement représenté
- La chronologie des traitements est représentée par leur enchaînement (ainsi que par la période)
- Le plus souvent, un traitement engendre un flux pour l’acteur qui l’a réalisé; ainsi, l’acteur conserve une trace de ce qu’il a effectué

Exemple

La gestion de l’enregistrement des diplômes de tous les étudiants

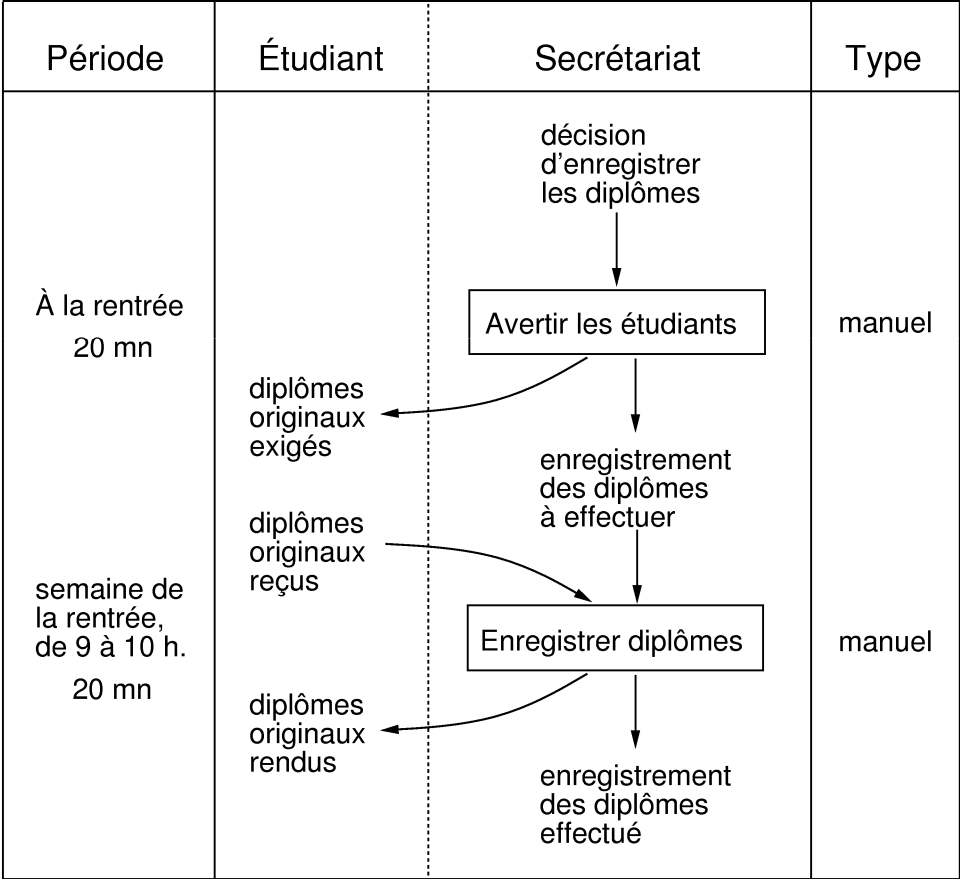
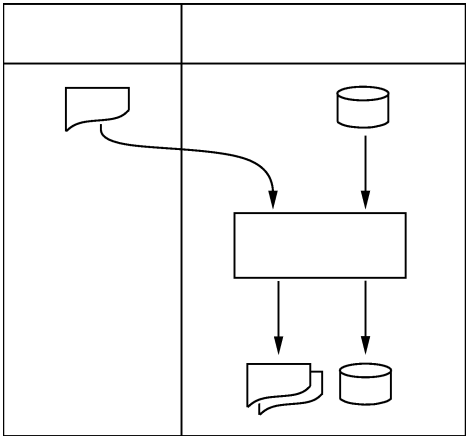


Diagramme de circulation des documents dans l'espace



Autre dénomination

Graphe de fonctionnement

Modèle quasiment identique au diagramme de circulation de l'information excepté le fait que ce sont les documents qui y sont représentés (et non l'information) et que des symboles précisent le support de l'information manipulée (la liste ci-dessous représente ceux qui sont les plus couramment utilisés)



document



clavier



écran



imprimé



carte
perforée



bande
magnétique



disque dur
magnétique



disque souple
magnétique



traitement



fusion



séparation

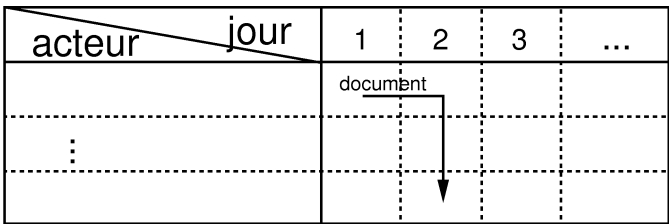


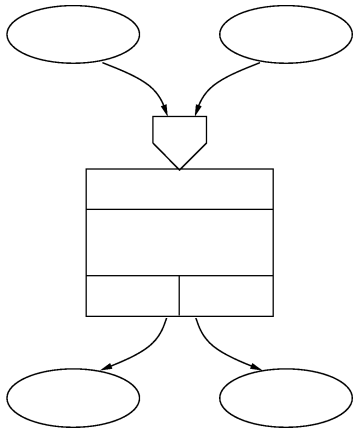
tri



interclassement

Diagramme de circulation des documents dans le temps





Modèle voisin

Actigramme

Modèle étendu

Modèle conceptuel des traitements analytique

Origine

1986 : H. TARDIEU et alter

Objectif

Modélisation des traitements (dynamiques)

Remarque

Modèle de MERISE

Domaine (de gestion)

Sous-ensemble cohérent du projet à traiter

Processus

Sous-ensemble cohérent d'un domaine

Règle de gestion

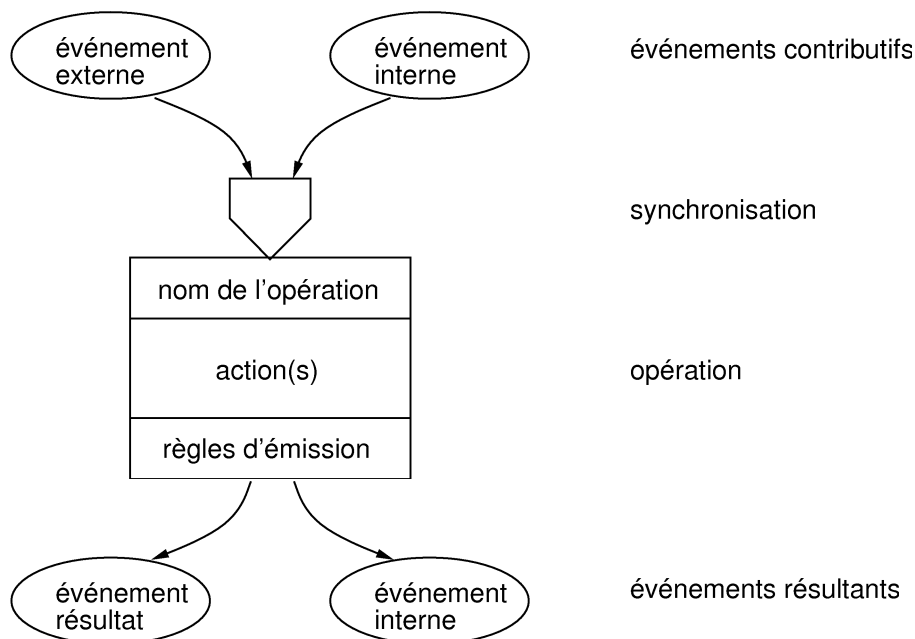
Loi générale à toute l'organisation s'appliquant à une ou plusieurs opérations

Traitement

Ensemble d'opérations déclenchées par des événements

Opération

Séquence ininterrompible d'actions, productive d'événements



Ses paramètres sont la durée, les actions, les événements contributifs et leur synchronisation, les événements émis et leurs règles d'émission

Action

Fonction élémentaire telles que la mise à jour (création, modification, suppression) ou l'interrogation

Événement

Circonstance portée à la connaissance du système et à laquelle celui-ci doit réagir, c'est à dire flux ou fait contribuant à déclencher une opération ou résultant d'une opération

Externe (provenant de l'extérieur), interne (généré par le système pour poursuivre son activité), résultat (généré par le système et émis vers l'extérieur)

Ses paramètres sont la capacité (nombre maximal d'occurrences), la fréquence (loi d'arrivée des occurrences)

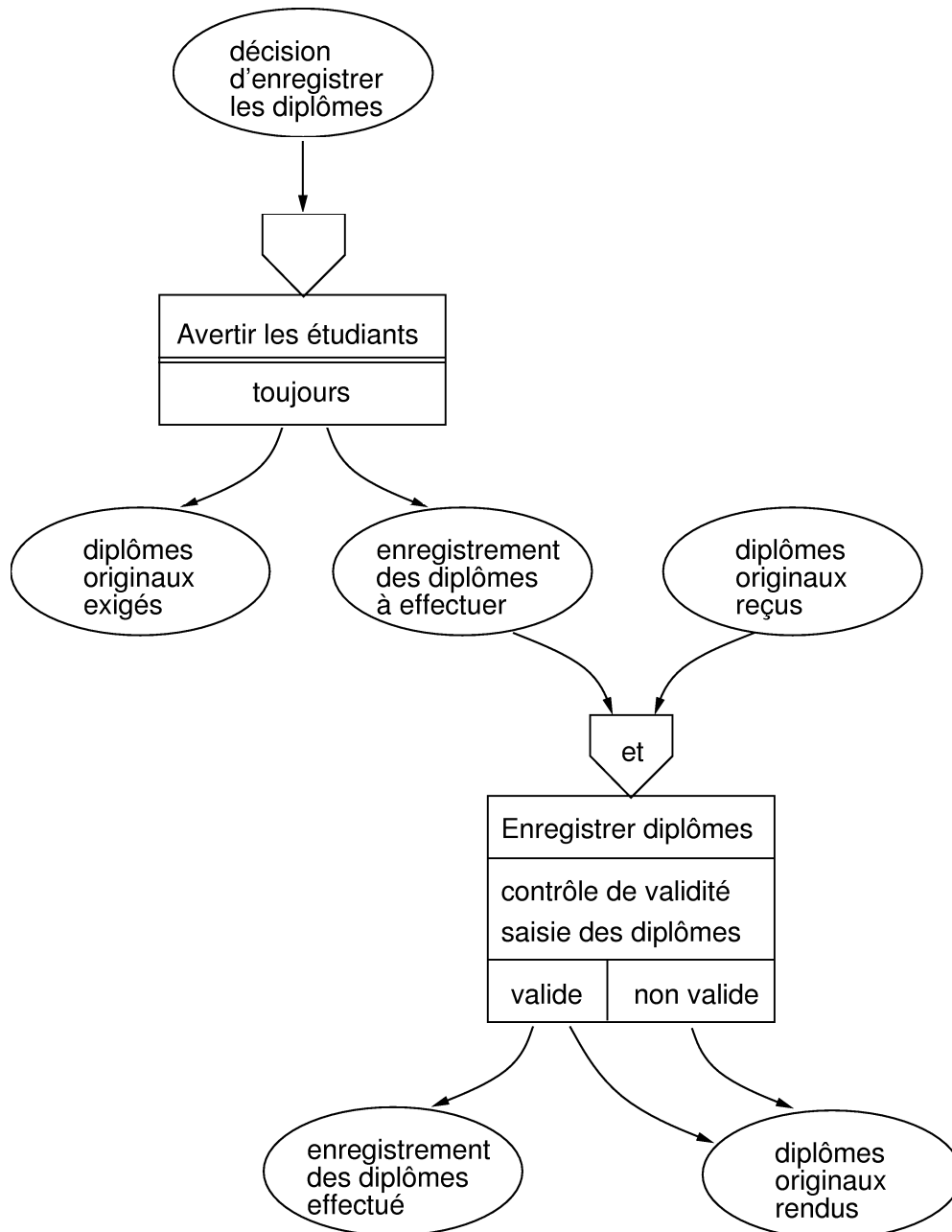
Synchronisation

Ensemble de conditions sur les événements contributifs déterminant la situation de déclenchement de l'opération

Règle d'émission

Condition pour laquelle des événements résultants seront produits par l'opération

La gestion de l'enregistrement des diplômes de tous les étudiants



Remarques

Le modèle conceptuel des traitements ne doit exprimer que ce qui doit être fait (correspond au *quoi* mais pas aux *qui*, *quand*, *où*, *comment*)

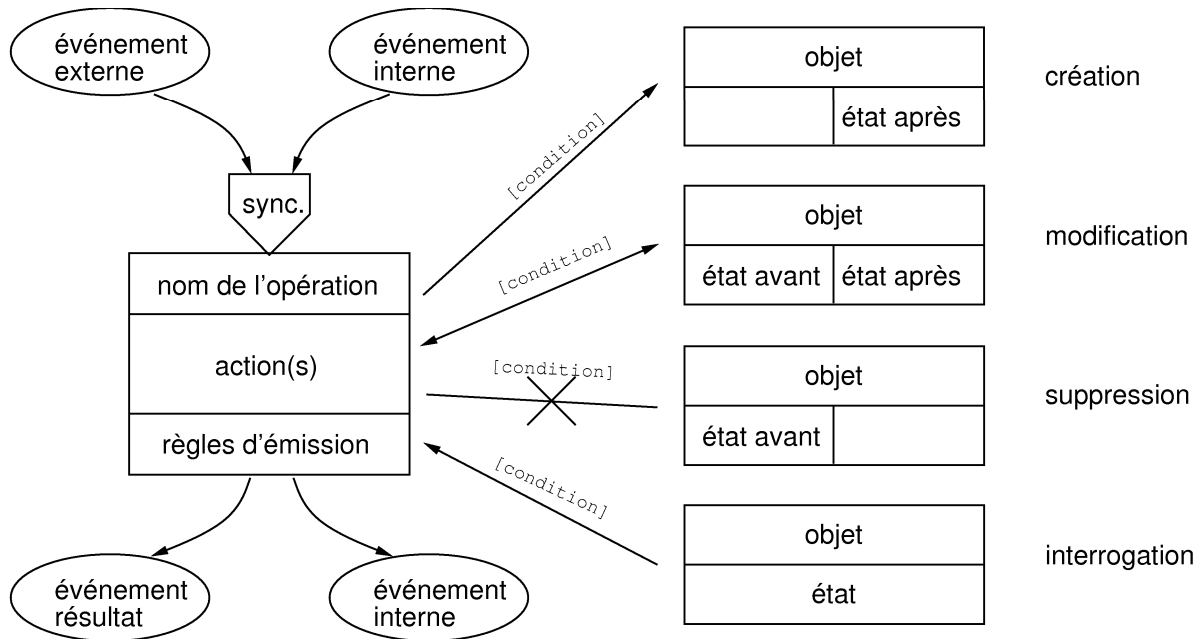
Regrouper/Éclater les traitements en opérations ininterruptibles

Toute opération doit avoir au moins un événement contributif, une règle de synchronisation s'il y a plusieurs événements contributifs et qui doit pouvoir prendre la valeur vraie, au moins un événement résultant pour chaque règle d'émission, être ininterruptible tout comme les actions la composant

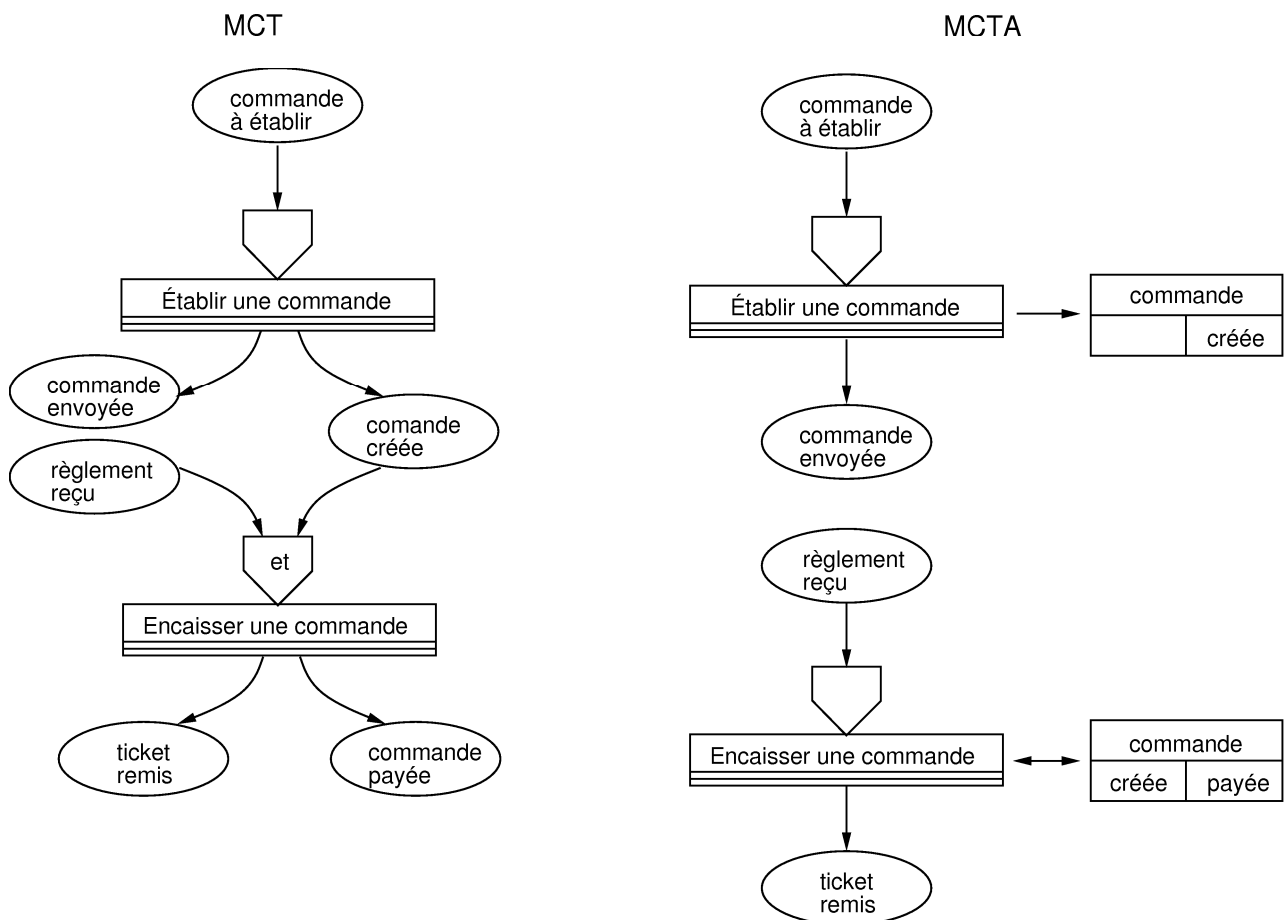
Une et une seule des règles d'émission de chaque opération doit être vérifiée

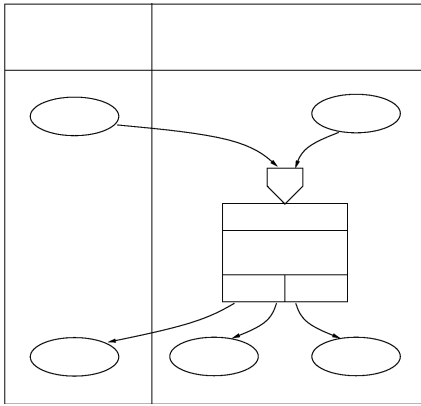
Introduit par MERISE/2 pour intégrer le cycle de vie des objets au modèle conceptuel des traitements

Nouvelle représentation graphique d'une opération (conceptuelle)



Comparaison entre deux schémas conceptuel des traitements (analytique ou non) décrivant la même réalité





Modèle voisin

Diagramme de circulation de l'information

Origine

1986 : H. TARDIEU et alter

Objectif

Modélisation dynamique des traitements (niveau organisationnel)

Remarque

Modèle de MERISE

Procédure

Succession de phases appartenant à un même processus et exécutée par un ensemble d'acteurs; c'est un sous-ensemble d'un processus du modèle conceptuel des traitements déclenché par un ou plusieurs événements

Règle d'organisation

Choix d'organisation s'appliquant à une ou plusieurs phases

Acteur

Entité organisationnelle chargée d'exécuter un certain nombre de phases

Interne (appartient au domaine d'étude) ou externe

Poste de travail

Concept d'acteur muni de caractéristiques organisationnelles

Phase

Suite ininterrompue de tâches exécutées par un acteur interne

Ses paramètres sont la période, le type, les tâches, les événements contributifs et leur synchronisation, les événements émis et leurs règles d'émission

Période

Description de caractéristiques organisationnelles telles que la périodicité (aléatoire, journalière, etc.), la durée (de début, moyenne ou maximale), le responsable et les ressources (moyens permettant la réalisation de certaines actions tels que des logiciels, du matériel, des objets divers, etc.)

Type (d'une phase)

Précision sur le degré d'automatisation

Manuel (aucune automatisation) , interactif (partiellement automatisé), différé (entièrement automatisé)

Tâche

Fonction élémentaire

Événement

Circonstance portée à la connaissance du système et à laquelle celui-ci doit réagir, c'est à dire flux ou fait contribuant à déclencher une phase ou résultant d'une phase

Externe (provenant de l'extérieur), interne (généralisé par le système pour poursuivre son activité), résultat (généralisé par le système et émis vers l'extérieur)

Ses paramètres sont la capacité (nombre maximal d'occurrences), la fréquence (loi d'arrivée des occurrences)

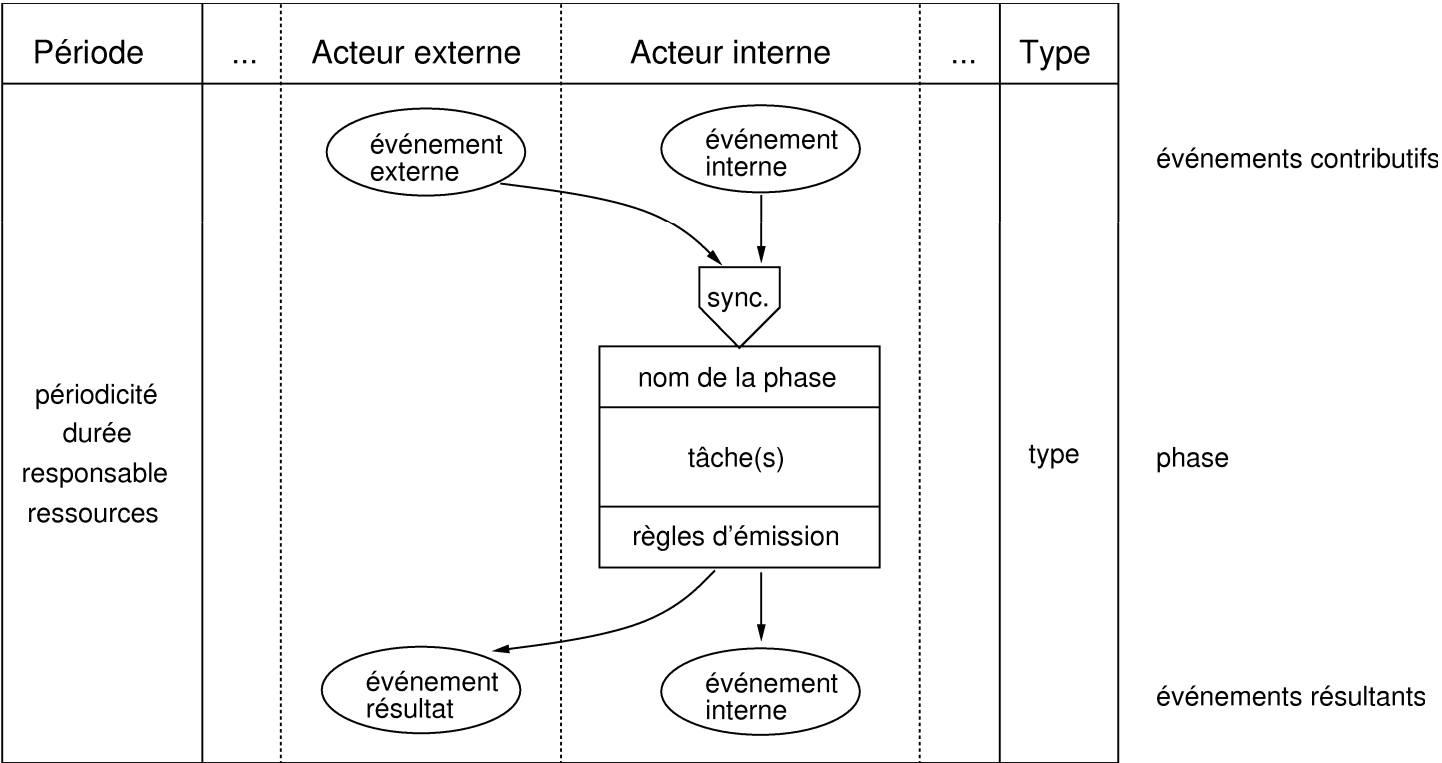
Remarque : comparativement au modèle conceptuel des traitements, des événements organisationnels peuvent apparaître

Synchronisation

Ensemble de conditions sur les événements contributifs déterminant la situation de déclenchement de la phase

Règle d'émission

Condition pour laquelle des événements résultants seront produits par la phase

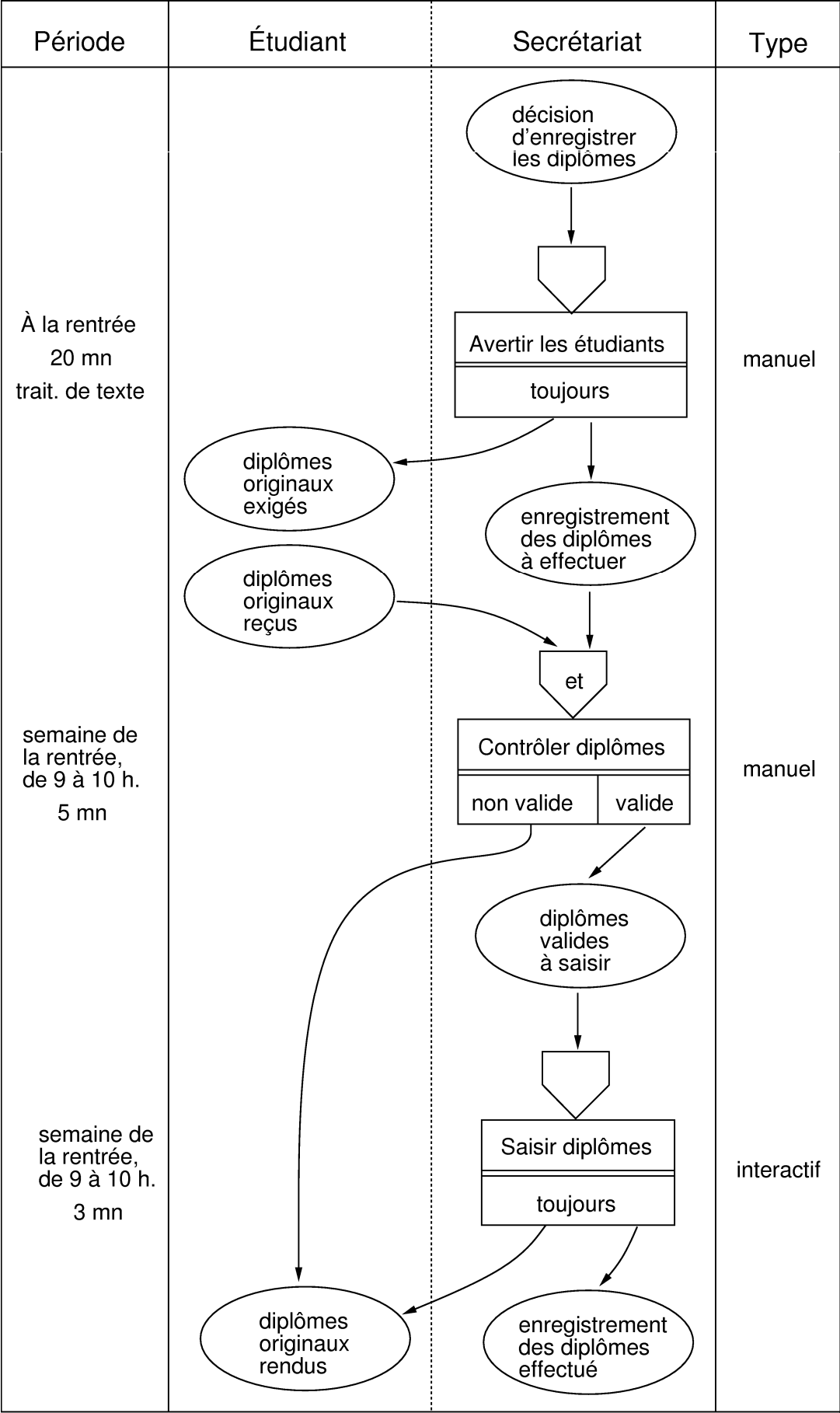


Remarques

Le modèle organisationnel des traitements exprime que ce qui doit être fait mais également qui fait, quand et où cela doit être fait ; par contre, il ne doit pas exprimer comment cela doit être fait

Toute phase doit avoir au moins un événement contributif, une règle de synchronisation s’il y a plusieurs événements contributifs et qui doit pouvoir prendre la valeur vraie, au moins un événement résultant pour chaque règle d’émission, être ininterrompible tout comme les tâches la composant

La gestion de l'enregistrement des diplômes de tous les étudiants



Définitions

Type d'acteur

Regroupement d'acteurs exerçant des activités identiques

Type de site

Regroupement géographique et/ou fonctionnel de types d'acteurs

Type de poste

Rapprochement entre un type d'acteur et un type de site

Nature

Précision sur le degré d'automatisation

Manuel (M), conversationnel (C), automatique (A)

Périodicité

Journalier (J), hebdomadaire (H), mensuel (M), trimestriel (T), semestriel (S), annuel (A)

Événement

Flux ou fait contribuant à déclencher une procédure fonctionnelle ou résultant d'une procédure fonctionnelle

Composant d'événement

Élément(s) affinant un événement (caractéristiques)

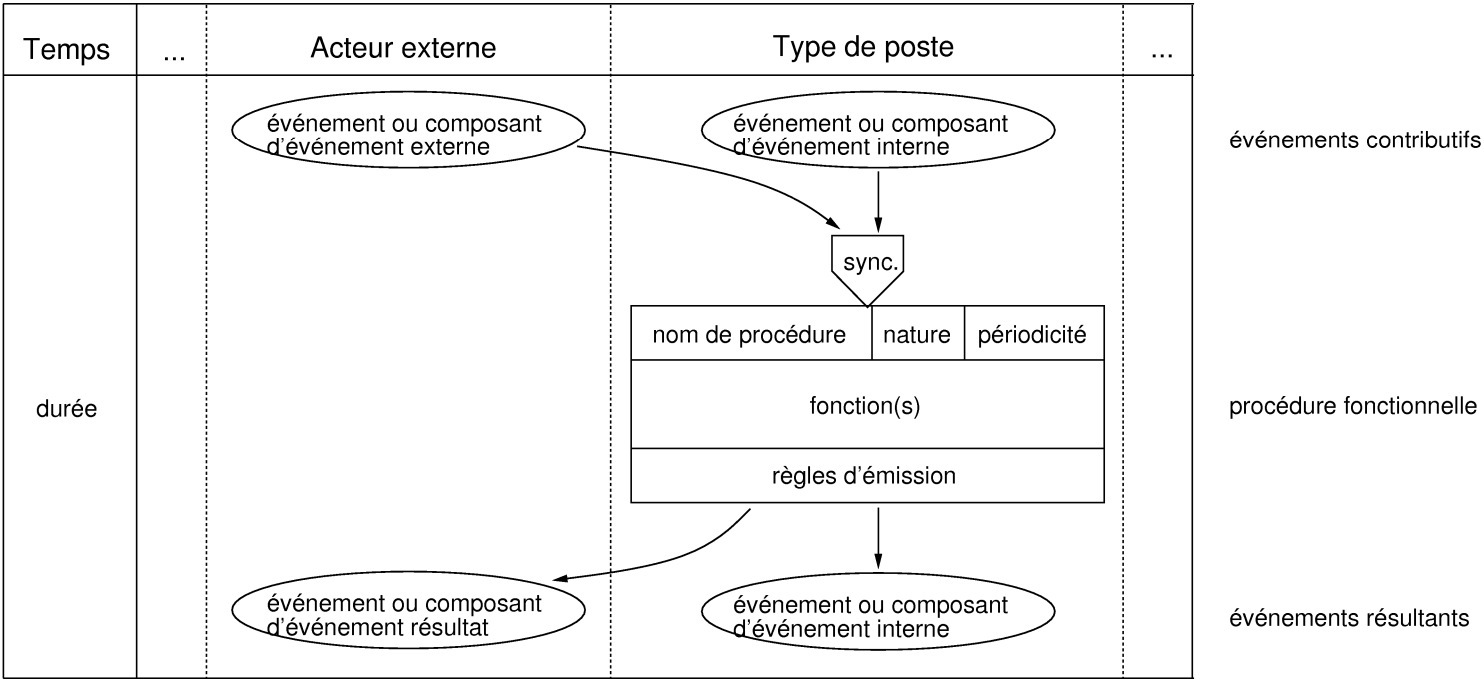
Fonction

Traitement élémentaire (de même nature) réutilisable

Procédure fonctionnelle

Ensemble de fonctions concourant au même objectif, exécutées par un type de poste à partir d'un même type d'interface, déclenchées par au moins un événement et fournissant au moins un résultat, selon une nature de traitement et un délai de réponse (immédiat ou différé) définis

Représentation graphique

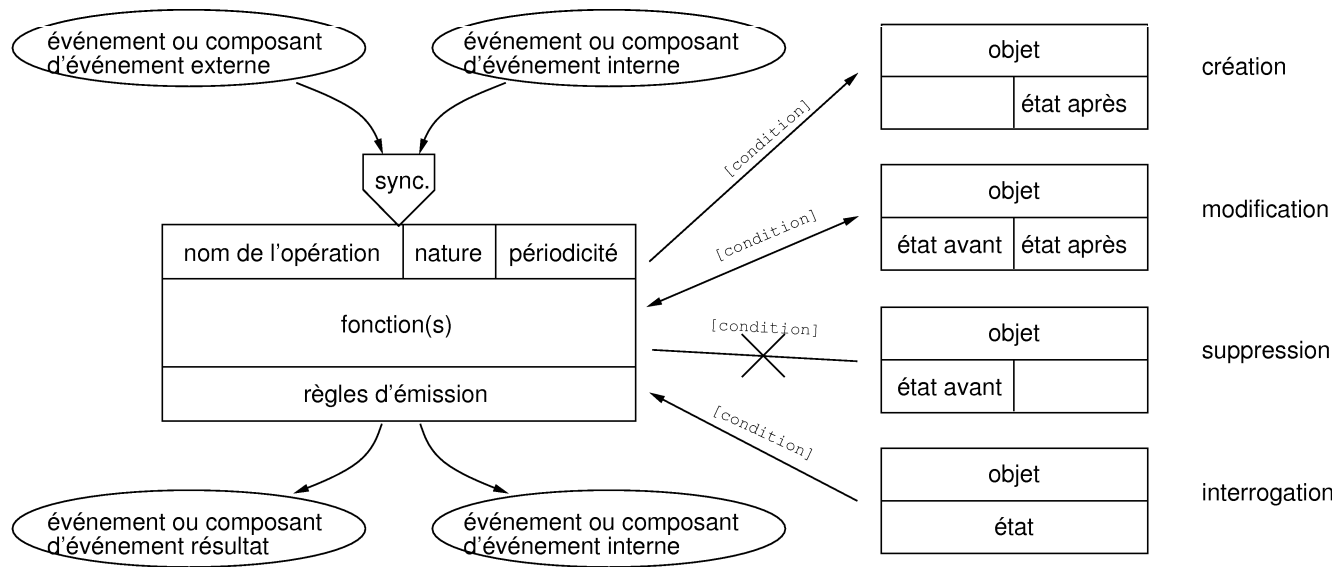


Introduit par MERISE/2

Définition d'une opération (organisationnelle)

Traitement (d'une seule nature) exécuté entièrement à partir d'un seul type de poste

Nouvelle représentation graphique d'une opération (organisationnelle)



Objectif

Modélisation des traitements d'une application dans le contexte technologique actuel

Remarque

Modèle de MERISE/2

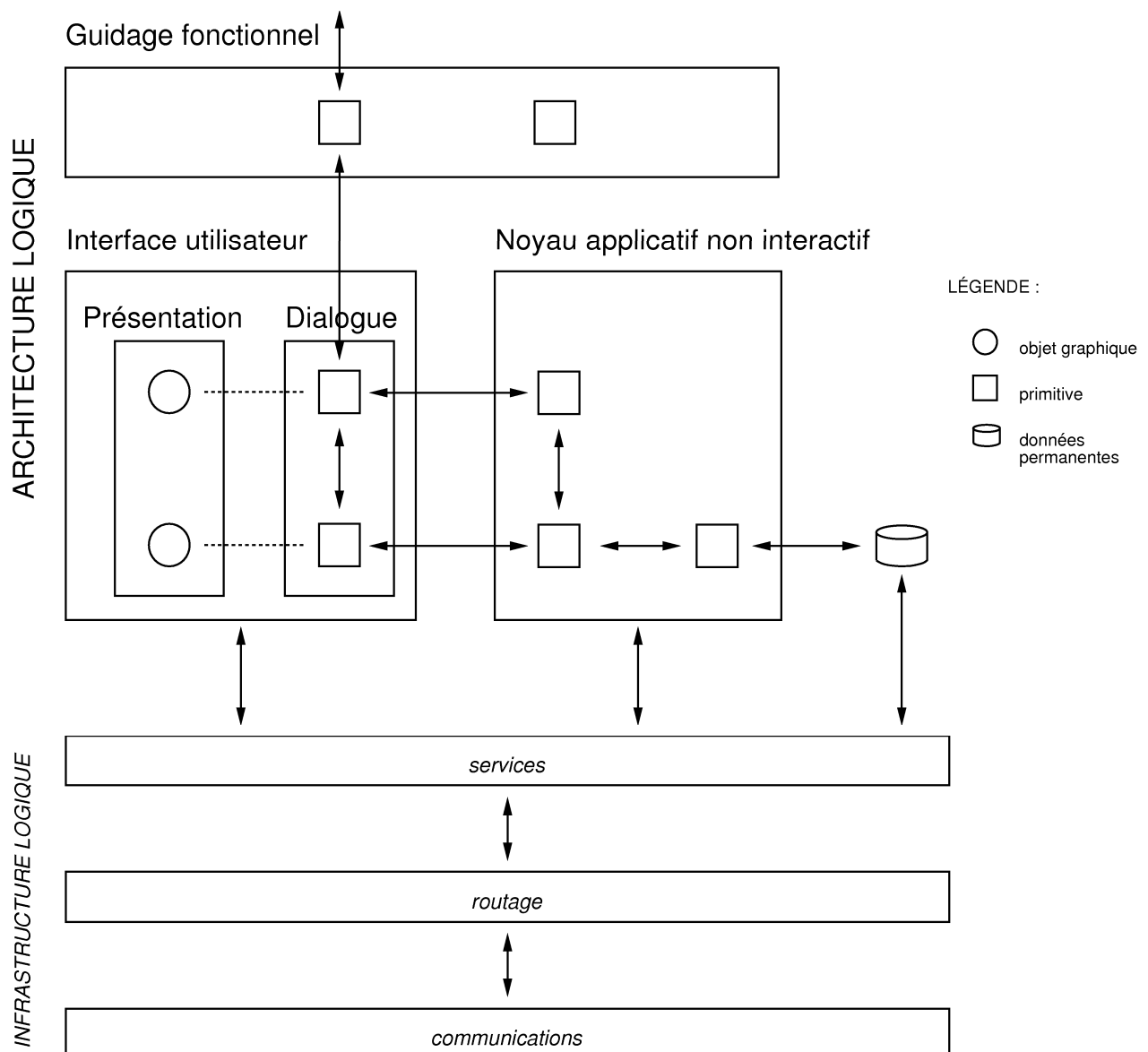
Interface utilisateur (IHM) graphique

Partie de l'application permettant les échanges avec l'utilisateur (en mode conversationnel ou même automatique)

Niveaux de réaction (modèle langage) : lexical, syntaxique, sémantique

Indépendance entre les couches présentation (fenêtre, objets réactifs), dialogue (animation, contrôles syntaxiques, fonctions standards) et noyau applicatif non interactif (traitements et données)

Architecture moderne d'application



Approche procédurale ou événementielle

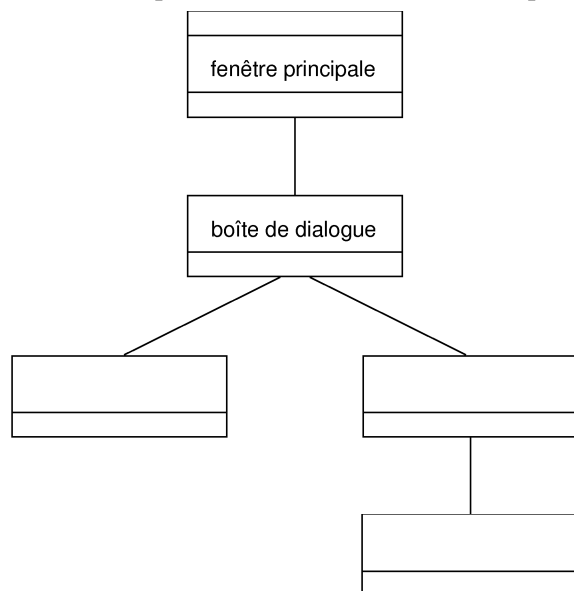
Rôle

Objets et actions affectés à l'utilisateur, c'est à dire la partie visible et audible de l'interface utilisateur

Assure un retour lexical (implicite) aux (profils) d'utilisateurs

Modèles

Arborescence représentant la hiérarchie statique des fenêtres et boîtes de dialogue



Description statique de fenêtre

Composition statique de fenêtre :						
Type :		Code :		Mode :		
Commentaires :						
Numéro	Type	Identificateur objet IHM	Contenu et observations	Attributs / Format externe	Valeur par défaut	Identificateur données MLD

Copies des écrans (fenêtres et boîtes de dialogue) et des imprimés

Normes

Normes générales (AFNOR Z 67-110 : AFNOR, Recueil de normes informatiques françaises, 1993)

Guide de style de l'entreprise

Guide de style de l'interface utilisateur concernée

APPLE human interface guidelines : the Apple desktop interface, Addison Wesley, 1986

IBM SAA : common user access advanced design guide, IBM, 1988

OSF/MOTIF style guide, rev. # 1.1, OSF, 1991

etc.

Guide de style du projet

Rôle

Assurer les échanges d'informations avec l'utilisateur, par l'intermédiaire de la couche présentation, en activant le noyau applicatif non interactif

Réalise le traitement de l'affichage, les contrôles de surface, les enchaînements des écrans, etc.

Assure un retour syntaxique et/ou sémantique aux (profils) d'utilisateurs

Prend en compte les actions des utilisateurs, assure un retour de premier niveau, réalise les contrôles de syntaxe ne nécessitant pas un accès à la base de données, fait exécuter au noyau applicatif non interactif les autres traitements, communique (par messages logiques) avec le guidage fonctionnel

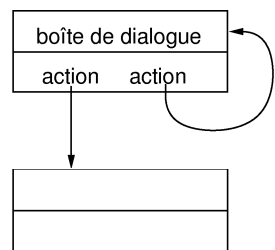
Définition

Primitive de dialogue

Événement portant sur un ou plusieurs objets graphiques de la présentation

Modèles

Diagrammes d'enchaînement de boîtes de dialogue



Description dynamique de fenêtre

Comportement dynamique d'une fenêtre :						
Type :		Code :		Mode :		
Commentaires :						
Numéro	Type	Identificateur objet IHM	État initial	Action utilisateur	Traitement dialogue	Traitement noyau applicatif non interactif

Matrice objets/actions

Objets \ Actions			...	
	...			
	...			
⋮	⋮	⋮	⋮	⋮
	...			

Autres formalismes : graphique, texte, langage

Rôle

Composant de l'application indépendant du degré d'automatisation de l'application et du type de l'interface utilisateur

Prend en compte la requête issue du dialogue, effectue les contrôles de fond et les calculs, accède aux données permanentes, assure à son niveau les cohérence/sécurité/intégrité, et restitue les résultats et les codes de retour

Définitions

Primitive applicative non interactive

Service ou ensemble de services, déclenché(s) par le dialogue ou une autre primitive applicative non interactive, appelé(s) par un requête (avec une liste de paramètres), restituant les valeurs des résultats et les codes de retour

Interroger ou mettre à jour des données, calculer

Garantir la cohérence, la sécurité, l'intégrité

Vue logique (d'une fonction logique)

Ensemble des data-items (plus petit élément logique d'information), segments logiques (ensemble de données élémentaires) et liens logiques (reliant les segments logiques) nécessaires à l'exécution de la fonction logique correspondante

Formalisme : modèle logique des données, en précisant les types d'accès

Procédure fonctionnelle (du niveau organisationnel [traitements])

Ensemble de fonctions concourant au même objectif, exécutées par un type de poste à partir d'un même type d'interface, déclenchées par au moins un événement et fournissant au moins un résultat, selon une nature de traitement et un délai de réponse définis

Fonction logique

Ensemble de composants applicatifs ayant un même degré d'automatisation (conversationnel ou automatique), utilisé par une ou plusieurs procédures fonctionnelles, doté d'une ou plusieurs interfaces utilisateurs de même nature (alphanumérique ou graphique), de niveaux de guidance (école, expert, etc.) quelconque

Transaction logique

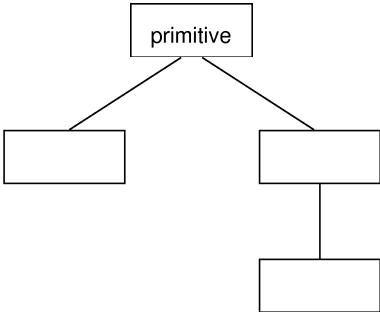
Ensemble unitaire (soit entièrement réalisé, soit entièrement annulé) de traitements algorithmiques et d'accès aux données

Message logique

Ensemble structuré de données échangées entre plusieurs composants de l'application

Modèles

Arbre de décomposition de primitives



Description d'une primitive

Description d'une primitive :				Type :	Code :
Commentaires :					
Messages logiques	Contenu (données)	Conditions	Traitements	Actions MLD	Objets MLD

Rôle

Assurer le bon fonctionnement de l'ensemble du système, notamment en assurant les liaisons entre fonctions logiques au sein d'une procédure fonctionnelle, ou entre procédures fonctionnelles au sein d'une application

Autorise le paramétrage de l'applicatif pour une organisation donnée

Effectue l'enchaînement, l'automatisation, la synchronisation des tâches utilisateurs, le contrôle des habilitations des utilisateurs

Modèles

Diagrammes d'enchaînement

Matrices de transition

Niveaux de complétude

Pilotage du routage logique

Gestion du contexte de travail (permettant d'assurer la synchronisation des interfaces utilisateurs sur le poste de travail)

Gestion des tâches, rôles, expertises d'un utilisateur (permettant d'assurer l'adaptation de la gestion du contexte à l'utilisateur)

Formalisme : modèle organisationnel des traitements

Gestion des tâches multi-utilisateurs (permettant d'assurer la gestion des flux de travail et du système de pilotage)

Formalisme : logique temporelle

Objectif

Adapter les modèles logiques des traitements dans le cadre d'une architecture technique à plusieurs niveaux, en affectant les ressources nécessaires (machines logiques types retenues dans le cadre du scénario de répartition) aux composants des fonctions logiques

Exemples

Affectation des composants de présentation et de dialogue sur la machine logique type de l'utilisateur (poste de travail) et des composants du noyau applicatif non interactif sur une machine serveur

Architecture client/serveur : le serveur supporte uniquement le SGBD et la base de données, tous les autres composants de l'application sont portés sur le poste de travail de l'utilisateur

Objectif

Description des infrastructures matérielles et logicielles en terme de machines logiques types (avec leurs flux) et de leurs regroupements sur des types de sites informatiques

Remarque

Modèle d'architecture logique de MERISE/2

Composants

Les schémas intermédiaires du schéma d'architecture logique des moyens informatiques sont les schémas d'architecture logique et les schémas d'architecture logique répartie

Définitions

Type d'acteur

Regroupement d'acteurs exerçant des activités identiques

Type de site

Regroupement géographique et/ou fonctionnel de types d'acteurs

Machine logique type

Ensemble de ressources matérielles et logicielles non séparables permettant d'effectuer des traitements et de stocker des données persistantes

Type de site informatique

Type de site doté de machine(s) logique(s) type(s)

Serveur logique

Regroupement de traitements et de données homogènes (d'un point de vue technico-organisationnel)

Serveur logique réparti

Serveur logique affecté à un type de site informatique ou à une machine logique type

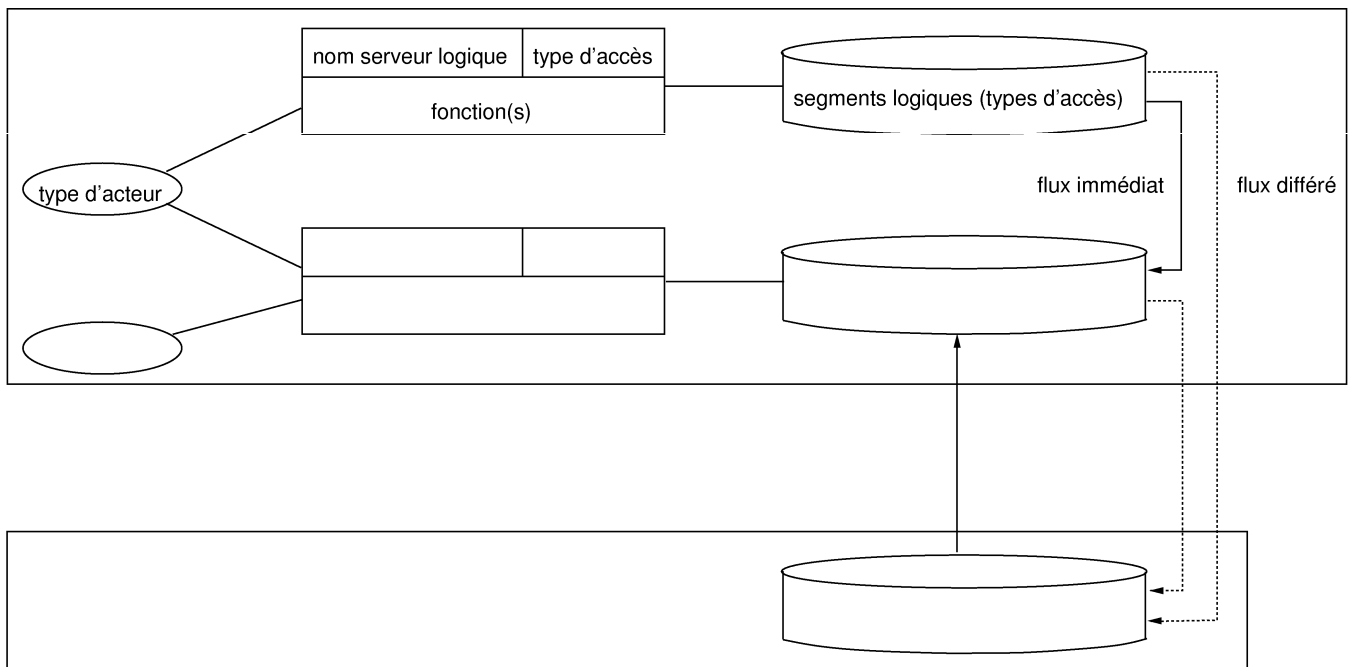
Type d'accès

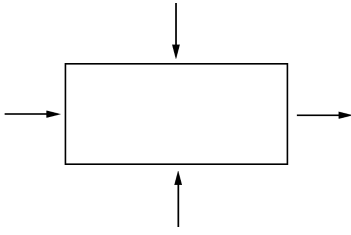
Restriction sur un type d'entités, un type d'associations ou un type de propriétés

Création (C), interrogation (I), modification (M), suppression (S)

Représentation graphique d'un schéma d'architecture logique répartie

Type de site informatique





Autre dénomination

Diagramme d'activités

Modèle voisin

Modèle conceptuel des traitements de MERISE

Objectif

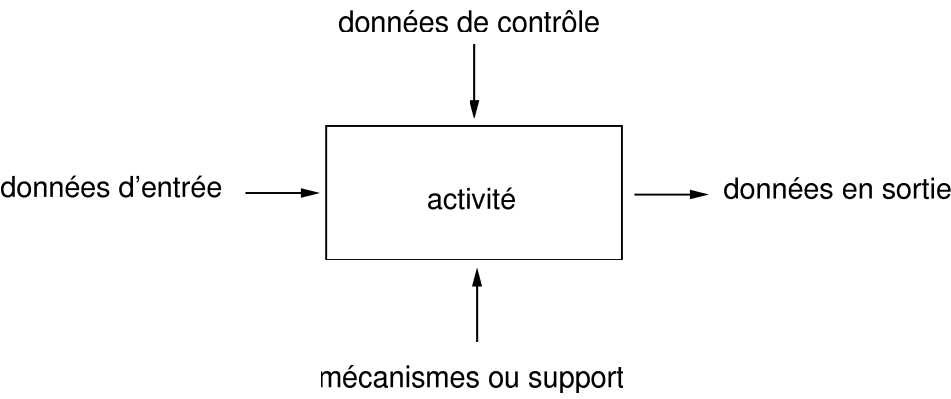
Représentation de l'ensemble des activités du système

Remarque

Modèle des traitements de la méthode SADT

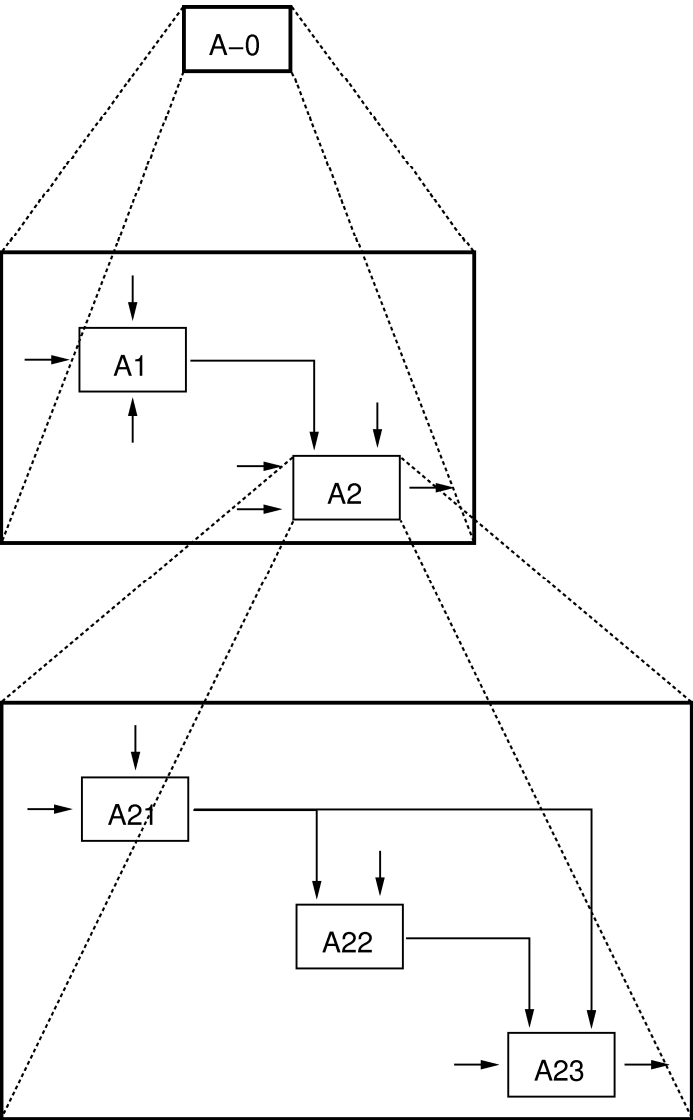
Une activité :

crée, génère une donnée en sortie
transforme, modifie, change d'état une donnée d'entrée
sollicite la donnée d'entrée à partir de données (directives) de contrôle non modifiées
en s'appuyant sur les potentialités des mécanismes ou supports de l'activité



Arbre hiérarchique des actigrammes

- $A - 0$: système global
- Ai : $i^{ème}$ fils de $A - 0$
- $A\omega j$: $j^{ème}$ fils de $A\omega$



Remarques

L'activité est désignée par un verbe tandis que les données d'entrée, de contrôle et de sortie sont identifiées par un nom

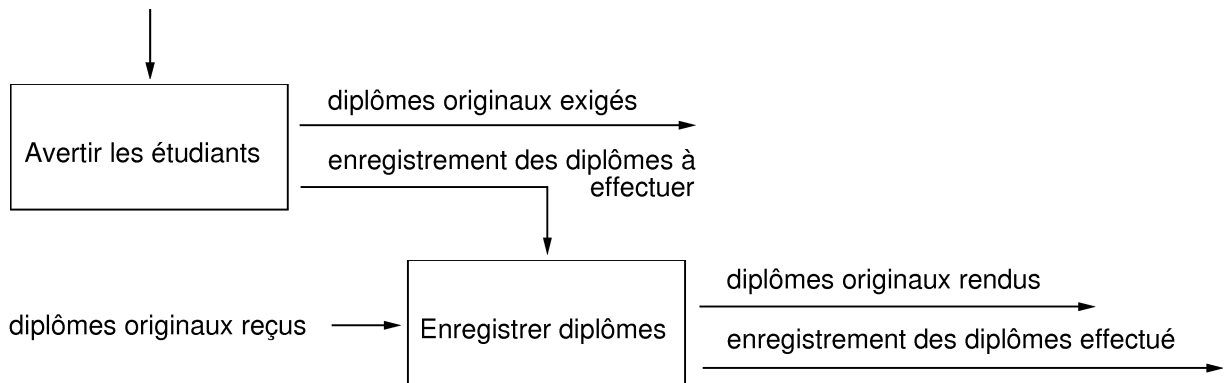
Les mécanismes précisent comment s'effectue l'activité ou encore qui la réalise

Il est possible de préciser les conditions d'activation d'une activité : les pré-conditions indiquent les circonstances d'activation et les post-conditions renseignent sur les effets en résultant ; on en déduit l'ordre dans lequel les contraintes sur les données sont satisfaites

Exemple

La gestion de l'enregistrement des diplômes de tous les étudiants

décision d'enregistrer les diplômes



Objectif

Regroupement des aspects du système tributaires du temps et des modifications par une description du schéma d'activités du système complet, c'est à dire de l'interaction entre le système et l'extérieur ou entre les objets du système

Description du contrôle, c'est à dire de l'ordonnancement des opérations à exécuter en réaction à des événements

Description des événements émis et reçus par les objets

Description des états pris par les objets

Remarque

Modèle de la méthode OMT

Composants

Un diagramme de suivi d'événements pour chaque scénario

Un ensemble de diagrammes de flots d'événements entre objets

Un ensemble de diagrammes de flots d'événements entre classes

Un diagramme d'états pour chaque classe ayant un comportement dynamique important

Le modèle dynamique consiste essentiellement en une collection de diagrammes d'états qui interagissent les uns les autres par l'intermédiaire d'événements partagés

Objectif

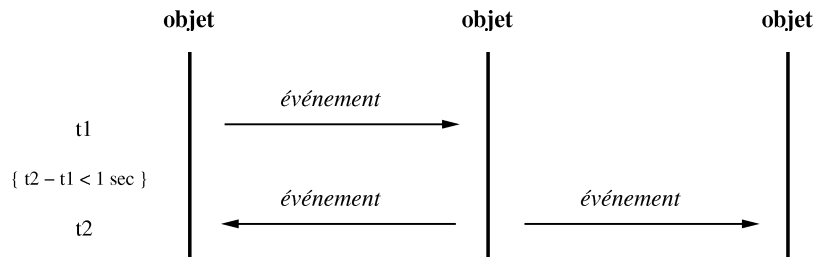
Description de l'ordonnancement des événements entre objets, pour un scénario

Remarque

Modèle composant en partie le modèle dynamique de la méthode OMT

Représentation graphique - Définitions

Diagramme de suivi d'événements



Graphe temporel orienté où les sommets correspondent aux objets et où les arcs correspondent aux événements, comportant éventuellement des contraintes (notamment temporelles)

Objet

Instance d'une classe, c'est à dire entité qui a un sens propre dans le système d'information

Événement

Circonstance pour laquelle un objet est susceptible de réagir

Contrainte

Fonction totale (une valeur est complètement précisée par une autre) ou partielle (valeur bornée par une autre, mais non complètement précisée) faisant apparaître la relation entre deux objets à un même moment

Objectif

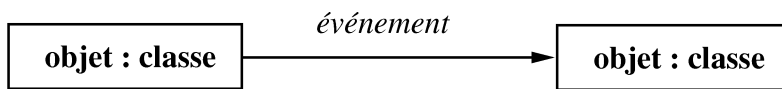
Description de l'ensemble des événements intervenant entre les objets, pour tous les scénarios

Remarque

Modèle composant en partie le modèle dynamique de la méthode OMT

Représentation graphique - Définitions

Diagramme de flots d'événements entre objets



Graphe orienté où les sommets correspondent aux objets et où les arcs correspondent aux événements

Objet

Instance d'une classe, c'est à dire entité qui a un sens propre dans le système d'information

Événement

Circonstance pour laquelle un objet est susceptible de réagir

Contrainte

Fonction totale (une valeur est complètement précisée par une autre) ou partielle (valeur bornée par une autre, mais non complètement précisée) faisant apparaître la relation entre deux objets à un même moment

Objectif

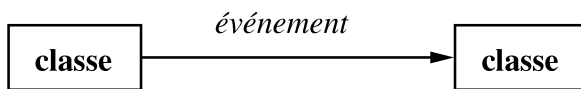
Description de l'ensemble des événements intervenant entre les classes

Remarque

Modèle composant en partie le modèle dynamique de la méthode OMT

Représentation graphique - Définitions

Diagramme de flots d'événements entre classes



Graphe orienté où les sommets correspondent aux classes et où les arcs correspondent aux événements

Classe

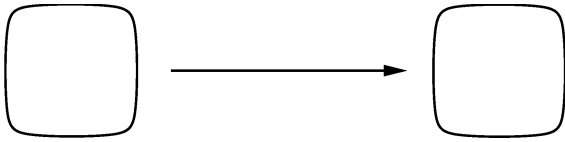
Groupe d'objets qui ont une même sémantique et des caractéristiques communes

Événement

Circonstance pour laquelle une classe est susceptible de réagir

Contrainte

Fonction totale (une valeur est complètement précisée par une autre) ou partielle (valeur bornée par une autre, mais non complètement précisée) faisant apparaître la relation entre deux classes d'objets à un même moment



Autre dénomination

Automate

Modèle voisin

Cycle de vie d'un objet

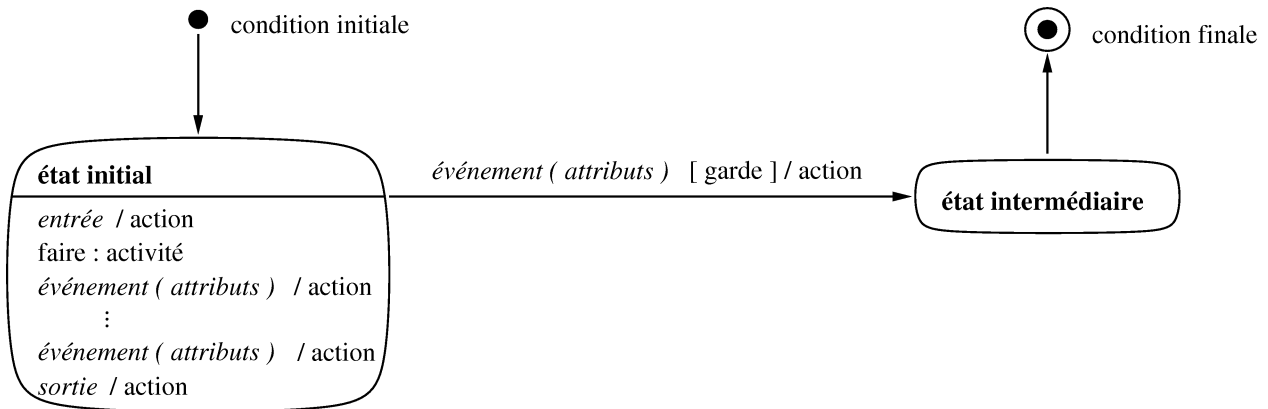
Objectif

Description du comportement des objets d'une classe en montrant l'ensemble des états que peut prendre tout objet d'une classe ainsi que l'ensemble des événements auxquels l'objet réagit

Remarque

Modèle composant en partie (mais principalement) le modèle dynamique de la méthode OMT

Diagramme d'états



Graphe orienté où les sommets correspondent aux états et où les arcs correspondent aux événements

Classe (d'objets)

Groupe d'objets qui ont une même sémantique et des caractéristiques communes

Objet

Instance d'une classe, c'est à dire entité qui a un sens propre dans le système d'information

Classe d'événements

Regroupement d'événements ayant une structure et un comportement communs

Peut posséder des attributs indiquant les informations dont sont porteurs les événements (l'instant où l'événement a lieu est un attribut implicite)

Événement

Circonstance (stimulus externe) qui se produit à un moment donné dans le temps

Part d'un état émetteur vers un état récepteur

Les événements sont concurrents s'ils n'ont pas de lien de causalité

Un événement dans un état correspond à une transition qui reste dans le même état (mais sans exécuter les actions en entrée et en sortie)

État (d'un objet)

Abstraction des valeurs des attributs et des liens d'un objet

Transition

Passage d'un état à un (autre ou même) état, suite à un événement

Tous les paramètres (événement avec ou sans attributs, garde, action) sont facultatifs; par exemple, une transition sans paramètre est automatiquement franchie dès que l'exécution de l'activité de l'état émetteur est terminée amenant ainsi l'objet dans l'état récepteur

Garde (ou condition)

Expression booléenne sur un événement devant être vérifiée pour franchir effectivement la transition

La condition peut porter sur l'état d'un objet

Scénario

Séquence d'événements se déroulant durant une exécution particulière du système

Action

Opération instantanée, ininterrompible, toujours exécutée, associée à un événement

Activité

Opération continue ou nécessitant un certain temps d'exécution, associée à un état, qui commence en entrant dans l'état et qui s'arrête soit lorsqu'elle se finit, soit quand un événement survient

Contrainte

Fonction totale (une valeur est complètement précisée par une autre) ou partielle (valeur bornée par une autre, mais non complètement précisée) faisant apparaître la relation entre deux objets à un même moment

Remarque

L'ordre d'exécution des opérations est le suivant, suite à un événement :

- action de sortie (du sous-état émetteur au super-état),
- action de la transition,
- action d'entrée (du super-état au sous-état récepteur),
- activité,
- si l'activité est finie, attente d'événement

Une commande (pour un client)

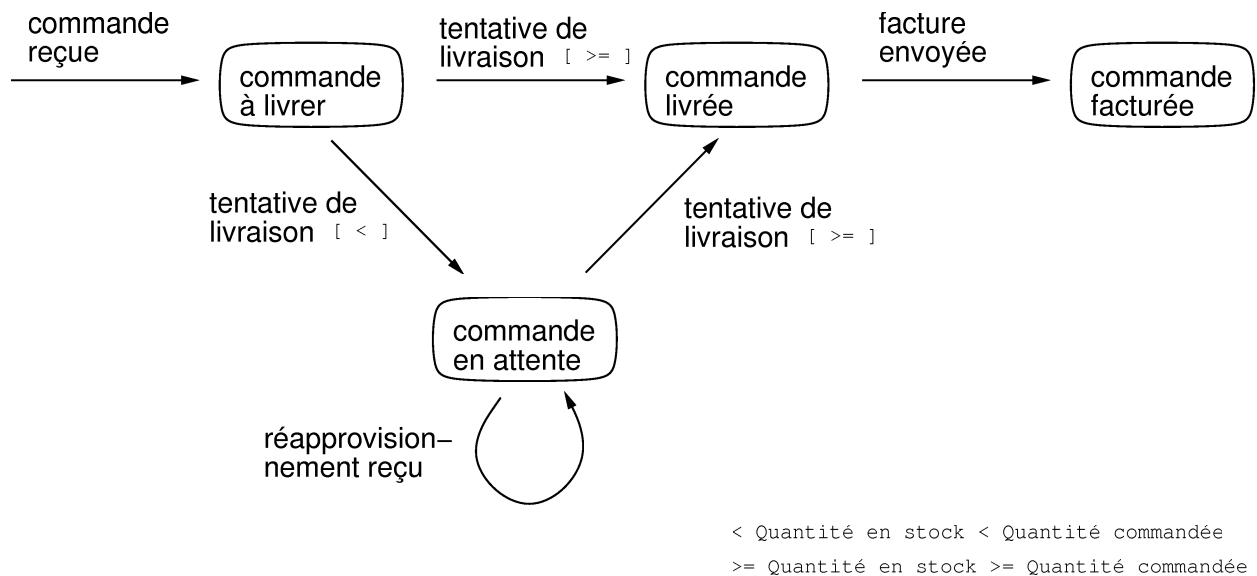
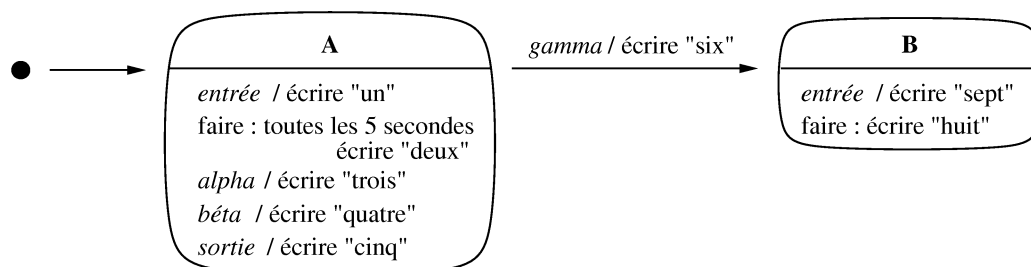


Illustration de l'enchaînement des actions

Considérons le diagramme d'états suivant :



Le scénario suivant engendre les affichages correspondants :

t : l'objet est créé

l'objet entre dans l'état **A** et s'affiche **un**

t + 3 secondes : l'événement *alpha* survient

s'affiche **trois**

t + 5 secondes

s'affiche **deux**

t + 7 secondes : l'événement *delta* survient

rien ne se passe

t + 8 secondes : l'événement *alpha* survient

s'affiche **trois**

t + 10 secondes

s'affiche **deux**

t + 12 secondes : l'événement *gamma* survient

l'objet quitte l'état **A** et s'affiche **cinq**,

l'objet franchit la transition allant de l'état **A** vers l'état **B** et s'affiche **six**,

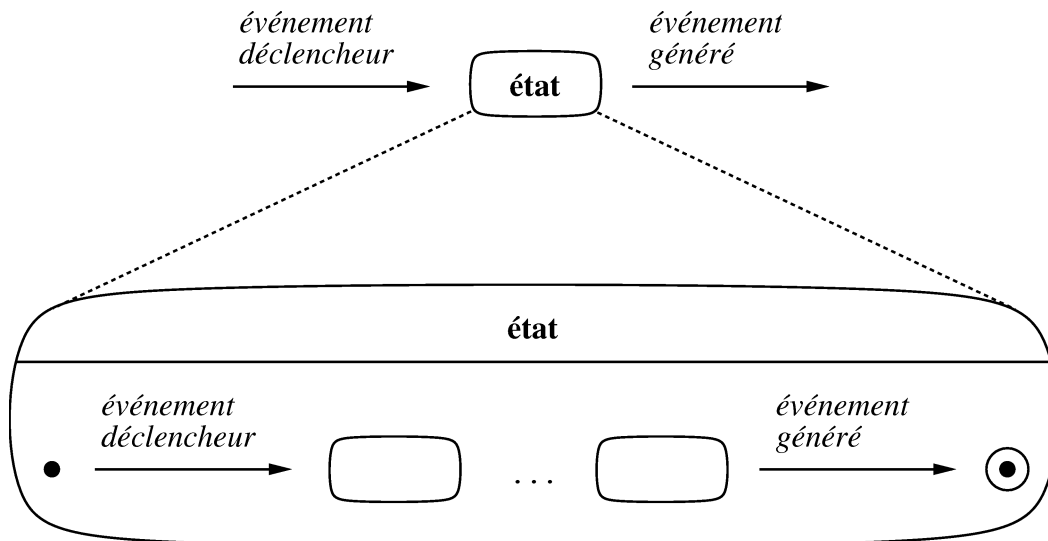
l'objet entre dans l'état **B** et s'affichent **sept** puis **huit**

Diagrammes d'états structurés : objectifs

Structuration par généralisation (description à un haut niveau puis par affinages successifs, hiérarchie de généralisation avec héritage de structures et de comportements communs) et par agrégation (décomposition d'un état en composants ayant peu d'interactions)

Imbrication d'états

Un état peut être étendu en un diagramme d'états de plus bas niveau



Imbrication d'événements

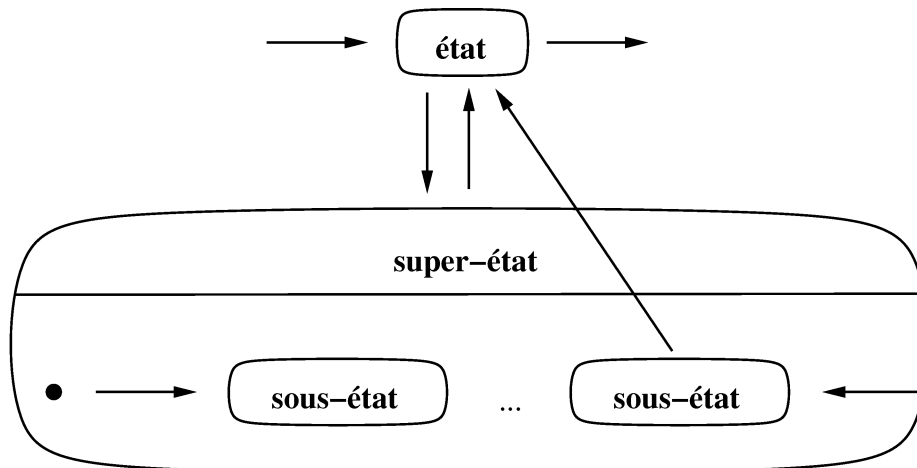
Un événement peut être étendu dans des diagrammes d'états de plus bas niveau

Imbrication d'activités

Une activité d'un état peut être étendue en un diagramme d'états de plus bas niveau (où chaque état représente un pas de l'activité)

Généralisation d'états : super-état

Chaque transition ou opération du super-état est appliquée à tous ses sous-états (sauf s'il y a réécriture de la transition pour un sous-état) : les transitions du super-état sont héritées à tous ses sous-états



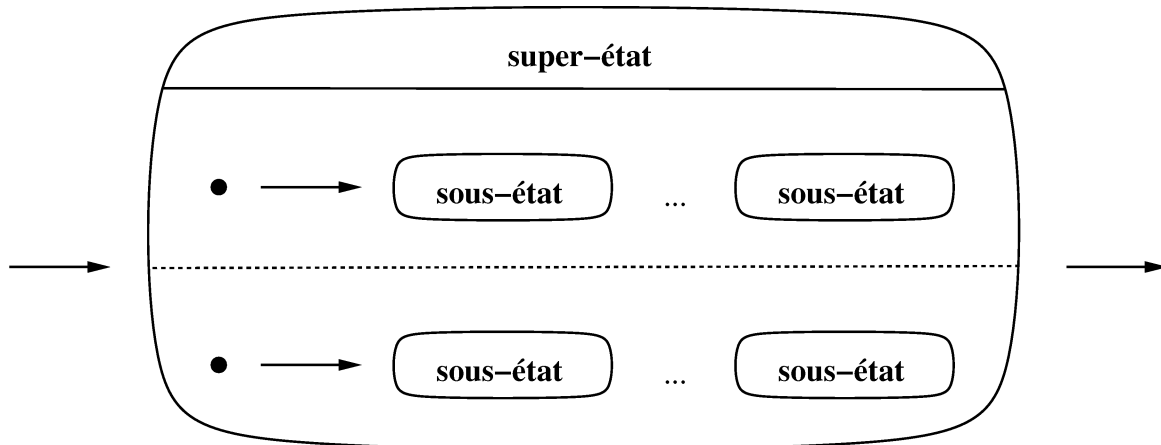
Généralisation d'événements

Organisation des événements en une hiérarchie de généralisation avec héritage des attributs d'événements (en utilisant le formalisme du modèle objet)

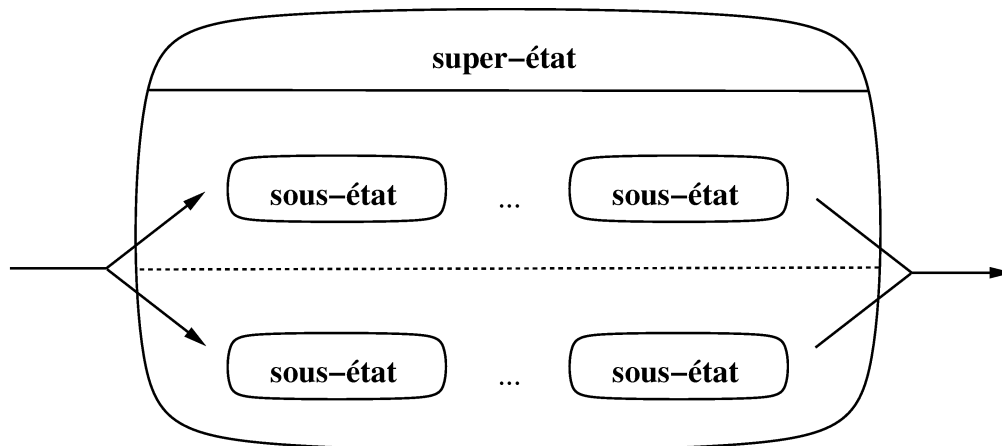
États concurrents

Concurrence à l'intérieur d'un objet quand il est partitionnable en sous-ensembles d'attributs ou de liens, chacun d'eux ayant son propre sous-diagramme d'états; l'état de l'objet englobe un état pour chacun

Concurrence d'agrégation lorsque l'objet est un assemblage (agrégat de composants dans le modèle objet) de sorte que le diagramme d'états de l'agrégat correspond aux états combinés des diagrammes d'états de chaque composant

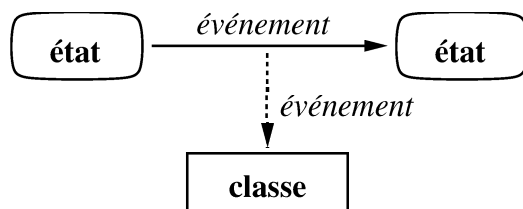


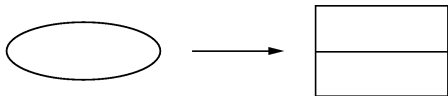
Décomposition et synchronisation du contrôle à l'entrée et en sortie des activités concurrentes



Échange d'événements

Soit un objet peut exécuter une action d'envoi d'un événement à un (ou plusieurs) autre(s) objet(s), soit une transition peut envoyer un événement à un (ou plusieurs) autre(s) objet(s) lorsqu'elle est franchie





Modèle voisin

Diagramme d'états

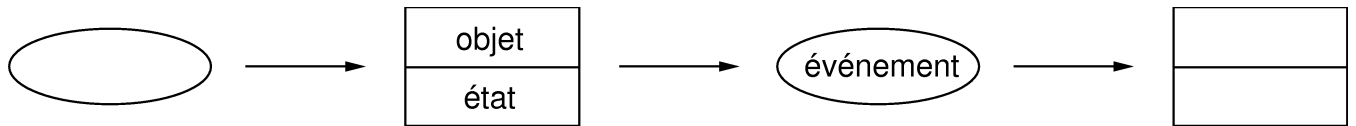
Objectif

Mise en évidence de l'ensemble des états que peut prendre un objet au cours de son cycle de vie ainsi que des événements qui font passer l'objet d'un état à un autre

Remarque

Modèle de MERISE/2

Cycle de vie d'un objet



Graphe orienté où les sommets correspondent aux états et aux événements (les sommets sont de deux types différents) et où les arcs correspondent aux transitions d'un état à un événement ou l'inverse

État

Stade transitoire par lequel passe un objet au cours de son cycle de vie

Exemples : à livrer, en attente, livrée, facturée sont des états de l'objet commande

Événement

Déclenche la transition d'un état à un autre état

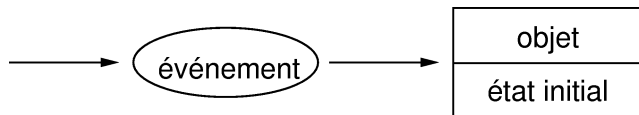
Exemple : la livraison d'une commande

Transition

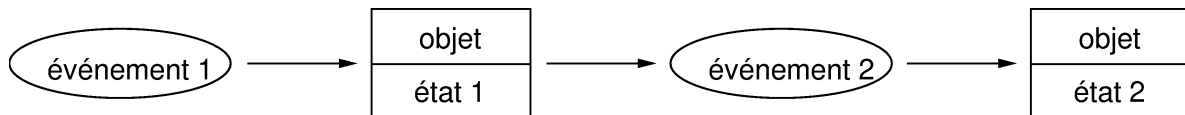
Passage de l'objet d'un état à un autre état

Exemple : suite à la livraison, la commande est livrée

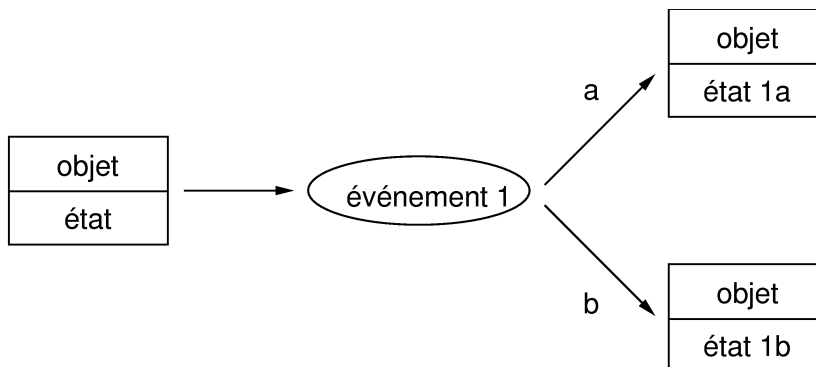
Création



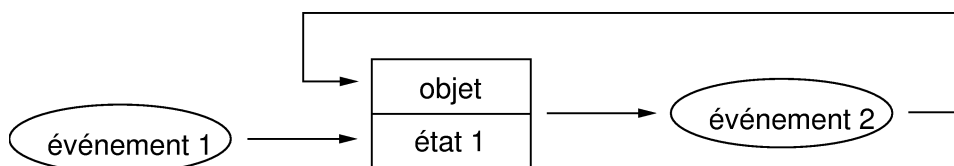
Séquence



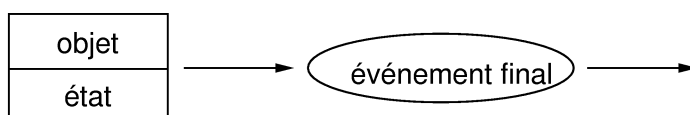
Alternative



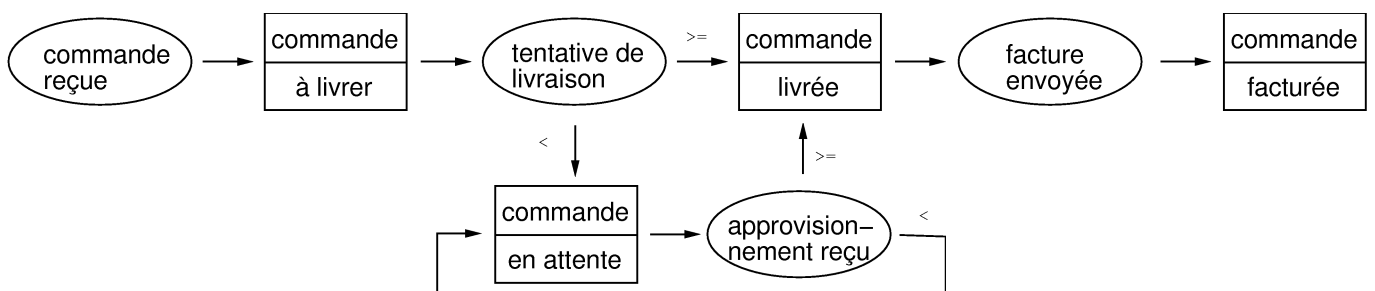
Itération



Suppression



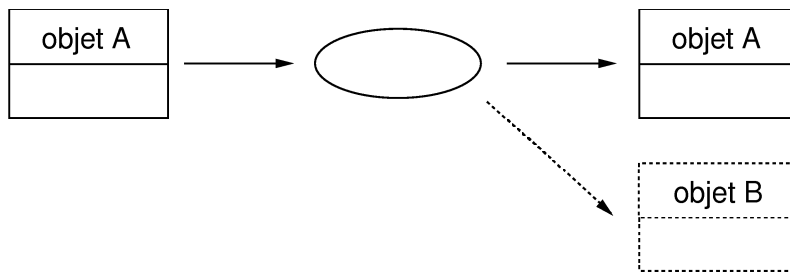
Exemple : une commande



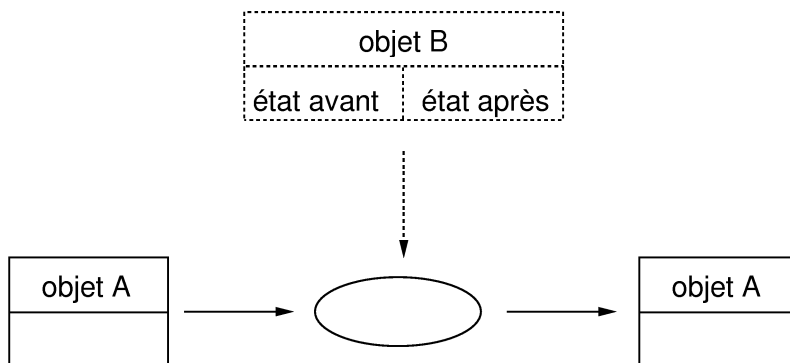
< Quantité en stock < Quantité commandée
 >= Quantité en stock >= Quantité commandée

Connexion entre cycles de vie d'objets

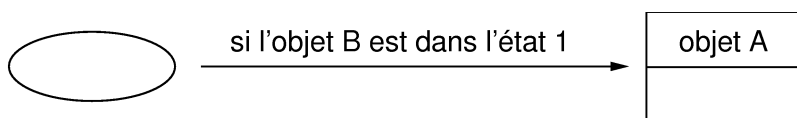
Un même événement déclenche des transitions sur plusieurs objets



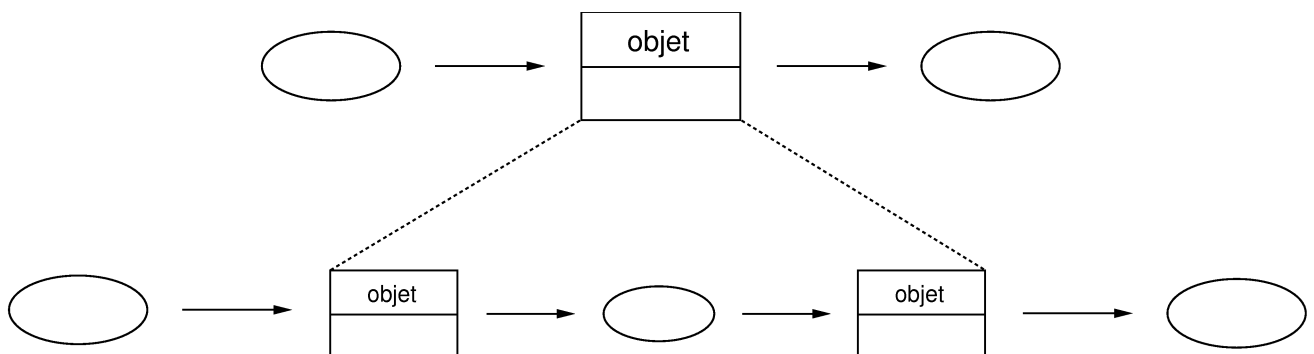
Une transition d'état à état dans un cycle de vie d'un objet constitue un événement interne qui déclenche une transition dans le cycle de vie d'un autre objet



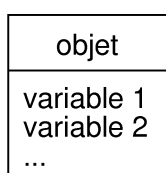
Consultation de l'état d'un autre objet



Utilisation à différents niveaux de détail



L'état d'un objet peut être constitué de plusieurs variables



Objectif

Modèle statique des traitements (niveau logique)

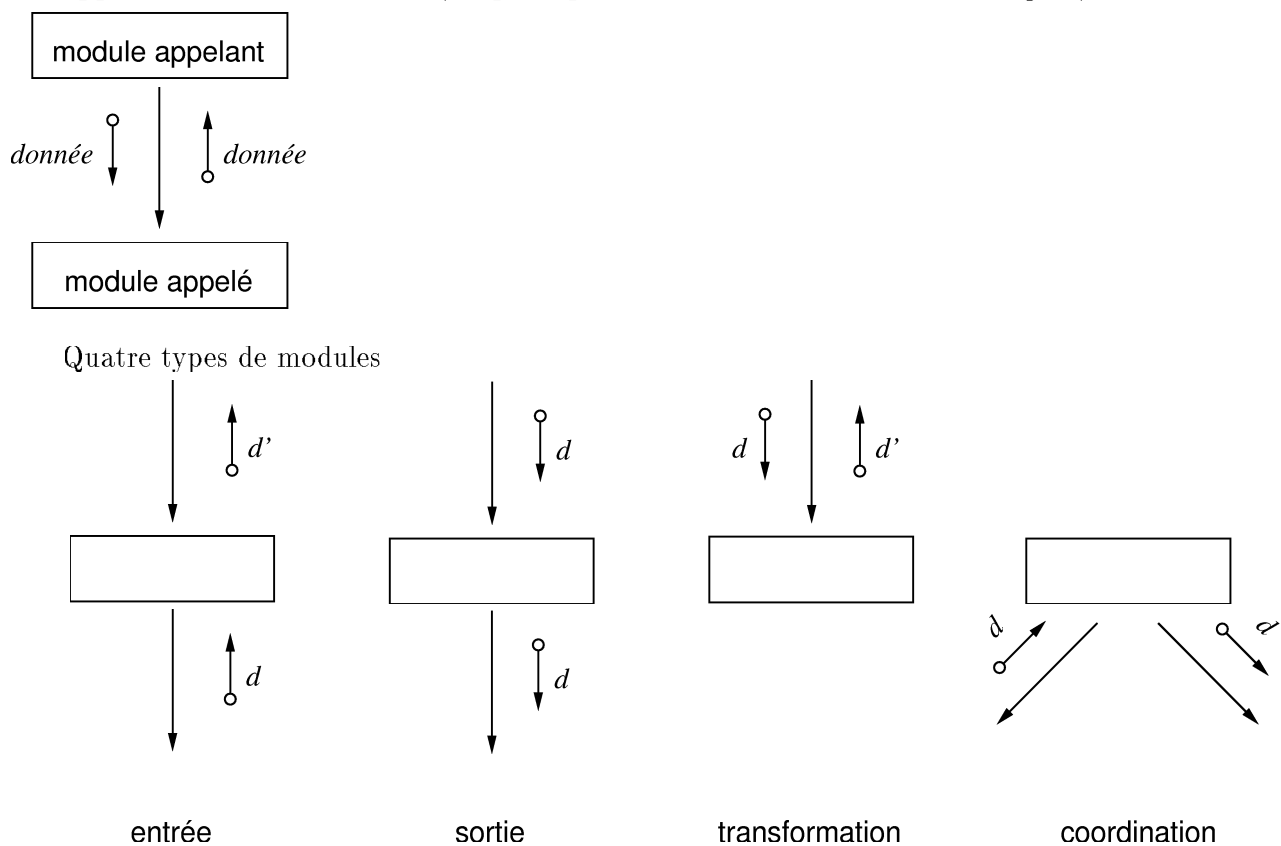
Établir la façon dont les éléments de transformation d'un diagramme de flux peut être réalisée par une hiérarchie de modules (ou unités de programme)

Remarque

Modèle très prisé par les anglo-saxons

Diagramme de structure

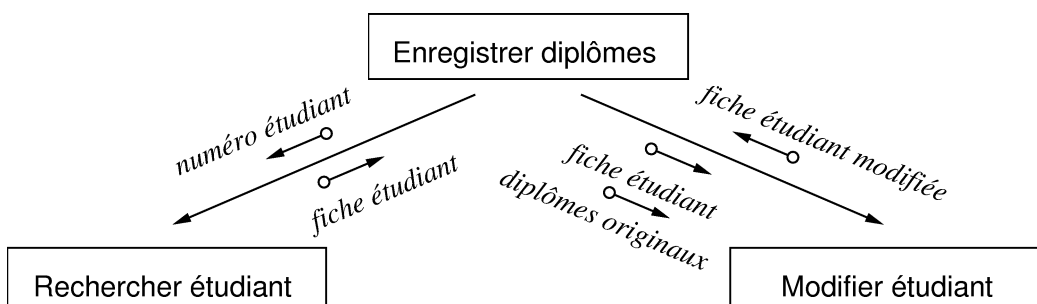
Graphe orienté sans cycle où les sommets correspondent aux modules et où les arcs correspondent aux appels entre deux modules (étiqueté par les éventuelles données échangées)



Ainsi, les entrées du logiciel sont représentées par les entrées des modules n'appelant aucun autre module

Remarque : l'ordre des modules (de gauche à droite) n'a aucune signification

Exemple



Objectif

Modèle statique des traitements

Table de décision

Indique quelles sont les actions à entreprendre (X) lorsqu'un ensemble de conditions est satisfait (O ou $-$)

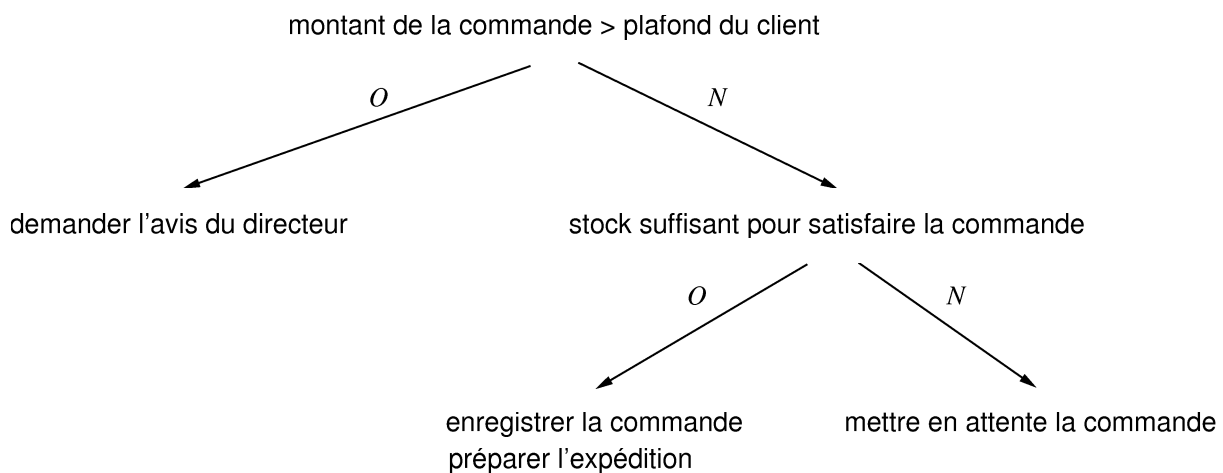
règles de traitement				
conditions			...	← O ou N ou $-$
	⋮			
actions				← X ou <i>espace</i>
	⋮			

Il faut savoir (sans ambiguïté ni redondance) quelles sont les actions à entreprendre pour chacune des $2^{\#conditions}$ conditions possibles

Exemple

montant de la commande > plafond du client	O	N	N
stock suffisant pour satisfaire la commande	$-$	O	N
demander l'avis du directeur	X		
enregistrer la commande		X	
préparer l'expédition		X	
mettre en attente la commande			X

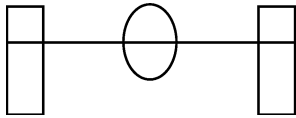
Arbre de décision correspondant :



Algorithme correspondant :

```

si montant de la commande > plafond du client
alors demander l'avis du directeur
sinon si stock suffisant pour satisfaire la commande
    alors debut
        enregistrer la commande
        préparer l'expédition
    fin
sinon mettre en attente la commande
  
```



Autres dénominations

Modèle entité-liaison

Modèle objet-relation

Modèle individuel

(Entity-Relationship model)

Modèles voisins

Modèle organisationnel des données

Modèle logique des données

Modèle relationnel

Modèle navigationnel

Modèle objet

Origine

1976 : CHEN (américain)

1976 : MOULIN, RANDON, SAVOYSKY, SPACCAPIETRA, TARDIEU, TEBOUL (français)

Normalisation

Modèle retenu par l'*International Standard Organization* pour décrire les aspects conceptuels statiques des données dans la conception des systèmes d'information

Remarques

Modèle conceptuel des données de MERISE et d'AXIAL

Outil de dialogue entre informaticiens et utilisateurs

Approche descendante

Entité : individu ou objet concret ou abstrait qui possède une existence intrinsèque et une certaine stabilité permettant de le repérer au cours du temps

Exemples : l'étudiant DURAND, la voiture immatriculée 4040 NT 40, le diplôme DEUG

Association : regroupement d'entités dans lequel chaque entité joue un rôle précis

Exemples : l'étudiant MARTIN possède la voiture immatriculée 4747 LA 47, les étudiants MARTIN et DUPOND ont tous deux obtenu une MIAGe (respectivement en 1982 et 1985)

Propriété : caractéristique d'une entité ou d'une association

Exemples : le département de naissance de l'étudiant DURAND est 33, la couleur de la voiture immatriculée 4747 LA 47 est rouge, l'intitulé complet du diplôme DUT est Diplôme Universitaire de Technologie

Remarque : une propriété est désignée par un **nom** et une **valeur**

Son type peut être simple, structuré ou répétitif

Type de propriétés (ou propriété-type) : nom d'une propriété

Exemples : le département de naissance des étudiants, la couleur des voitures, l'intitulé complet des diplômes, l'année d'obtention des diplômes par les étudiants

Remarque : contraintes d'**unicité** sur les valeurs des propriétés (toute propriété prend une valeur et une seule) et d'**élémentarité** sur les types de propriétés (entier, réel, booléen, caractère, texte, date, ... mais pas tableau, structure, ...)

Type d'entités (ou entité-type) : ensemble d'entités définies par un même ensemble de types de propriétés jouant un rôle identique et représentant une classe naturelle d'objets

Exemples : les types d'entités Etudiants, Voitures et Diplômes

Type d'associations (ou association-type) : ensemble d'associations ayant la même sémantique

Exemples : Posséder entre les types d'entités Etudiants et Voitures, AvoirObtenu entre les types d'entités Etudiants et Diplômes

Remarque : types d'associations binaire ou n -aire (i.e. mettant en jeu deux ou plusieurs types d'entités, de **dimension** 2 ou n), réflexive (i.e. mettant en jeu au moins deux fois le même type d'entités)

Lien : couple type d'entités et type d'associations, dont la signification est donnée par le rôle

Cardinalités : pour un lien donné, nombre minimum (**cardinalité minimale**) et maximum (**cardinalité maximale**) d'occurrences du type d'associations pouvant exister pour une seule occurrence du type d'entités

(i.e. les nombres minimum —au moins— et maximum —au plus— des cardinaux des images de l'application qui fait correspondre à un élément du type d'entités les éléments des autres types d'entités qui lui sont reliés par le type d'associations)

La cardinalité minimale prend pour valeur 0 ou 1 tandis que la cardinalité maximale prend pour valeur 1 ou n (n pour plusieurs)

Exemple : Les cardinalités minimale et maximale du lien entre le type d'entités Etudiants et le type d'associations Posséder sont respectivement 0 et n . En effet, il suffit de choisir les bonnes affirmations parmi les suivantes.

Un étudiant peut ne pas posséder de voiture (cardinalité minimale = 0) ou bien tout étudiant possède au moins une voiture (cardinalité minimale = 1)

Tout étudiant possède au plus une voiture (cardinalité maximale = 1) ou bien un étudiant peut posséder plusieurs voitures (cardinalité maximale = n)

Classification des liens d'un type d'associations binaire

Liens 1 : 1 (un à un) : les deux cardinalités maximales ont pour valeur 1

(i.e. qu'à une occurrence de l'un des deux types d'entités peut correspondre par le type d'associations au plus une occurrence de l'autre type d'entités, et réciproquement)

Liens 1 : n (un à plusieurs) : l'une des deux cardinalités maximales a pour valeur 1 tandis que l'autre a pour valeur n

(i.e. qu'à une occurrence de l'un des deux types d'entités peut correspondre par le type d'associations au plus une occurrence de l'autre type d'entités, mais qu'à une occurrence de celui-ci correspond plusieurs occurrences du premier type d'entités cité)

Liens n : n (plusieurs à plusieurs) : les deux cardinalités maximales ont pour valeur n

(i.e. qu'à une occurrence de l'un des deux types d'entités peut correspondre par le type d'associations plusieurs occurrences de l'autre type d'entités, et réciproquement)

Identifiant : type(s) de propriété dont les valeurs permettent de distinguer toutes les occurrences du type d'entités correspondant

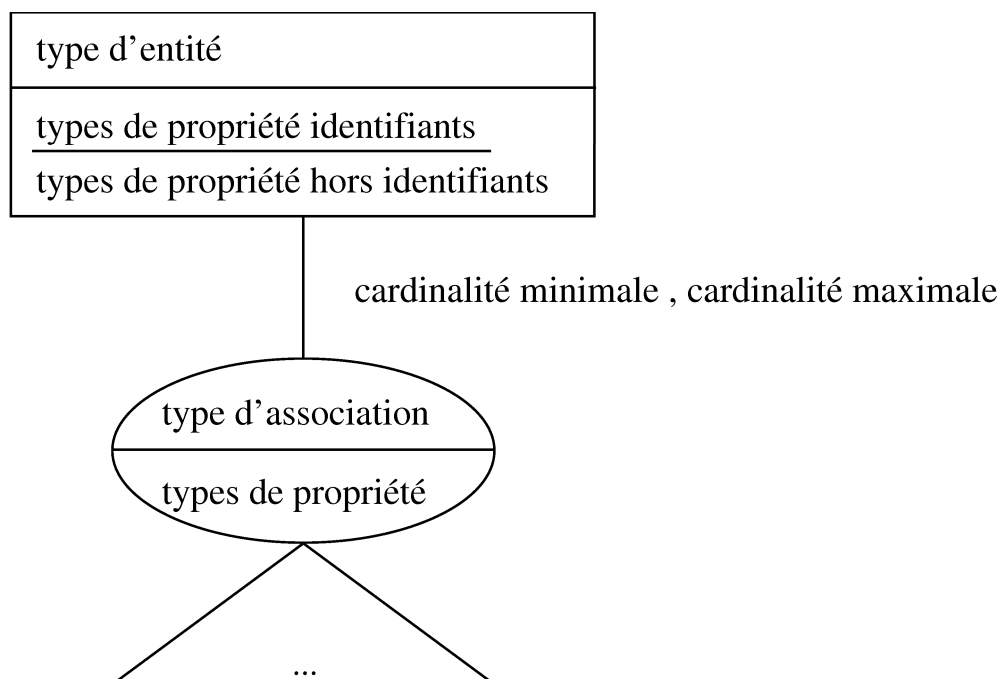
Exemples : numéro INE des étudiants, numéro d'immatriculation complet (chiffres, lettres et département) des voitures

Contre-exemple : nom des étudiants (homonymies)

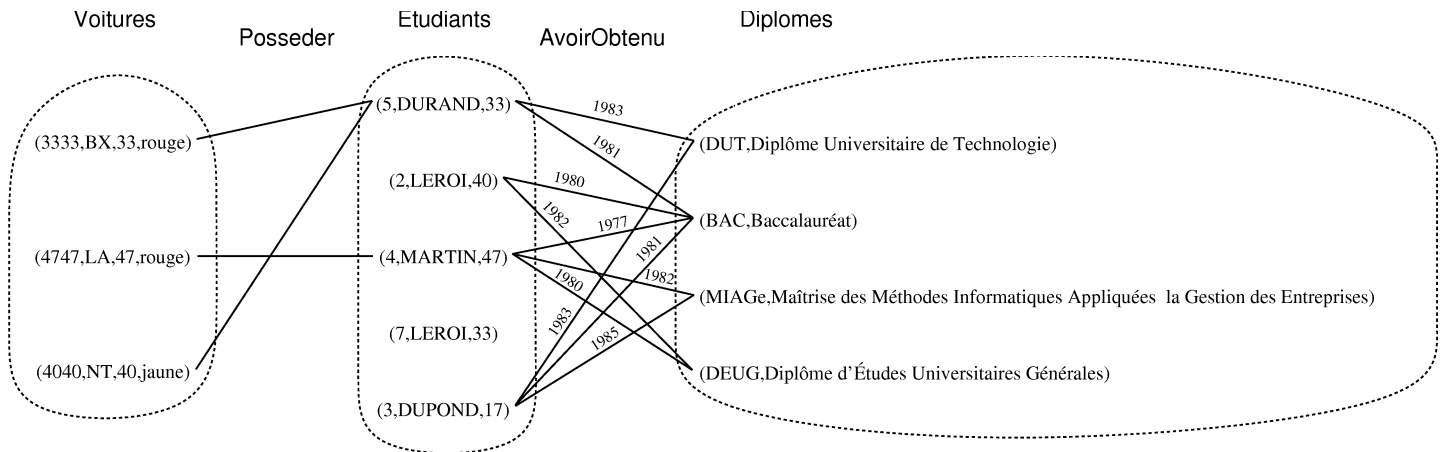
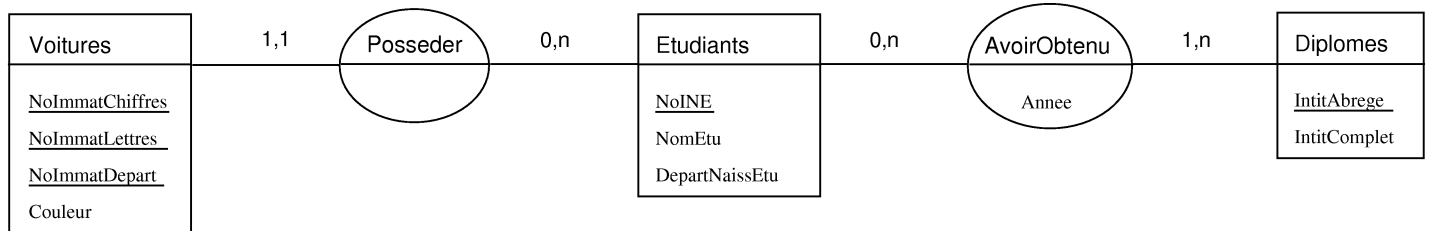
Remarque : l'identifiant d'un type d'associations s'obtient implicitement par composition de tous les identifiants des types d'entités reliés

Le **dictionnaire des données** est une représentation textuelle de toutes les informations d'un système d'information; il se subdivise en trois sous-dictionnaires pour les types d'entités, les types d'associations et les types de propriétés

Le **schéma entité-association** est une représentation graphique du résultat de la modélisation du système d'information



Remarque : par commodité de langage, les termes entité, association, propriété désignent respectivement les termes type d'entités, type d'associations, type de propriétés



Partir de l'univers du discours

Reconnaître les types d'entités et les types d'associations

Un objet est reconnu soit en tant que type d'entités, soit en tant que type d'associations

Chaque type d'associations relie deux ou plusieurs types d'entités

Plusieurs types d'associations différents peuvent relier les mêmes types d'entités, à condition d'avoir des significations différentes

Le schéma est connexe

Les types d'entités et types d'associations sont des classes d'objets de sorte qu'ils possèdent tous plusieurs occurrences

Problème : perception de la réalité!

Les livres suivants sont disposés sur une étagère :

RACINE <i>Phèdre</i>

RACINE <i>Phèdre</i>

ZOLA <i>Germinal</i>

Le bibliothécaire voit trois volumes tandis que le libraire voit deux œuvres

Reconnaître les types de propriétés, les attribuer aux types d'entités ou types d'associations, choisir les identifiants

Ne relever que les types de propriétés pertinents pour l'organisation :

pas de synonyme, polysème, paramètre

pas d'information calculable (ou sinon munie d'une contrainte d'intégrité)

Un type de propriétés est attribué une fois et une seule, soit à un type d'entités, soit à un type d'associations

Pour chaque occurrence de type d'entités et de type d'associations, une et une seule valeur doit être affectée (à terme) à chaque type de propriétés

Chaque type de propriétés d'un type d'associations doit décrire exactement (en nombre et en nature) les types d'entités reliés

Chaque type d'entités doit posséder un identifiant (éventuellement composé de plusieurs types de propriétés) et au moins un autre type de propriétés (hors identifiant)

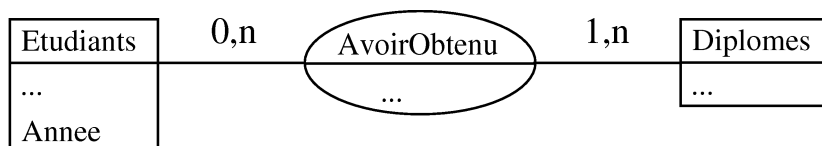
Aucun type d'associations ne doit posséder d'identifiant (il est implicitement donné par la composition des identifiants des types d'entités reliés) mais peut éventuellement posséder des types de propriétés

Pour chaque type d'entités, choix d'un identifiant (minimal) parmi les types de propriétés candidats de préférence celui couramment utilisé dans l'organisation ou à défaut création d'un type de propriétés identifiant

Tous les types de propriétés hors identifiant doivent dépendre pleinement de l'identifiant

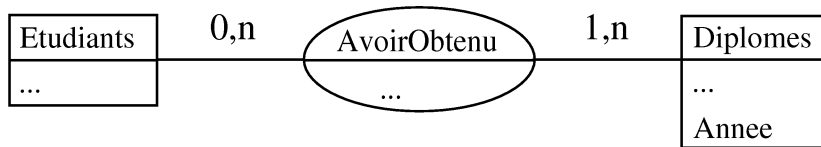
Chaque type de propriétés hors identifiant doit dépendre directement de l'identifiant

Problème : où attribuer un type de propriétés?



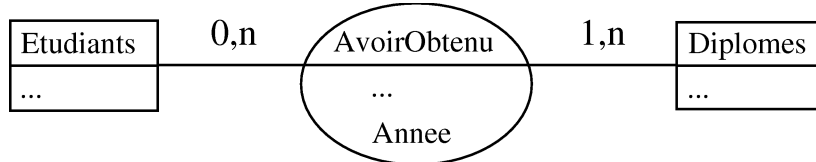
Un étudiant peut avoir obtenu plusieurs diplômes différents deux à deux, et de même un diplôme peut avoir été obtenu par plusieurs étudiants différents deux à deux (liens $n : n$)

Chaque étudiant n'a qu'une année possible pour l'obtention de tous ses diplômes (le type de propriétés 'Annee' appartient au type d'entités 'Etudiants')



Un étudiant peut avoir obtenu plusieurs diplômes différents deux à deux, et de même un diplôme peut avoir été obtenu par plusieurs étudiants différents deux à deux (liens $n : n$)

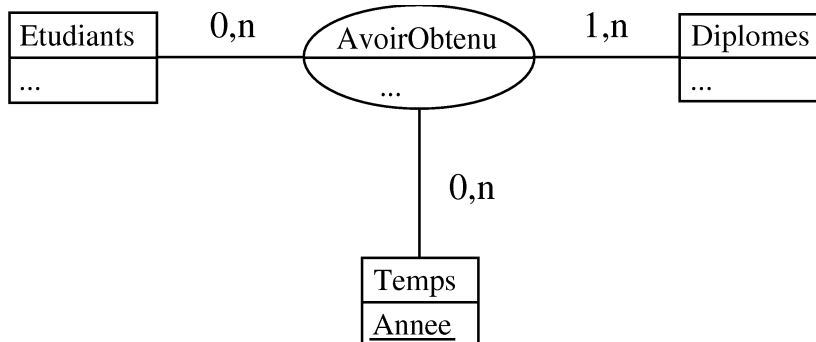
Chaque diplôme n'a qu'une année possible pour avoir été obtenu par tous les étudiants concernés (le type de propriétés Année appartient au type d'entités Diplomes)



Un étudiant peut avoir obtenu plusieurs diplômes différents deux à deux, et de même un diplôme peut avoir été obtenu par plusieurs étudiants différents deux à deux (liens $n : n$)

L'année d'obtention peut varier pour un diplôme donné ou encore pour un étudiant donné (le type de propriétés Année appartient au type d'associations AvoirObtenu)

On parle de représentation synchrone (vision instantanée)



Un étudiant donné peut avoir obtenu un même diplôme à des années différentes

On parle de représentation diachronique (vision historique)

Conclusion : ces quatre solutions sont toutes valables (même si dans ce contexte les deux premières semblent peu réalistes) mais ont des significations différentes; le choix dépendra donc toujours de la réalité à modéliser

Déterminer les cardinalités

Pour chaque lien, se demander si une occurrence du type d'entités est forcément liée par le type d'associations, et combien de liens au maximum sont associés à une occurrence du type d'entités

Valider le schéma (ou quelques règles supplémentaires à vérifier)

Supprimer un type d'associations redondant, c'est à dire ayant la même sémantique que plusieurs autres types d'associations reliant transitivement par des liens 1,1 les mêmes types d'entités

Selon certains concepteurs, décomposer les types d'associations de dimension supérieure ou égale à trois en autant de types d'associations binaires que nécessaire

Compléter le schéma par des contraintes d'intégrité

Ne pas confondre les types d'entités et les acteurs du système d'information

Ne pas confondre les types d'associations et les traitements

Représentation graphique simple et standard

Assure une communication efficace entre :

- les participants à la conception (utilisateurs et concepteurs)

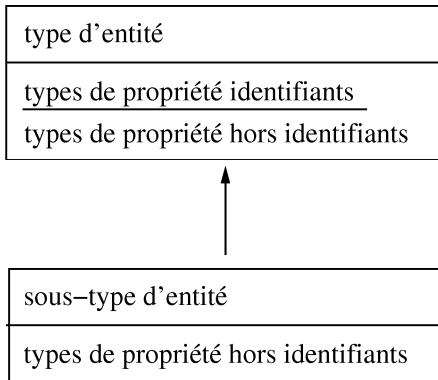
- les participants à la mise en œuvre de la solution informatisée (informaticiens)

Conception facile (il est rapide de définir une solution possible)

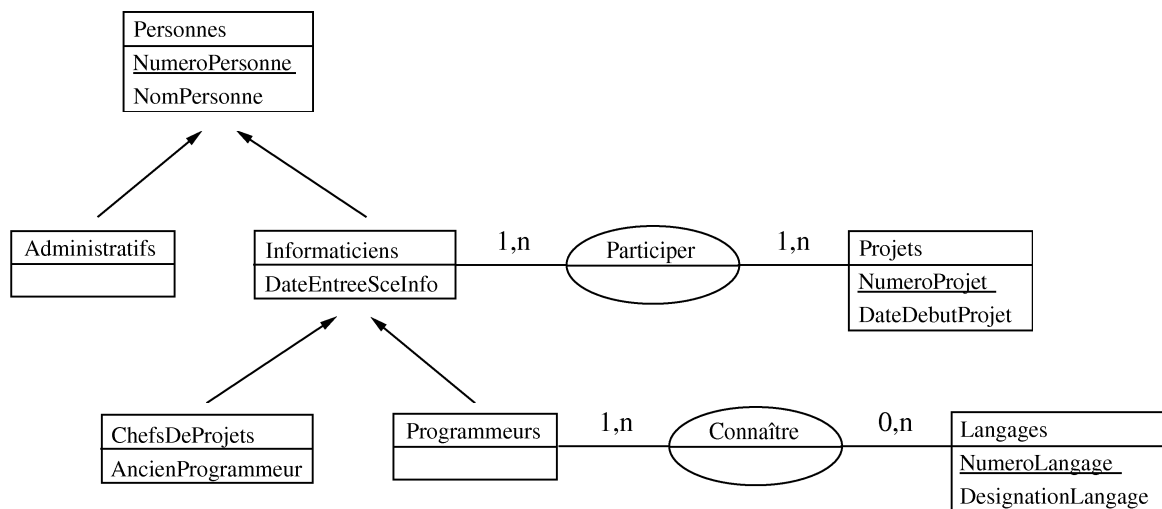
mais délicate (choix difficiles pour exprimer correctement la réalité)

Sous-type d'entités

Sous-ensemble d'occurrences d'un type d'entités qui sont dotées de types d'associations et/ou de types de propriétés spécifiques (en plus des types de propriétés héritées du type d'entités concerné)
On parle de type d'entités générique ou sur-type (d'entité) pour le type d'entités et de type d'entités spécialisé ou sous-type (d'entité) pour le sous-type d'entités



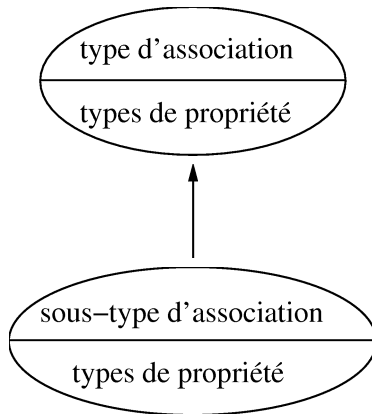
Exemple :



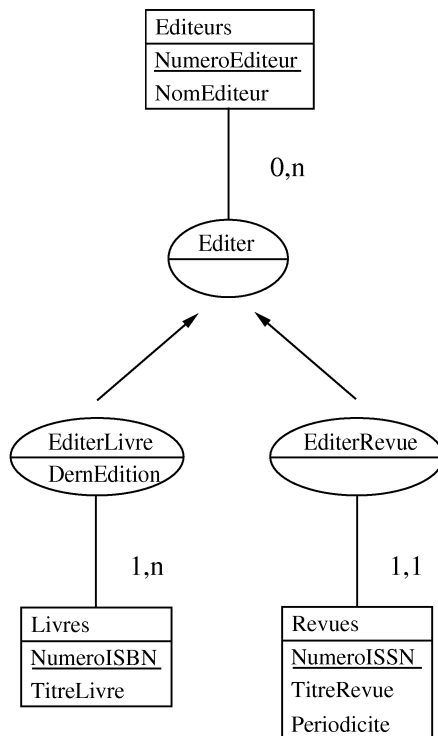
Sous-type d'associations

Sous-ensemble d'occurrences d'un type d'associations qui sont dotées de types de propriétés et/ou de cardinalités spécifiques (en plus des types de propriétés héritées du type d'associations concerné)

On parle de type d'associations générique pour le type d'associations et de type d'associations spécialisée pour le sous-type d'associations



Exemple :



Un livre peut être édité par plusieurs éditeurs tandis qu'une revue n'est éditée que par un et un seul éditeur

Contraintes s'appliquant aux sous-types d'entités

Partition

Toute occurrence du type d'entités appartient à une et une seule occurrence du sous-type d'entités

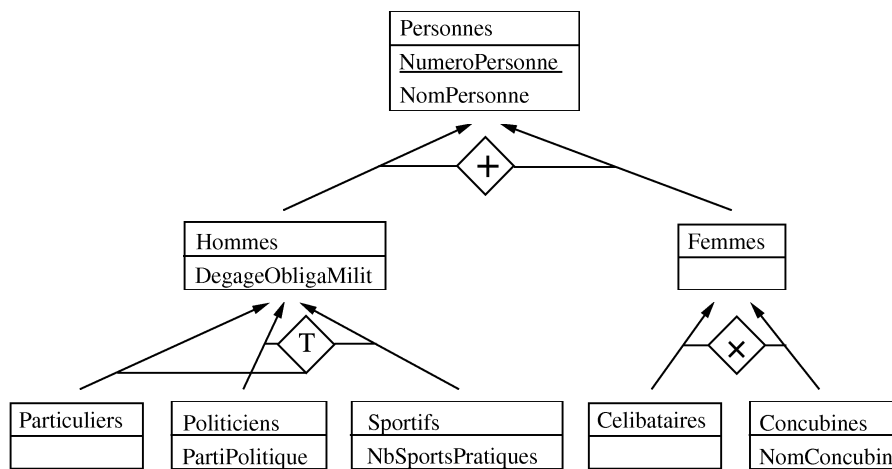
Totalité

Toute occurrence du type d'entités appartient à au moins une occurrence du sous-type d'entités

Exclusion

Toute occurrence du type d'entités appartient à au plus une occurrence du sous-type d'entités

Exemple :



: une personne est soit un homme, soit une femme



: une homme peut être à la fois un particulier, un politicien et un sportif, ou encore appartenir à deux des trois catégories; de plus, une personne appartient à au moins l'une des trois catégories



: une femme ne peut pas être à la fois célibataire et concubine, mais peut n'être ni l'une ni l'autre (mariée par exemple)

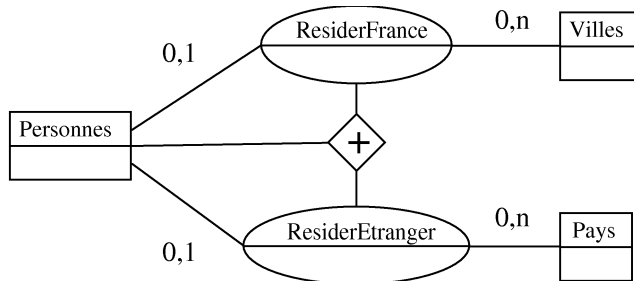
Contraintes s'appliquant aux types d'associations

Partition



Toute occurrence du(des) type(s) d'entité correspond à une et une seule des occurrences des types d'associations

Exemple :



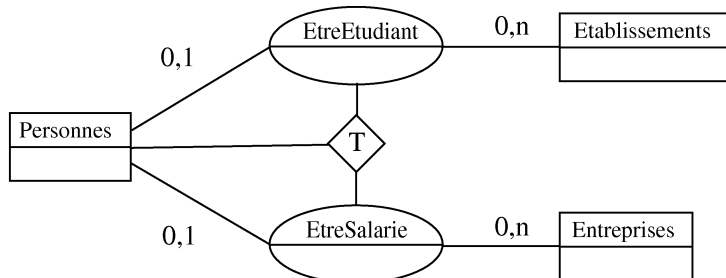
Chaque personne réside soit en France, soit à l'étranger

Totalité



Toute occurrence du(des) type(s) d'entité correspond à au moins une des occurrences des types d'associations

Exemple :



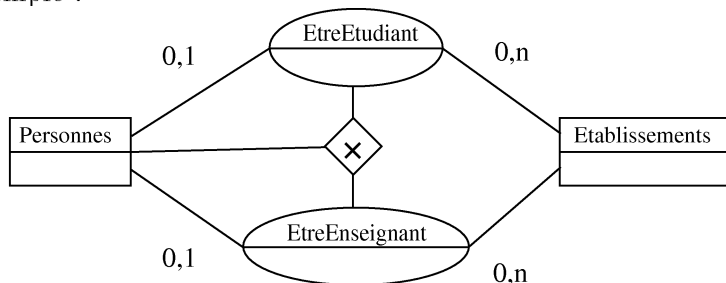
Chaque personne est soit étudiant dans un établissement, soit salarié dans une entreprise, ou les deux à la fois

Exclusion



Toute occurrence du(des) type(s) d'entité correspond à au plus une des occurrences des types d'associations

Exemple :



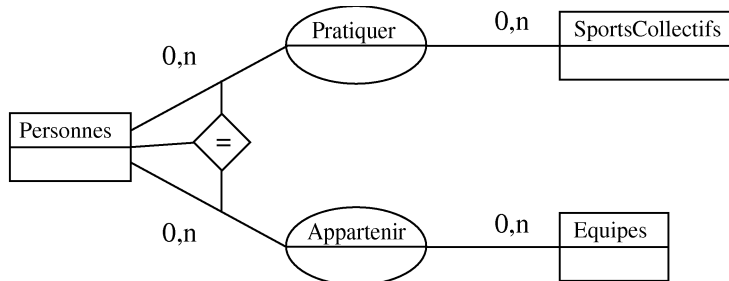
Une personne ne peut pas être à la fois étudiant et enseignant

Contraintes s'appliquant aux types d'associations (suite)

Égalité 

Les occurrences des types d'associations sont exactement les mêmes relativement au(x) type(s) d'entité impliqué(s)

Exemple :

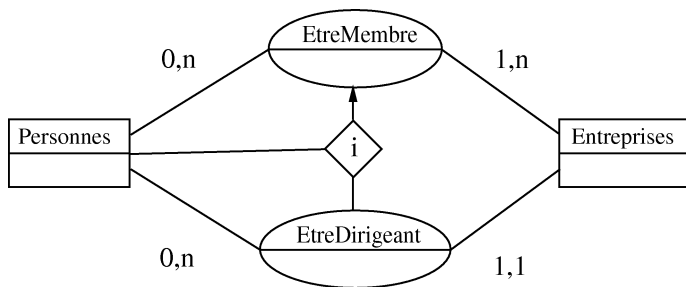


Une personne pratique un sport collectif si et seulement si elle appartient à une équipe

Inclusion 

Les occurrences du type d'associations pointé font toutes parties des occurrences du(des) autre(s) type(s) d'association relativement au(x) type(s) d'entité impliqué(s)

Exemple :

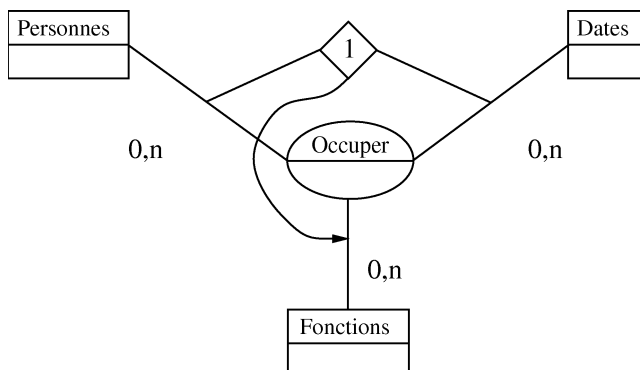


Le dirigeant d'une entreprise fait partie des membres de cette entreprise

Unicité 

Les occurrences du type d'associations pointé sont uniques relativement au(x) type(s) d'entité reliés par l(es) autre(s) lien(s)

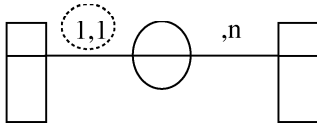
Exemple :



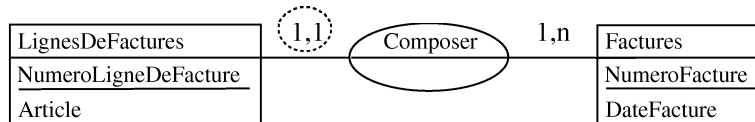
Une personne à une date donnée ne peut occuper qu'une et une seule fonction

Héritage d'identifiant

Un type d'entités a pour identifiant son propre identifiant composé avec ceux hérités des autres types d'entités reliés par un lien identifiant



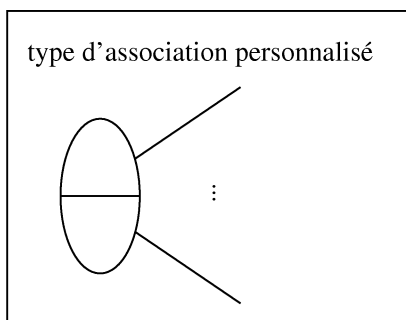
Exemple :



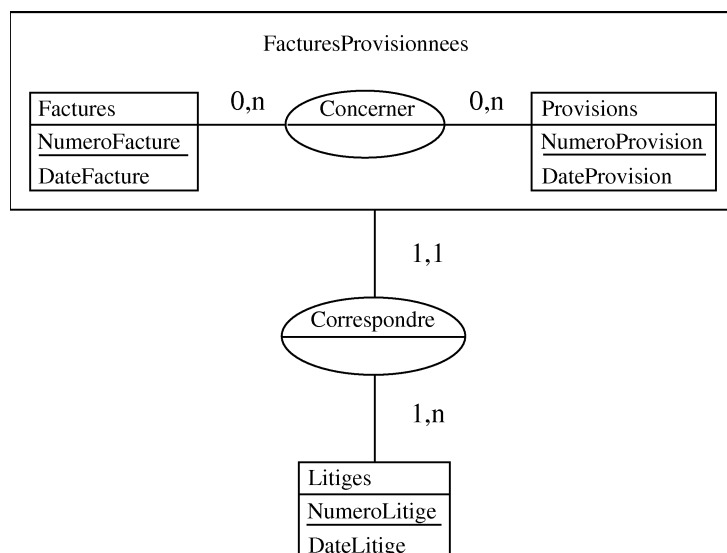
Ainsi, l'identifiant du type d'entités LignesDeFactures est le couple d'attributs (NumeroFacture , NumeroLigneDeFacture)

Type d'associations personnalisé

Regroupement d'un sous-schéma (de types d'entités et de types d'associations reliés) en l'équivalent d'un type d'entités



Exemple :



Un litige correspond à des factures provisionnées

Contraintes s'appliquant aux types de propriétés

Intervalle de valeurs d'un type de propriétés

Exemple : plus de 1901

Énumération des valeurs d'un type de propriétés

Exemple : roi, reine, tour, cavalier, fou, pion

Contraintes entre valeurs de plusieurs types de propriétés

Exemple : date de retour d'un questionnaire postérieure ou égale à sa date d'envoi

Format d'un type de propriétés

Exemple : le code produit est composé de quatre caractères, les trois premiers sont des lettres mnémoniques et le dernier est un chiffre séquentiel

...

Type de propriétés agrégée

Un type de propriétés est composé à partir d'autres types de propriétés

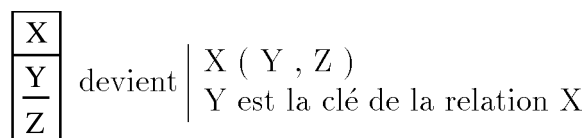
Exemple : le type de propriétés Adresse est en fait une agrégation des types de propriétés Numéro, Voie, CodePostal, Ville

Type de propriétés multi-valuée

Pour une occurrence du type d'entités ou du type d'associations, la valeur du type de propriétés est une liste de valeurs appartenant chacune au même domaine

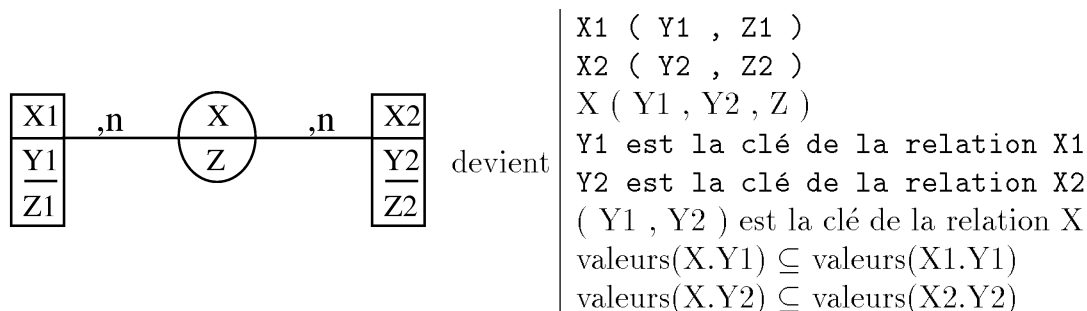
Exemple : les prénoms de chaque étudiant

Transformation de tous les types d'entités en relations



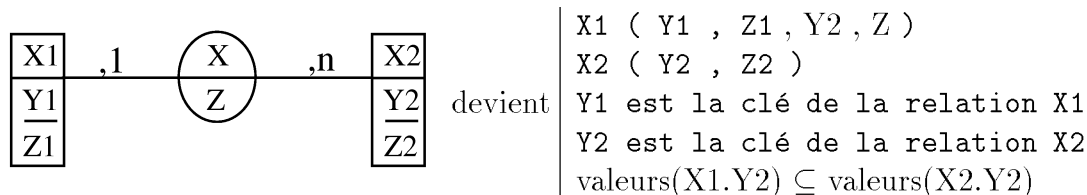
Le type d'entités devient la relation, les types de propriétés identifiants deviennent les attributs de la clé, les types de propriétés hors identifiants deviennent les attributs hors clé

Transformation de tous les types d'associations de liens $n : n$ en relations



Le type d'associations devient la nouvelle relation dont la clé est composée des types de propriétés identifiants des types d'entités associés et qui récupère comme attributs hors clé les types de propriétés du type d'associations; de plus, des contraintes d'intégrité référentielles sont à considérer pour chaque attribut composant la clé de la nouvelle relation relativement aux clés des relations issues des types d'entités

Transformation de tous les types d'associations de liens $1 : n$ en attributs supplémentaires



Les types de propriétés identifiants du type d'entités impliqué dans le lien $,n$ constituent une clé étrangère pour la relation issue du type d'entités impliqué dans le lien $,1$ (et engendre donc une contrainte d'intégrité référentielle); les types de propriétés du type d'associations suivent le même mouvement

Cet algorithme procède en deux passes, l'une pour les types d'entités et l'autre pour les types d'associations; pour les transformations des types d'associations, le résultat obtenu par la première passe est ainsi écrit : **transformations des types d'entités**

$\text{valeurs}(A.B) \subseteq \text{valeurs}(C.D)$ signifie que l'ensemble des valeurs que peut prendre l'attribut B de la relation A est inclu ou égal à l'ensemble des valeurs que prend l'attribut D de la relation C, et ceci à tout moment : il s'agit d'une contrainte d'intégrité référentielle à respecter

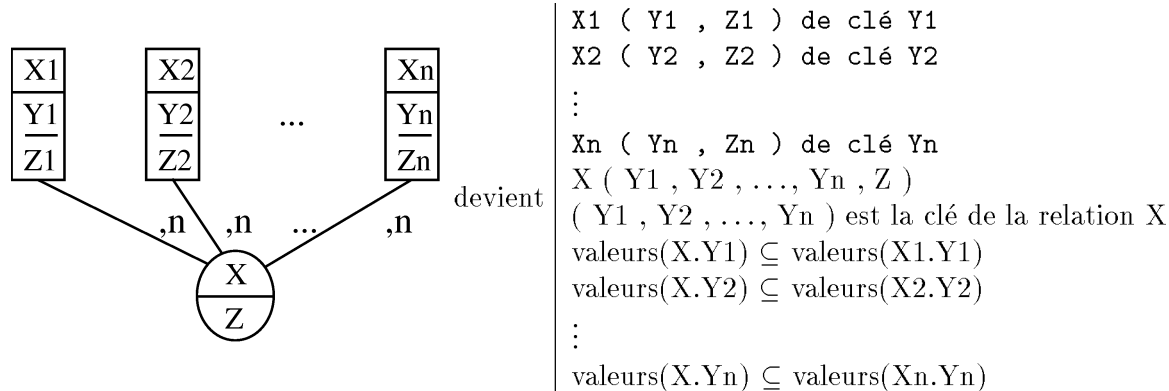
Les contraintes d'intégrité suivantes s'appliquent :

unicité de chaque clé (primaire)

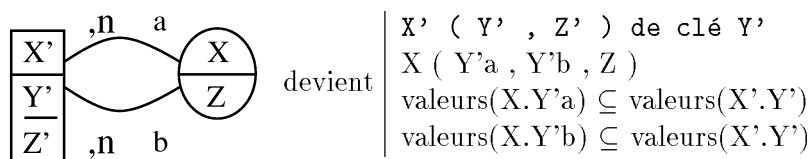
pas d'indétermination pour chaque attribut composant la clé (primaire)

pas d'indétermination pour chaque attribut composant la clé étrangère si le lien correspondant admet la valeur 1 pour sa cardinalité minimale

Type d'associations n -aire (avec $n > 2$) de liens $n : n : \dots : n$

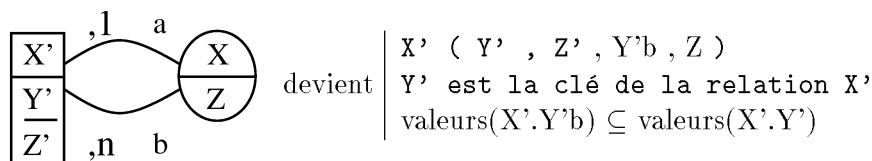


Type d'associations binaire réflexive de liens $n : n$



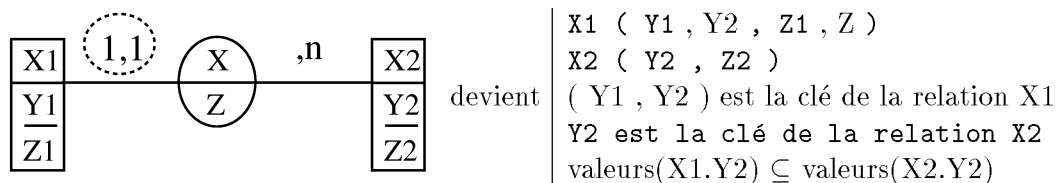
où a et b sont des rôles

Type d'associations binaire réflexive de liens $1 : n$

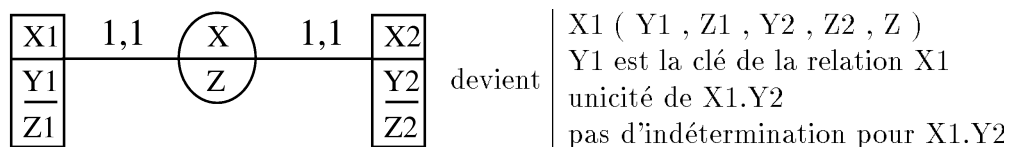


où a et b sont des rôles

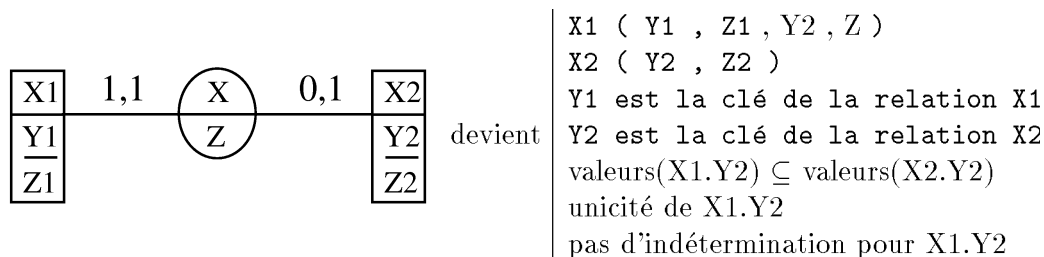
Type d'associations binaire avec un lien identifiant



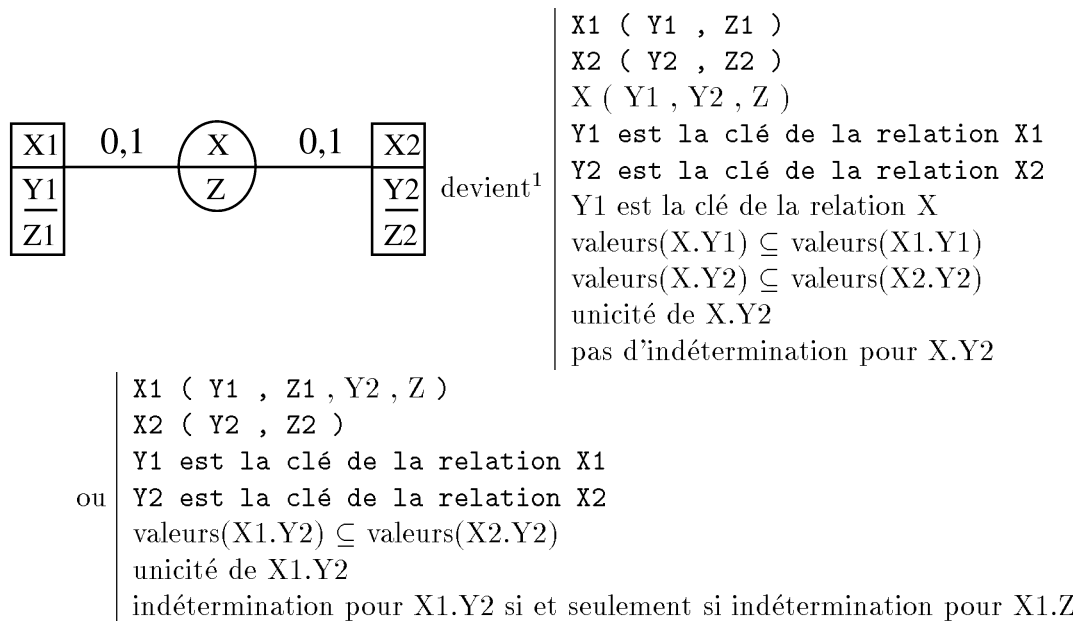
Type d'associations binaire de liens $1,1 : 1,1$



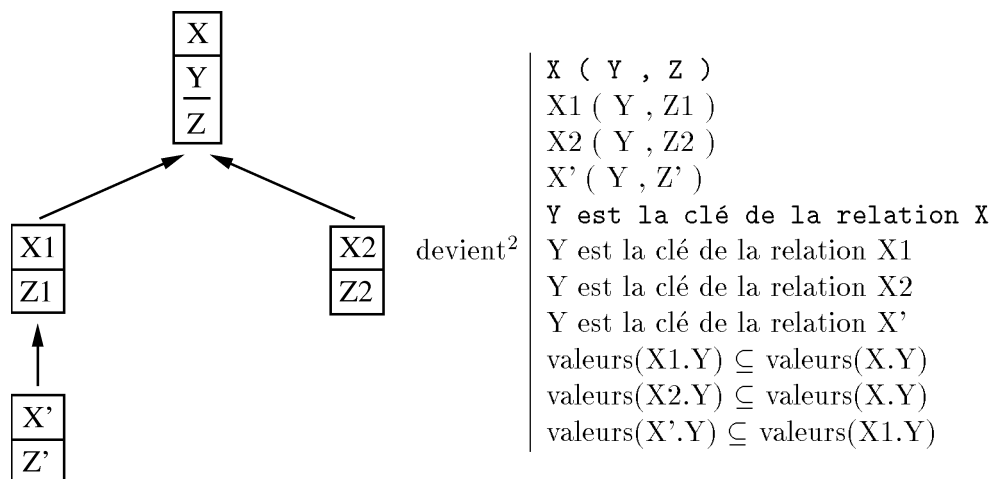
Type d'associations binaire de liens $1,1 : 0,1$



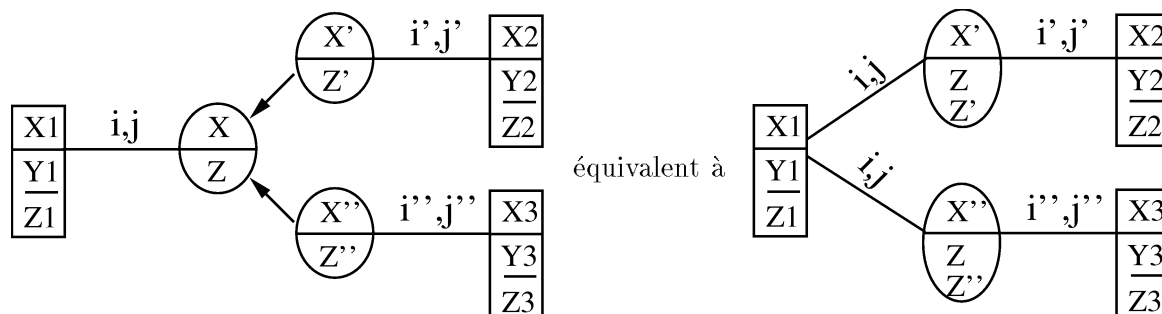
Type d'associations binaire de liens 0,1 : 0,1



Sous-type d'entités



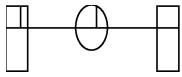
Sous-type d'associations



¹ : la seconde solution n'est à retenir que si le nombre d'entités de X1 non liées à des entités de X2 par l'association X est très faible

² discutable! (à confronter avec AMC*Designor par exemple)

Exercice : trouver l'algorithme inverse



Modèles voisins

Modèle entité-association

Modèle logique des données

Modèle relationnel

Modèle navigationnel

Modèle objet

Objectif

Représentation des données conceptuelles et organisationnelles

Remarque

Modèle organisationnel des données de la méthode MERISE/2

Composants

1 modèle organisationnel des données global

Différentes vues du modèle organisationnel des données pour un type de site ou un type de poste

Formalisme

Reprend celui du modèle entité-association étendu, complété de contraintes organisationnelles

Entité - Type d'entités - Association - Type d'associations - Lien - Cardinalités - Propriété - Type de propriétés - Identifiant

(cf. le modèle entité-association)

Type d'acteur

Regroupement d'acteurs exerçant des activités identiques

Type de site

Regroupement géographique et/ou fonctionnel de types d'acteurs

Type de poste

Rapprochement entre un type d'acteur et un type de site

Nature

Précisions sur la répartition homme/machine et sur le support d'historisation

Informatisé (**I**), manuel (**M**), historisation sur support informatique (**H**), historisation sur support manuel (**HM**)

Type d'accès

Restriction sur un type d'entités, un type d'associations ou un type de propriétés

Création (**C**), interrogation (**I**), modification (**M**), suppression (**S**)

Restriction d'accès

Un type de site/poste n'a accès qu'à un sous-ensemble de types de propriétés (de type d'entités ou de type d'associations)

Un type de site/poste n'a accès qu'à un sous-ensemble d'occurrences (de type d'entités ou de type d'associations)

Un type de site/poste n'est autorisé à effectuer que certains types d'accès (sur un type d'entités, un type d'associations ou un type de propriétés)

Système d'autorisation d'utilisation des données

Représente la vue sur les données de chaque type de site et de chaque type de poste

Vue de type d'entités ou de type d'associations

Représentation de l'ensemble des données du type d'entités ou du type d'associations qu'un type de site ou qu'un type de poste peut manipuler, suivant un type d'accès

Il se restreint à un sous-ensemble de types de propriétés ou d'occurrences

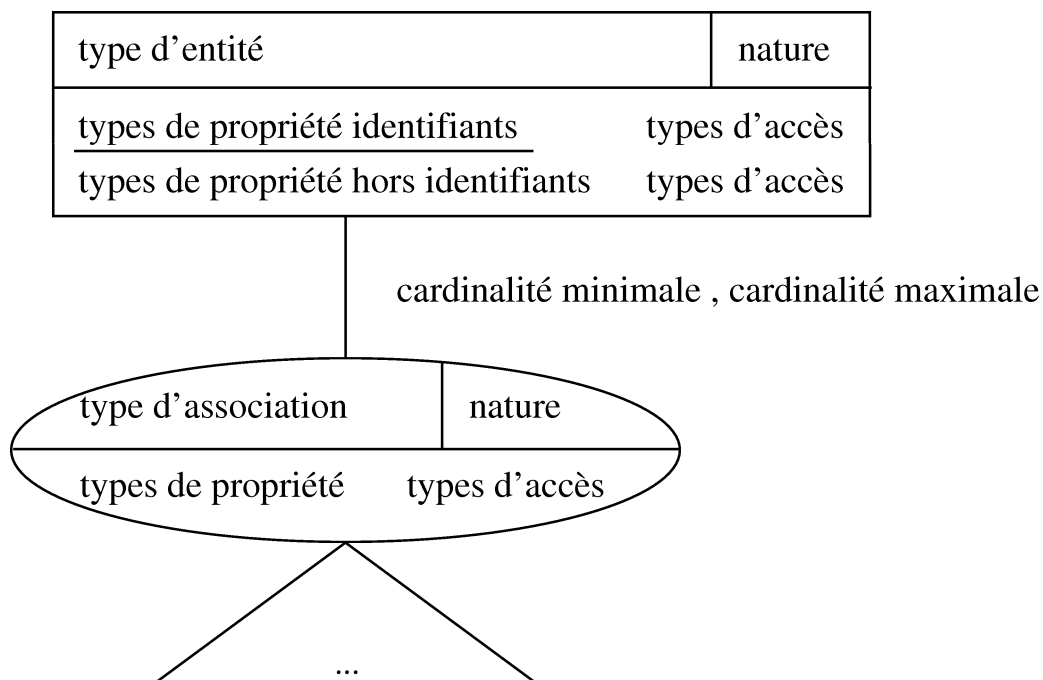
Groupe de données

Partition homogène de la visibilité d'un type de site ou d'un type de poste sur un ensemble de données

Données privées (mises à jour par le type de site et inaccessibles aux autres types de sites), protégées (mises à jour par le type de site et consultables par d'autres types de sites), partagées (mises à jour par plusieurs types de sites et consultables par plusieurs types de sites), consultables (pour le type de site)

Les données privées d'un type de site peuvent être privées ou protégées ou partagées pour ses types d'acteurs, les données protégées d'un type de site peuvent être protégées ou partagées pour ses types d'acteurs, et les données partagées d'un type de site sont partagées pour ses types d'acteurs

Représentation graphique du schéma organisationnel des données (global)



Représentation tabulaire d'une vue de type d'entités ou de type d'associations par un type de site ou un type de poste

Vue du type de site/poste :				
type d'entités / type d'associations	Vues			types d'accès
	nom de la vue	prédicat de sélection	type de propriétés	

Modèles voisins

- Modèle entité-association
- Modèle organisationnel des données
- Modèle relationnel
- Modèle navigationnel
- Modèle objet

Objectif

Représentation des données à mémoriser sur des supports informatiques permanents

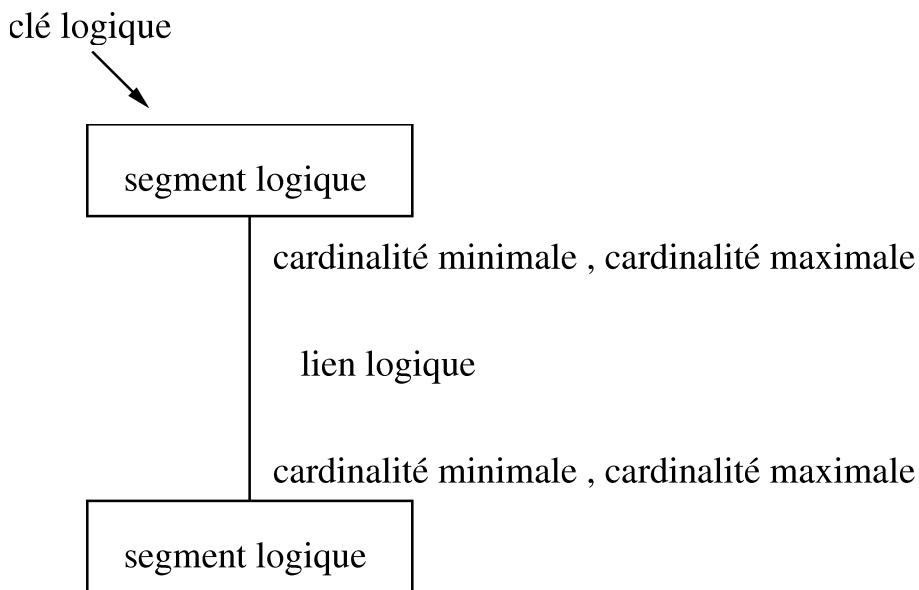
Remarque

Modèle logique des données de la méthode MERISE/2

Formalismes

- Formalisme générique dans un contexte hétérogène
- Formalisme spécifique au SGBD cible (modèle relationnel pour un SGBD relationnel, etc.)

Représentation graphique (formalisme générique) - Définitions



Segment logique

Ensemble de données élémentaires

Lien logique

Permet de retrouver les segments logiques qu'il relie

Data-item

Plus petit élément logique d'information

Data-type

Caractéristiques d'un data-item (format, nature, etc.)

Structure de données

Structure complexe de data-items

Clé logique

Data-item ou ensemble de data-items permettant de retrouver une occurrence de segment logique

Modèle logique des données de la méthode MERISE/2 précisant l'implantation logique des données permanentes sur chacune des machines logiques types du système

Formalisme générique (du MLD global), puis dans ceux spécifiques aux SGBD cibles

Composé de plusieurs MLD locaux (c'est à dire MLD propres à chacune des machines logiques types)

Définitions

Machine logique type

Ensemble de ressources matérielles et logicielles non séparables permettant d'effectuer des traitements et de stocker des données persistantes

Segment logique réparti

Segment logique (du MLD global) utilisé par des machines logiques types et stocké en tant que segment de référence ou cliché

Segment logique de type référence : les mises à jour ne se font que sur la machine logique type de référence le gérant

Segment logique de type cliché : les mises à jour se font soit périodiquement sur les machines logiques types qui l'utilisent (sous contrôle de la machine logique type le gérant), soit à la demande des machines logiques types

Segment logique de type dossier : segment logique de type référence sur l'une des machines logiques types et de type cliché sur d'autres, à un instant donné

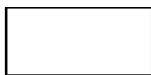
Lien logique réparti

Lien logique reliant deux segments logiques répartis

Représentation graphique (formalisme générique)

Segment logique stocké sur la machine logique type considérée (segment logique réel)

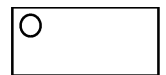
Segment
logique
de référence
sur la machine logique considérée



Segment logique
cliché rafraîchi
périodiquement
sur la machine logique considérée
depuis la machine logique de référence



Segment
logique
dossier
sur plusieurs machines logiques



Segment logique utilisé mais non stocké sur la machine logique type considérée (segment logique virtuel)

Segment logique
utilisé
mais non stocké
sur la machine logique considérée,
disponible en mise à jour
sur la machine logique de référence



Segment logique
utilisé
mais non stocké
sur la machine logique considérée,
disponible en consultation seule
depuis la machine logique de référence



Lien logique

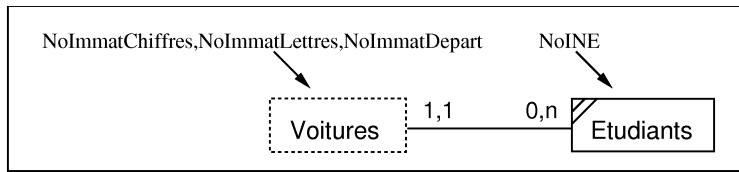
Lien logique reliant
un segment réel à un segment réel ou virtuel



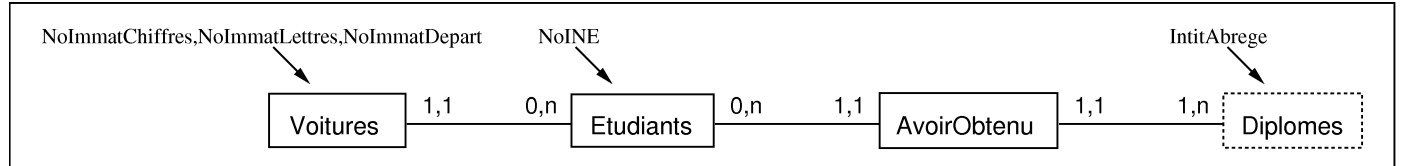
Lien logique
reliant deux segments virtuels



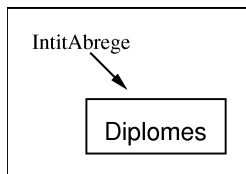
Machine logique type « Étudiants »



Machine logique type « Secrétariat »



Machine logique type « Rectorat »



Le segment logique **Diplomes** est tenu à jour (référence) par la machine logique type « Rectorat », lu par la machine logique type « Secrétariat » à chaque utilisation, ignoré de la machine logique type « Étudiants »

Le segment logique **Etudiants** est rafraîchi périodiquement par la machine logique type « Secrétariat » sur la machine logique type « Étudiants », ignoré de la machine logique type « Rectorat »

Autre dénomination

Modèle réseau

Modèles voisins

Modèle entité-association

Modèle organisationnel des données

Modèle logique des données

Modèle relationnel

Modèle objet

Origines

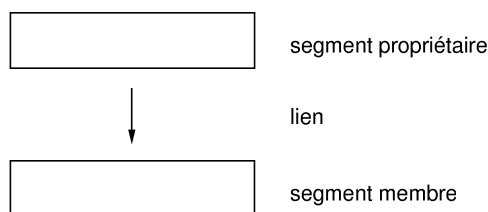
BACHMAN

CODASYL DBTG (*conference on data system languages database task group*)

Remarque

Un des modèles logiques des données de MERISE

Représentation graphique - Définitions



Segment ou article (*record*)

Collection nommée de plusieurs propriétés

Lien (set)

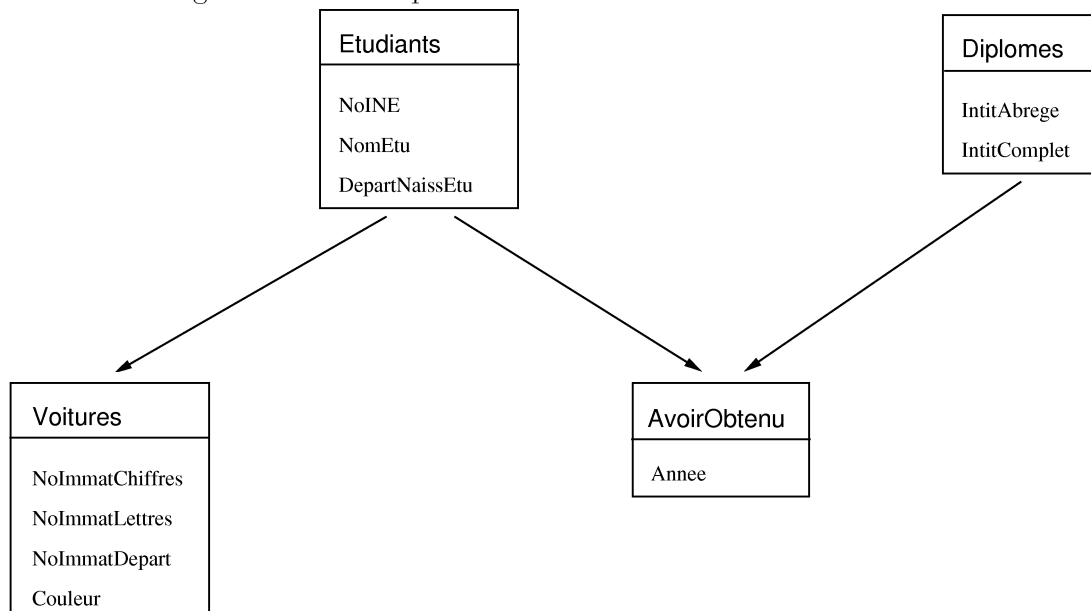
Relie un segment propriétaire (*owner*) et un segment membre (*member*)

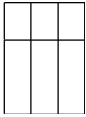
Signification : une occurrence du segment propriétaire peut « posséder » plusieurs occurrences du segment membre

Remarque : tout segment peut être propriétaire de plusieurs segments et réciproquement tout segment peut être membre de plusieurs segments

Exemple

La gestion de l'enregistrement des diplômes de tous les étudiants





Modèles voisins

Modèle entité-association

Modèle organisationnel des données

Modèle logique des données

Modèle navigationnel

Modèle objet

Origine

1970 : E. F. CODD

Remarques

Un des modèles logiques des données de MERISE

Modèle très formalisé (mathématique)

Fait suite à deux courants

Modèle hiérarchique

Modèle navigationnel (ou réseau)

Une application moyenne compte entre 30 et 50 relations

Ensemble : collection d'éléments (sans répétition ni ordre)

Exemple : les chiffres

Définition en extension

Exemple : $Chiffres = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Définition en compréhension

Exemple : $Chiffres = \{x \in \mathbb{N} : x < 10\}$

Produit cartésien de deux ensembles

Exemple :

Soit $A = \{a, b\}$

Soit $B = \{1, 2, 3\}$

Alors, $A \times B = \{(a, 1), (a, 2), (a, 3), (b, 1), (b, 2), (b, 3)\}$

Partie d'un ensemble

Exemple :

Soit $A = \{a, b\}$

Soit $B = \{1, 2, 3\}$

Alors, une des parties peut être $\mathcal{P}(A \times B) = \{(a, 2), (b, 2), (b, 3)\}$

Union de deux ensembles

Exemple :

Soit $A = \{a, b\}$

Soit $B = \{b, c\}$

Alors, $A \cup B = \{a, b, c\}$

Intersection de deux ensembles

Exemple :

Soit $A = \{a, b\}$

Soit $B = \{b, c\}$

Alors, $A \cap B = \{b\}$

Différence de deux ensembles

Exemple :

Soit $A = \{a, b\}$

Soit $B = \{b, c\}$

Alors, $A \setminus B = \{a\}$

Domaine D : ensemble de valeurs

Exemples : chaînes de caractères de longueur 25, entiers

Attribut A (ou colonne) : nom de constituant (ou variable) prenant ses valeurs dans un domaine

Exemples : département de naissance d'un étudiant, N°INE d'un étudiant, couleur d'une voiture

Relation R (ou table) (définition en extension)

Partie (ou sous-ensemble) du produit cartésien $D_1 \times D_2 \times \dots \times D_n$ sur les attributs $A_1, A_2, \dots, A_n$ dont chaque élément $(d_1, d_2, \dots, d_n)$ est un **tuple** (ou n -uplet) avec $d_i \in D_i$ le domaine de l'attribut A_i pour tout $1 \leq i \leq n$

R				avec $\{d_i, d'_i, \dots, d''_i\} \subseteq D_i$ pour tout $1 \leq i \leq n$
A_1	A_2	$\dots$	A_n	
d_1	d_2	$\dots$	d_n	
d'_1	d'_2	$\dots$	d'_n	
$\vdots$	$\vdots$	$\ddots$	$\vdots$	
d''_1	d''_2	$\dots$	d''_n	

Chaque cellule (intersection d'une ligne et d'une colonne) admet une et une seule valeur

Degré d'une relation : nombre d'attributs (ou de colonnes)

Cardinal d'une relation : nombre de tuples (ou de lignes)

Exemple :	Etudiants		
	N°INE	NomEtu	DepartNaissEtu
	5	DURAND	33
	2	LEROI	40
	4	MARTIN	47
	7	LEROI	33
	3	DUPOND	17

Schéma de Relation $R(A_1, A_2, \dots, A_n)$ (définition en compréhension)

Liste des attributs d'une relation

Exemple : Etudiants (N°INE , NomEtu , DepartNaissEtu)

Remarque : la définition d'une relation (en terme ensembliste) implique que tous les tuples sont différents deux à deux; cependant, les SGBDR ne vérifient pas cette propriété (doublons autorisés) et cela conduit à choisir systématiquement une clé (parmi les clés candidates) pour chaque relation

Clé (primaire ou principale)

Attribut ou groupe d'attributs minimal qui identifie de façon unique chaque tuple d'une relation

Exemples : numéro INE de la relation Etudiants, numéro d'immatriculation complet (chiffres, lettres et département) de la relation Voitures

Contre-exemple : nom des étudiants (homonymies)

Clé étrangère

Attribut(s) d'une relation qui correspond à la clé (primaire) d'une autre (ou de la même) relation

Exemple : le N°INE de la relation Voitures (indiquant quel étudiant possède cette voiture)

Remarque : la clé étrangère ne fait pas partie de la clé primaire de la même relation

Schéma [de la base de données] relationnel[le]

Collection de schémas de relations et de contraintes d'intégrité, qui représentent l'univers réel du système d'information

::= conditions à vérifier par toutes les relations : dépendance entre attributs, prédicat sur les valeurs d'un ou plusieurs attributs des tuples d'une relation, ...

Remarque : on distingue des contraintes statiques/dynamiques, sur les tuples/attributs/relation/inter-relation

Contrainte de clé (primaire) : une relation possède une clé primaire

Exemple : l'attribut N°INE pour la relation Etudiants

Contrainte de clé étrangère : l(es) attribut(s) d'une relation ont la même sémantique que la clé primaire correspondante d'une autre (ou de la même) relation

Exemple : le N°INE des relations Voitures et Etudiants

Contrainte d'intégrité référentielle : l'ensemble des valeurs d'une clé étrangère est inclu dans ou égal à l'ensemble des valeurs de la clé primaire correspondante

Exemple : $\text{valeurs}(\text{Voitures.N}^\circ\text{INE}) = \{5, 4\} \subseteq \text{valeurs}(\text{Etudiants.N}^\circ\text{INE}) = \{5, 2, 4, 7, 3\}$

Contrainte existentielle : les valeurs d'un attribut sont définies pour tous les tuples de la relation

Exemple : valeur d'indétermination (NULL) interdite pour le nom de l'étudiant

Contrainte de domaine : précision sur le domaine des valeurs d'un attribut (liste ou intervalle de valeurs, formats, ...)

Exemples : la couleur d'une voiture doit être jaune ou orange ou rouge, un département (de naissance d'un étudiant ou d'immatriculation de voiture) doit être compris entre 1 et 97

Contrainte de valeur par défaut : valeur à affecter à un attribut lors de l'insertion d'un tuple (si l'instruction correspondante n'affecte pas de valeur!)

Exemples : année courante pour l'année d'obtention d'un diplôme, 33 pour le département

Contrainte sur un tuple : prédicat (ou formule) mettant en jeu plusieurs attributs

Exemple : le prix de vente d'un produit doit être supérieur à son prix d'achat

Contrainte de dépendance fonctionnelle : vérification des dépendances fonctionnelles entre groupes d'attributs

Contrainte de normalité : vérification du respect d'une forme normale

Contrainte de cardinalité : borne pour le nombre de tuples d'une relation

Exemple : un étudiant ne peut pas avoir obtenu plus de cinq diplômes

Contrainte de type pré-post condition : condition vérifiée par la BD lors de son évolution, sous forme d'une condition explicite avant, pendant ou après une opération de mise à jour

Exemples : ne supprimer un diplôme (les intitulés) que si aucun étudiant ne l'a obtenu, n'insérer une voiture que si l'étudiant qui la possède est déjà enregistré

Règle active : action(s) à exécuter suite à un changement d'état de la BD

Exemple : dès qu'un étudiant est supprimé, supprimer les voitures qu'il possède et les diplômes qu'il a obtenus

...et toute autre contrainte spécifique

Dépendances fonctionnelles

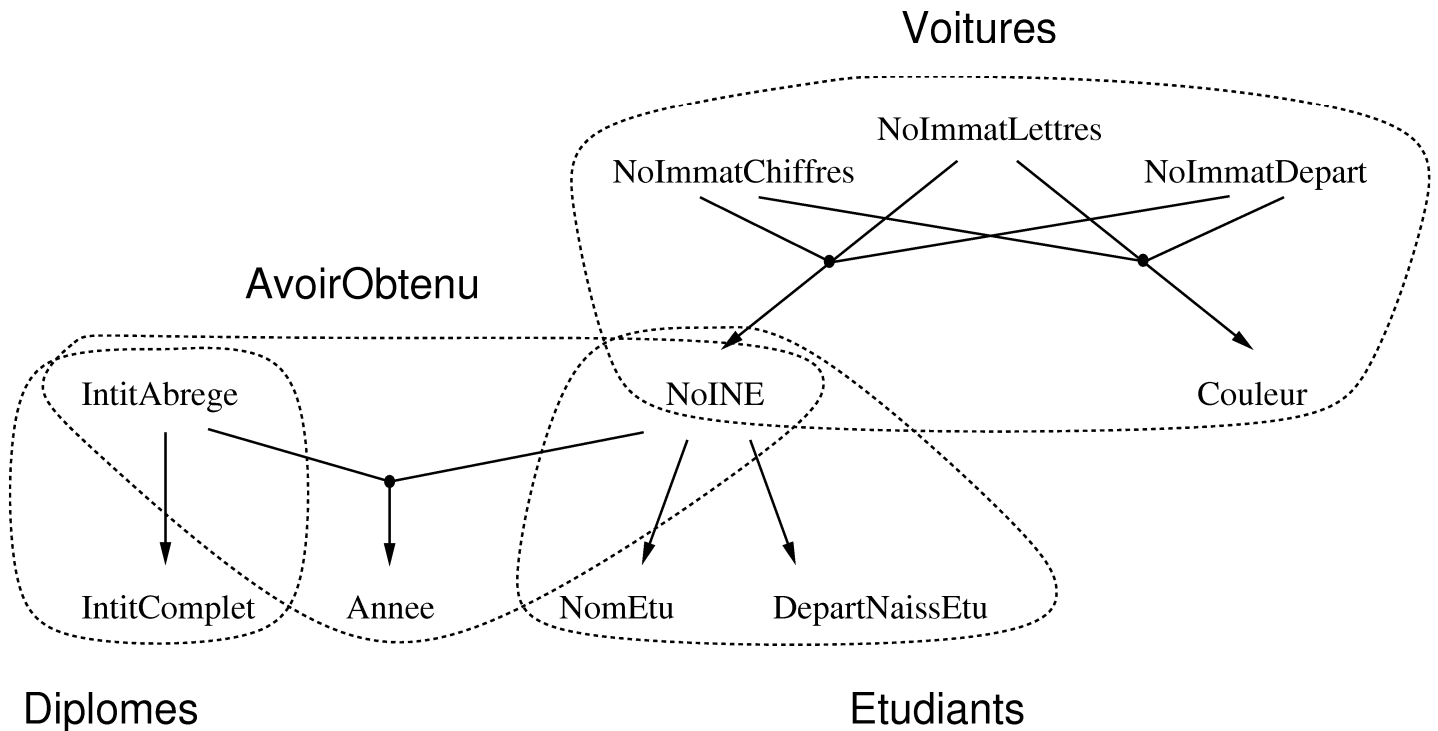
$\text{IntitAbrege} \longrightarrow \text{IntitCompleet}$

$\text{N}^\circ\text{INE} \longrightarrow \text{NomEtu}, \text{DepartNaissEtu}$

$\text{N}^\circ\text{INE}, \text{IntitAbrege} \longrightarrow \text{Annee}$

$\text{N}^\circ\text{ImmatChiffres}, \text{N}^\circ\text{ImmatLettres}, \text{N}^\circ\text{ImmatDepart} \longrightarrow \text{Couleur}, \text{N}^\circ\text{INE}$

Hyper-graphe des dépendances fonctionnelles



Schémas des relations

Etudiants (N°INE , NomEtu , DepartNaissEtu)

Voitures ($\text{N}^\circ\text{ImmatChiffres}$, $\text{N}^\circ\text{ImmatLettres}$, $\text{N}^\circ\text{ImmatDepart}$, Couleur , N°INE)

Diplomes (IntitAbrege , IntitCompleet)

AvoirObtenu (N°INE , IntitAbrege , Annee)

Contraintes d'intégrité

Contraintes de clé (primaire)

N°INE est la clé de la relation **Etudiants**

($\text{N}^\circ\text{ImmatChiffres}$, $\text{N}^\circ\text{ImmatLettres}$, $\text{N}^\circ\text{ImmatDepart}$) est la clé de la relation **Voitures**

IntitAbrege est la clé de la relation **Diplomes**

(N°INE , IntitAbrege) est la clé de la relation **AvoirObtenu**

Contraintes d'intégrité référentielle

$\text{valeurs}(\text{Voitures.N}^\circ\text{INE}) \subseteq \text{valeurs}(\text{Etudiants.N}^\circ\text{INE})$

$\text{valeurs}(\text{AvoirObtenu.N}^\circ\text{INE}) \subseteq \text{valeurs}(\text{Etudiants.N}^\circ\text{INE})$

$\text{valeurs}(\text{AvoirObtenu.IntitAbrege}) \subseteq \text{valeurs}(\text{Diplomes.IntitAbrege})$

EXEMPLE « JOUET » (SUITE)

Etudiants

N°INE	NomEtu	DepartNaissEtu
5	DURAND	33
2	LEROI	40
4	MARTIN	47
7	LEROI	33
3	DUPOND	17

Voitures

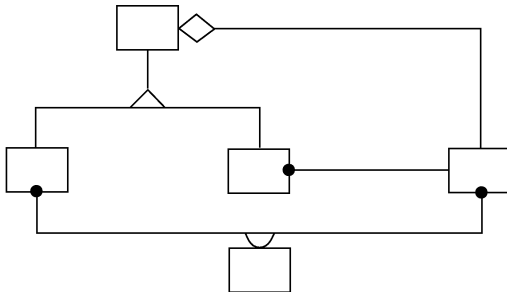
N°ImmatChiffres	N°ImmatLettres	N°ImmatDepart	Couleur	N°INE
3333	BX	33	rouge	5
4747	LA	47	rouge	4
4040	NT	40	jaune	5

Diplomes

IntitAbrege	IntitComple
DUT	Diplôme Universitaire de Technologie
BAC	Baccalauréat
MIAGe	Maîtrise des Méthodes Informatiques Appliquées à la Gestion des Entreprises
DEUG	Diplôme d'Études Universitaires Générales

AvoirObtenu

N°INE	IntitAbrege	Annee
4	DEUG	1980
2	DEUG	1982
5	DUT	1983
4	MIAGe	1982
3	DUT	1983
2	BAC	1980
4	BAC	1977
3	MIAGe	1985
5	BAC	1981
3	BAC	1981



Modèles voisins

Modèle entité-association

Modèle organisationnel des données

Modèle logique des données

Modèle relationnel

Modèle navigationnel

Objectif

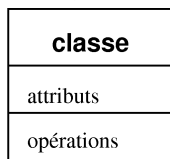
Représentation de la structure statique du système

Remarque

Modèle des données de la méthode OMT

Classe

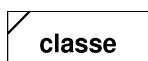
Groupe d'objets qui ont une même sémantique et des caractéristiques communes (attributs, opérations, relations avec d'autres objets)



Concrète (ou instanciable) ou abstraite (sans instance directe mais instanciable pour au moins l'une de ses sous-classes descendantes)

Classe dérivée

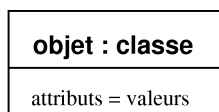
Classe redondante obtenue en composant d'autres classes et associations



Objet

Instance d'une classe

Entité qui a un sens propre dans le système d'information



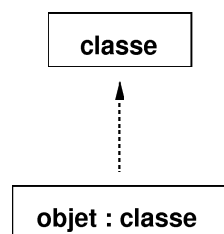
Contrainte sur les objets

Contrainte d'intégrité portant sur les objets d'une classe

Syntaxe : { ◁ contrainte ▷ }

Instanciation

Représentation de la relation existant entre une classe et ses objets



Attribut

Propriété (ou caractéristique) d'une classe

Structurée ou non

Identifiante ou non

Syntaxe : $\triangleleft \text{attribut} \triangleright [: \triangleleft \text{type de données} \triangleright [= \triangleleft \text{valeur initiale} \triangleright]]$

Remarque : on ne donne pas d'attribut identifiant aux classes

Attribut de classe

Attribut d'une classe dont la valeur est la même pour tous les objets de la classe

Syntaxe : $\$ \triangleleft \text{attribut} \triangleright [: \triangleleft \text{type de données} \triangleright [= \triangleleft \text{valeur initiale} \triangleright]]$

Attribut dérivé

Attribut dont la valeur est calculable à partir des valeurs d'autres attributs

Syntaxe : $/ \triangleleft \text{attribut} \triangleright [: \triangleleft \text{type de données} \triangleright]$

Opération

Fonction applicable aux objets de la classe

Syntaxe : $\triangleleft \text{opération} \triangleright [(\triangleleft \text{arguments} \triangleright)] [: \triangleleft \text{type de données en retour} \triangleright]$

Opération de classe

Fonction applicable à la classe

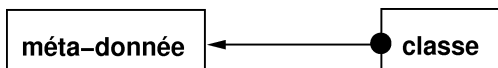
Syntaxe : $\$ \triangleleft \text{opération} \triangleright [(\triangleleft \text{arguments} \triangleright)] [: \triangleleft \text{type de données en retour} \triangleright]$

Méthode

Mise en œuvre d'une opération

Méta-donnée

Une instance de la classe possède non seulement tous les attributs de la classe mais également ceux de la classe méta-donnée



Module

Regroupement logique de classes et d'associations (et de généralisations)

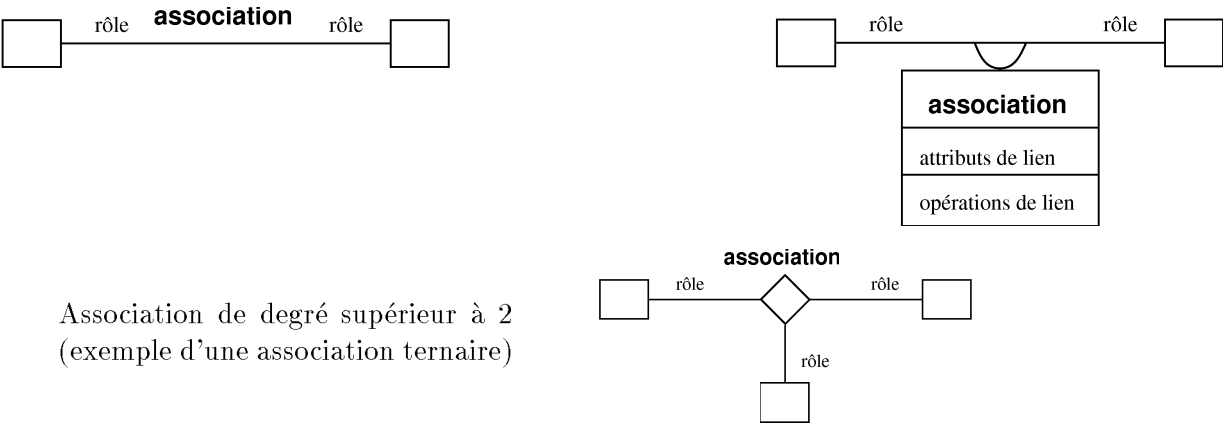
Association

Relation sémantique entre classes représentant ainsi l'ensemble des liens entre les objets des classes participant à l'association

Réflexive ou non

Mise en œuvre soit par un attribut de classe (dont la valeur pointe sur un objet), soit par une classe

Degré : nombre de classes participant à l'association



Association de degré supérieur à 2
(exemple d'une association ternaire)

Lien

Instance d'une association

Connexion entre objets

Cardinalité (ou multiplicité des associations)

Nombre d'objets de classe dans un lien, pour chaque classe participant à l'association

_____	un	1 et 1 seul
_____●	plusieurs	0 ou plus
_____○	facultatif	au plus 1
_____● ¹⁺	obligatoire	au moins 1
_____● ^{a-b,c}	a,...,b ou c	de a à b ou c
_____● ^{ordered}	ordonné	0 ou plus

Rôle

Précision sur la sémantique de l'association

Attribut de lien

Propriété (ou caractéristique) d'une association

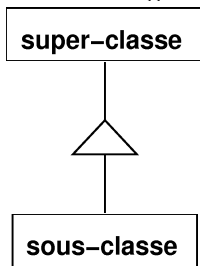
Association dérivée

Association (redondante) obtenue en composant d'autres associations



Généralisation (ou héritage)

Relation $\ll$ est un $\gg$



Création d'une hiérarchie (graphe orienté sans cycle), avec la possibilité d'avoir des héritages multiples

La super-classe possède les attributs et opérations communs à toutes ses sous-classes descendantes

En plus de ses attributs, opérations et associations propres, une sous-classe hérite de tous les attributs, opérations et associations de ses super-classes ascendantes

Surcharge d'opération : opérations mises en œuvre de façons différentes dans les classes d'une hiérarchie mais dont la spécification est la même (même nom, mêmes nombre et type d'arguments, même type du résultat)

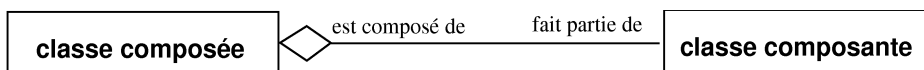
Remarque : il est possible de préciser qu'un objet puisse appartenir à (aucune, une ou) plusieurs sous-classes

Remarque : il est possible de différencier les sous-classes relativement à la valeur d'un attribut discriminant

Remarque : un attribut ou une opération commun à plusieurs super-classes ascendantes est hérité une seule fois par la sous-classe

Assemblage (ou agrégation)

Relation $\ll$ est composé de / fait partie de $\gg$



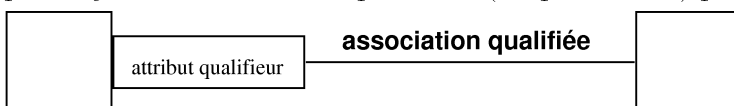
Légende : le composé (i.e. le tout : ensemble, assemblage, agrégat) est constitué du composant (i.e. la partie : l'élément)

Remarque : il existe une forme alternative

Remarque : propagation des opérations du composé vers le composant, implicitement ou explicitement

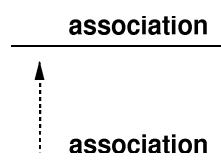
Association qualifiée

Réduction de la multiplicité d'une association sur une cardinalité un-plusieurs (ou plusieurs-plusieurs) par l'ajout d'un attribut qualifieur (le qualificatif) pour distinguer les objets

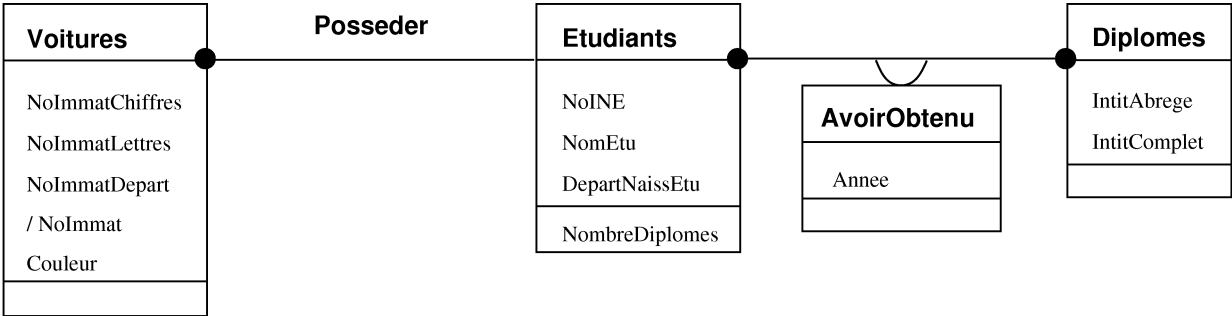


Contrainte entre associations

Contrainte d'intégrité portant sur les associations



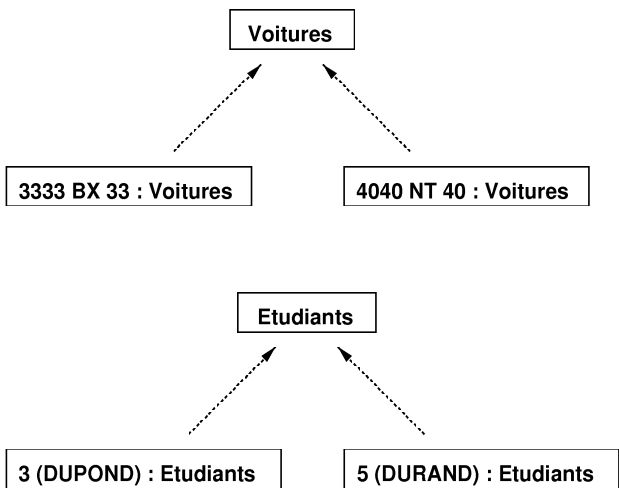
La gestion de l'enregistrement des diplômes de tous les étudiants



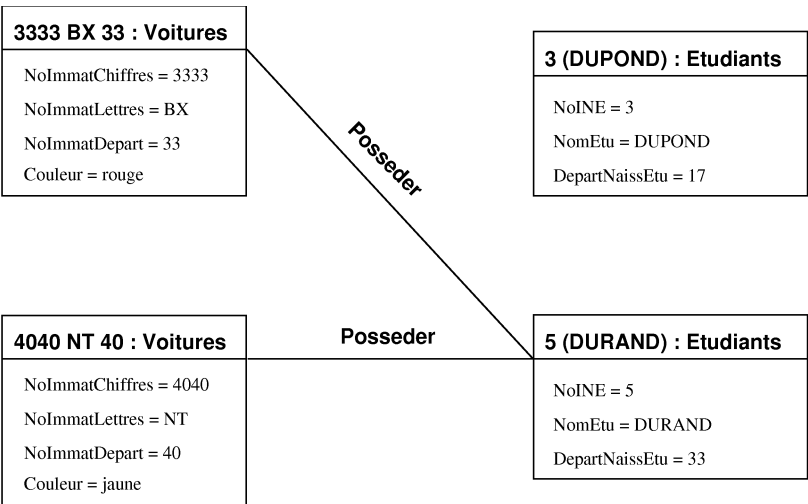
Contraintes

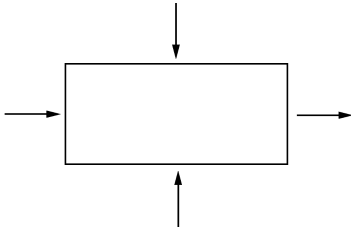
- { Unicité de (NoImmatChiffres , NoImmatLettres , NoImmatDepart) }
- { Unicité de NoINE }
- { Unicité de IntitAbrege }
- { Unicité de IntitComplet }
- { NoImmatDepart prend ses valeurs dans 1-19,2A,2B,21-95,97 }
- { DepartNaissEtu prend ses valeurs dans 1-19,2A,2B,21-95,97 }
- { / NoImmat est la concaténation des attributs NoImmatChiffres , NoImmatLettres et NoImmatDepart }

Graphe (partiel) d'instanciation



Graphe (partiel) des objets





Autre dénomination

Diagramme des données

Objectif

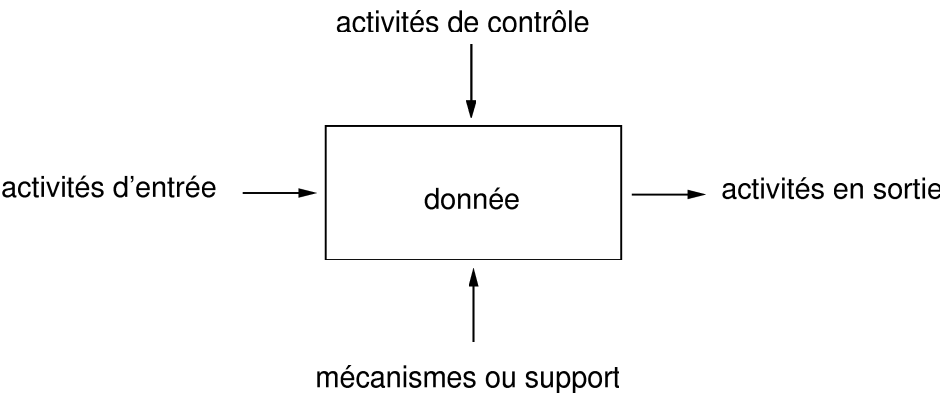
Représentation de l'ensemble des données du système

Remarque

Modèle des données de la méthode SADT

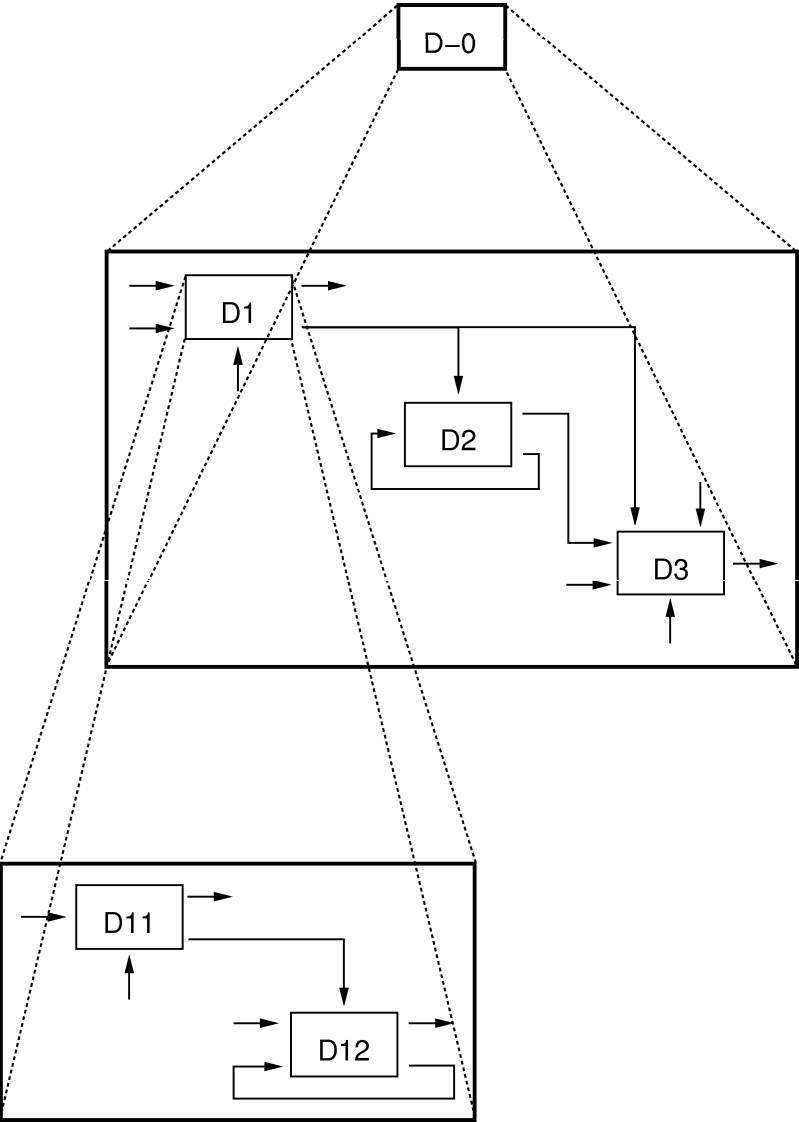
Une donnée :

manipulée, observée, consommée par des activités de sortie (activités utilisatrices)
créée à partir d'activités d'entrée (activités génératrices)
sous le contrôle d'activités de création et de manipulation, observation, utilisation
en s'appuyant sur les potentialités des mécanismes ou supports de la donnée



Arbre hiérarchique des datagrammes

- $D - 0$: système global
- D_i : $i^{ème}$ fils de $D - 0$
- $D\omega_j$: $j^{ème}$ fils de $D\omega$

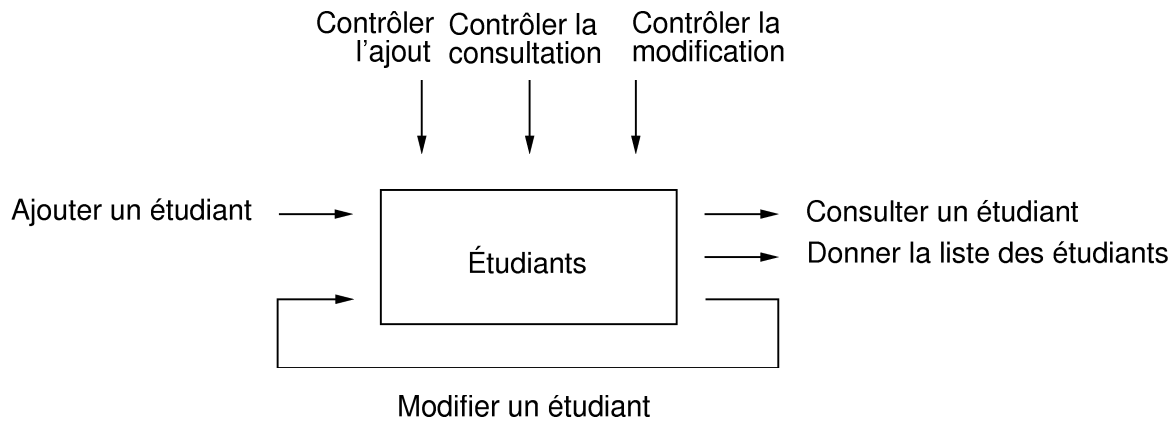


Remarques

La donnée est identifiée par un nom tandis que les activités d'entrée, de contrôle et de sortie sont désignées par un verbe

Les mécanismes expriment le dispositif de mémorisation

Exemple



Autres dénominations

Grille de recueil des données
(*Document/data element grid*)

Objectif

Description de l'utilisation de chaque rubrique

Remarque

Modèle de l'étude de l'existant de la méthode d'informatisation traditionnelle

Représentation graphique

Indique si une rubrique figure sur un document/fichier (*X*)

Rubriques	Documents en entrée				Fichiers				Documents en sortie			
			...				...				...	

Un document peut être en entrée (informations saisies) ou en sortie (informations résultats), sur support papier ou écran

Un fichier, qu'il soit informatisé ou non, correspond à des données internes (informations stockées)

Objectif

Description d’une rubrique d’information

Remarque

Modèle de l’étude de l’existant de la méthode d’informatisation traditionnelle

Représentation graphique

Identification de la rubrique

- Fichier de rattachement :
- Nom :
- Nom symbolique :
- Description :
- Type :
- Longueur :
- Format :
- Contrôle :

Contrainte d’accès

- Type d’accès :
- Confidentialité :
- Responsable de la mise à jour :

Contexte d’utilisation

- Événement source de consultation :
- Événement source de mise à jour :

Objectif

Description d'un fichier (et de ses rubriques)

Remarque

Modèle de l'étude de l'existant de la méthode d'informatisation traditionnelle

Représentation graphique

Identification du fichier :

Intitulé :

Support :

Longueur :

Volumes moyen / maximum :

Règle d'évolution :

Rubriques

Nom symbol. de la rubrique	Nom de la rubrique	Clé?	Index?	Fichier lié	Commentaires

Contraintes d'intégrité :

Objectif

Description d'un document (et de ses rubriques)

Remarque

Modèle de l'étude de l'existant de la méthode d'informatisation traditionnelle

Représentation graphique

Identification du document :

Intitulé :

Support :

Volume / Périodicité :

Rubriques

Nom symbol. de la rubrique	Nom de la rubrique	Format	Commentaires

Compléter le tableau suivant, en indiquant si un concept est couvert ou non par un modèle

Concepts	Modèles	graphe acteurs flux	réseaux de Pétri	diagr. circula ^o informa ^o	modèle concept. traite ^t	modèle organisa. traite ^t	acti- gramme
flux, événement, place							
synchronisation							
traitement, opération, phase, transition							
action, tâche							
règle d'émission							
acteur							
enchaînement							
périodicité							
durée							
type							
ressource							
responsable							

Étude de l'existant

Collecte

Représentation

Validation

Dossier de l'existant validé

Étude d'opportunité

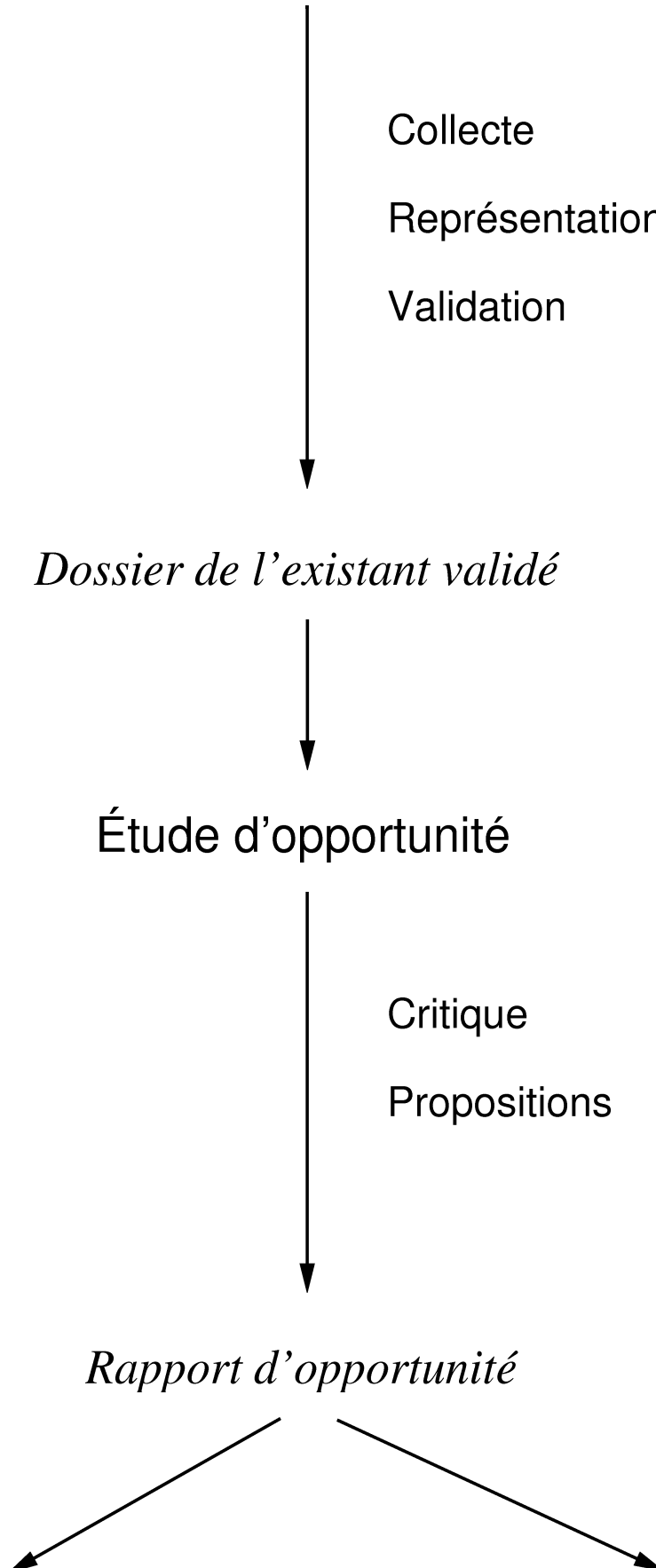
Critique

Propositions

Rapport d'opportunité

Cahier des charges

Plan directeur de réalisation



Objectif

Description de l'existant (par différentes représentations et modèles) en collectant toutes les informations utiles et nécessaires

Phases

Collecte

- Aller sur le terrain
- Observer
- Questionner
- Prendre des notes
- etc.

Représentation

- Rédiger
- Formaliser les renseignements collectés
- Modéliser
- etc.

Importance

Toute l'application en dépend $\implies$ exhaustivité et exactitude

Gravité croissante d'une étude préalable se révélant incomplète ou inexacte lors de l'analyse fonctionnelle et/ou organique (peu grave), de la programmation (dommage), de l'exploitation (catastrophique)

Niveaux de précision

- Global pour l'étude d'opportunité
- Détaillé pour l'analyse fonctionnelle
- Plus approfondie encore pour l'analyse organique

Délimitation et caractérisation du domaine de l'étude de l'existant

En fonction de la taille de l'organisation

Décomposition en sous-systèmes

- Informatisation de chaque sous-système indépendant et réalisation d'interfaces permettant de transférer des informations
- Relativement aux grandes fonctions de l'organisation ou aux sous-systèmes d'information ayant la moindre dépendance
- Plusieurs niveaux : système d'information, application, projet

Objectif

Recueillir et sélectionner les informations intéressantes parmi toutes les informations vues ou entendues

Informations à recueillir

Nature, volume, fréquence, précision observée ou requise, durée de vie, ancienneté, etc.

Exemplaires vierges et renseignés

Critères

Informations sur le système actuel et futur

Informations sur le système ou du système

Ne recueillir que les informations directement utiles et non les informations générales sur l'environnement non directement liées à l'étude

Informations de type

Dynamique

Circulation des documents dans l'espace et dans le temps

De transformation

Procédure de traitement, règle de gestion, enchaînement des tâches, formule de calcul, condition de déclenchement des traitements

Statique

Données élémentaires et documents, services et postes de travail

Degré de conscience ou d'expression de l'information

Collecter les informations exprimées (par écrit ou oralement)

Détecter les informations conscientes non exprimées

Deviner les informations inconscientes

Remarque : selon le cas, faire exprimer/reconnaître les informations non exprimées ou les laisser dans l'ombre

Moyens

À partir de documents (écrits et collectés)

Documents existants : d'exécution, de gestion ou à établir entièrement

Documents à compléter (questionnaire)

Entretien (ou enquête orale)

Accompagnant des documents écrits ou sans document écrit préalable

Contraignant ou peu directif

Quelques conseils : fixer un rendez-vous, préparer l'entretien, être ponctuel, préciser l'objectif, questionner, écouter, noter, demander tous les documents nécessaires, conclure, faire un compte-rendu

Observation (ou enquête visuelle)

Après un entretien par exemple

Qualitative ou quantitative

Organigramme, fiche de description de fonction

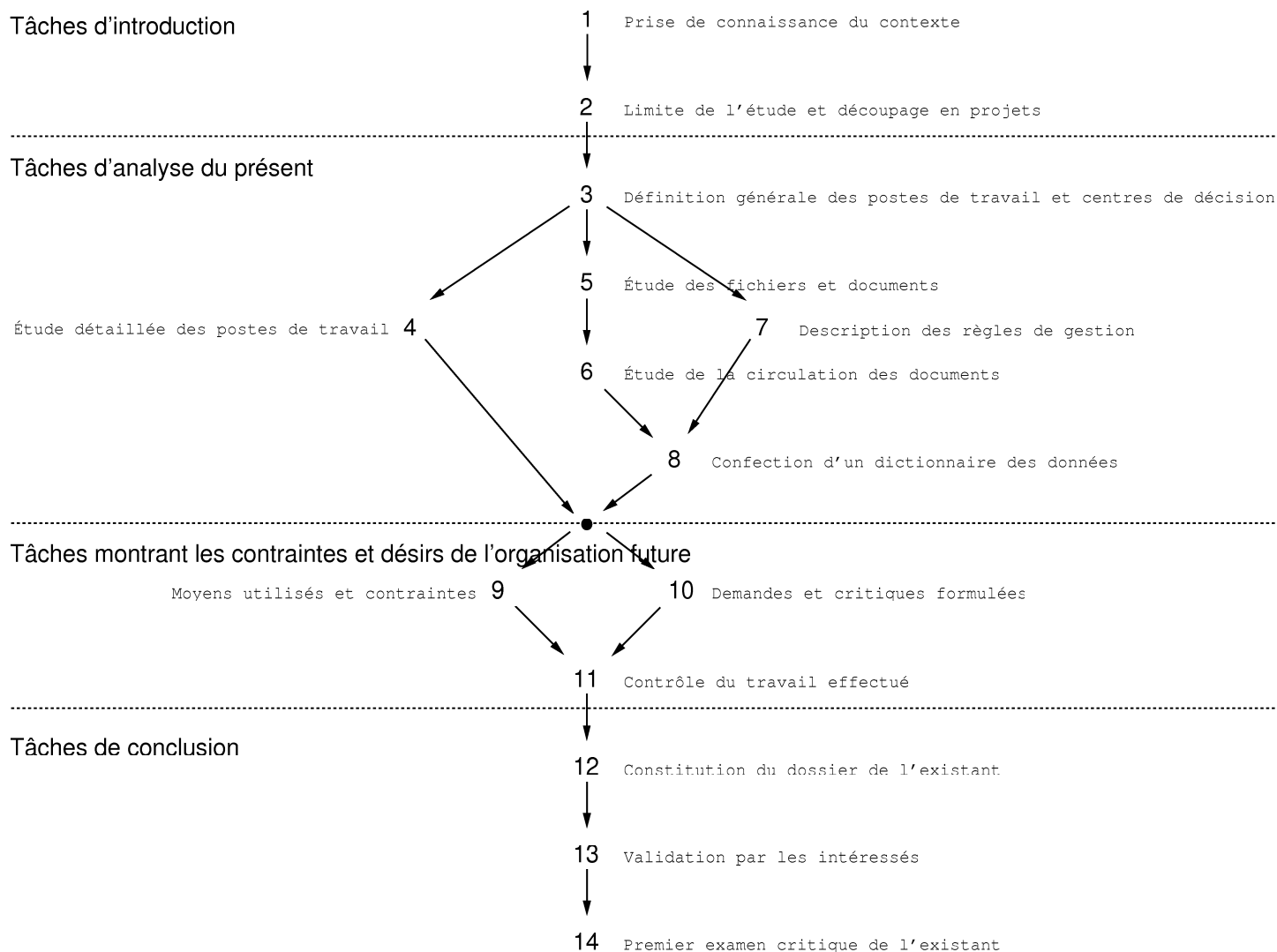
Graphe acteurs-flux ou modèle de flux ou diagramme de flots de données

Diagramme de circulation de l'information (ou des documents, dans l'espace et/ou dans le temps) ou modèle organisationnel des traitements ou modèle conceptuel des traitements ou réseaux de Pétri

Table de décision, algorithme

Dictionnaire des données ou grille d'analyse des rubriques, fiche de description de rubrique

Modèle entité-association ou modèle relationnel ou modèle navigationnel ou fiches de description de fichier/document



Remarques

Il existe des tâches séparées dont la collecte est commune, et inversement une tâche peut nécessiter des collectes séparées

Avancées ou retour en arrière possibles

Il ne s'agit que d'un ordonnancement possible

- 1** Prise de connaissance du contexte c'est à dire de la structure hiérarchique de l'organisation et de son environnement social, technique et économique

Organigramme, description littéraire

- 2** Re-formulation des limites de l'étude et du découpage en projets à partir de ce qui a été décrit ou demandé

Techniques d'expression et de re-formulation

Document écrit présentant sommairement les projets soumis et ses limites

- 3** Au niveau du projet retenu, étude de la structure hiérarchique et liste des postes de travail et des centres de décision

Techniques d'organisation d'entreprise et du travail

Diagramme de circulation (sans traitements)

- 4** Étude détaillée des postes de travail

Techniques d'organisation ou d'économie d'entreprise

Fiches de fonction, listes des tâches effectuées

- 5** Établissement d'une liste des fichiers et des documents

Tableau donnant pour chaque fichier et chaque document ses volume, activité, périodicité, etc.

- 6** Représentation de la circulation des documents mentionnant les traitements

Diagramme de circulation des documents (avec traitements et fichiers) ou réseaux de Pétri ou algèbre relationnelle

- 7** Recensement et description des règles de gestion, c'est à dire les procédures et règles de traitement

Table de décision, algorithme

- 8** Confection d'un dictionnaire des rubriques

Grille d'analyse des rubriques, fiche de description de rubrique, éventuellement schéma entité-association ou schéma relationnel ou fiche de description de fichier

- 9** Récapitulation des moyens et ressources utilisés et des contraintes (durée, délai, fréquence, volume, coût, réglementation, ergonomie)

Standard de productivité en matière d'organisation d'entreprise et du travail

- 10** Récapitulation des demandes d'information et des critiques formulées par le personnel consulté

Entretiens

- 11** Contrôle du travail effectué c'est à dire des éléments du système d'information existant répertoriés au cours de l'analyse

- 12** Constitution du dossier de l'existant

Première version du cahier des charges détaillé

- 13** Validation de l'étude auprès des personnes compétentes et concernées par l'étude

- 14** Premier examen critique des personnes ayant réalisé cette analyse mentionnant leurs avis sur l'existant

Objectif

Faciliter la prise de décision par la direction générale en commission informatique sur la suite à donner à l'étude (par un rapport synthétique présentant les principales critiques formulées et les diverses solutions envisageables) c'est à dire la mise en œuvre d'un certain nombre de projets d'automatisation parmi ceux proposés

Critique du système d'information existant

Niveaux : général, des domaines d'étude, des services et postes de travail

Causes possibles de dysfonctionnement

- Insuffisance des moyens de traitement de l'information

- Méthodes de traitements mal formalisées ou archaïques

- Mauvaise organisation

- Circuits informationnels mal étudiés

- Documents inexistants ou inutiles ou incomplets

- Fichiers inexistants, mal structurés, incomplets, redondants, etc.

Exposé des besoins nouveaux exprimés par les utilisateurs

Propositions de solutions

Pallier les dysfonctionnements et améliorer le système

Solutions non informatisées

- Personnel

- Matériels

- Documents

- Méthodes

- Fichiers

Solutions informatisées

- Définition des tâches devant être automatisées

- Découpage en projets d'automatisation homogènes et relativement indépendants, en faisant apparaître les priorités de réalisation

Évaluation financière (coût estimé et gain escompté) de chaque proposition, informatisée ou non

Présentation de l'ordre des priorités entre les différentes solutions, informatisées ou non

Mesure de la faisabilité et établissement de la mise en œuvre de chaque proposition, informatisée ou non

Destinataires

Service informatique, constructeur ou société de services en informatique

Objectif

Définir les besoins en matériel et en logiciel du futur système informatique (pour permettre de choisir l'une des solutions) afin d'établir un contrat entre utilisateurs et informaticiens

Renseignements informatiques

Description détaillée des fonctionnalités attendues

Évaluation chiffrée des volumes à mettre en œuvre

Données à stocker, sauvegarder, saisir, imprimer

Modes de travaux envisagés : immédiat ou en temps différé, unitaire ou par lot

Nombre maximum d'utilisateurs connectés simultanément

Définition des besoins en matériel

Types de postes de travail

Réseau de communication utilisé

Périphériques particuliers

Définition des besoins en logiciel

Progiciels systèmes

Progiciels d'application

Renseignements technico-commerciaux

Avant la livraison

Conditions financières des matériels et logiciels

Conditions d'extension de la configuration matérielle et des logiciels, en assurant portabilité et compatibilité

Conditions d'implantation des matériels et logiciels

Conditions d'essais

Conditions de livraison

Après la livraison

Conditions de maintenance

Durée d'utilisation

Formation du personnel

Aide à la mise en œuvre

Documentation

Conditions de reconversion des applications existantes

Présentation de toutes les modalités de réalisation (des programmes d'application spécifiques) :

responsabilités, personnel d'exécution, planning de réalisation de chaque projet, liaisons entre les différents projets (ou logiciels acquis), interventions d'informaticiens extérieurs à l'organisation

Cahier des charges



Conception

[Modèle de communication]

Modèle des traitements

Modèle des données

Validation



Schéma conceptuel

Objectif

Obtenir un schéma général de structuration des traitements et des données, à un niveau conceptuel (c'est à dire indépendant de tout matériel ou logiciel de base)

Quels sont les objectifs à atteindre? (*Quoi faire ?*)

Critères d'un schéma conceptuel

Communicable - Conforme - Valide (complet et cohérent) - Réalisable

Principe d'indépendance des traitements et des données

Pour cela, un logiciel doit être capable, au moment de l'exécution des programmes, de retrouver les données nécessaires aux traitements à effectuer

Indépendance logique (respectivement physique) lorsque le schéma conceptuel (respectivement logique) des données peut être modifié sans changer les programmes

Représentation de la communication au sein de l'organisation

Modèles : graphe acteurs-flux (ou modèle de flux, diagramme de flots de données)

Représentation de l'ensemble des traitements

Modélisation des traitements avec leurs conditions d'activation, leurs règles d'utilisation et de transformation, leur enchaînement, etc.

Statique : description d'un traitement

Modèles : table de décision, langage naturel ou formel, etc.

Dynamique : spécification des conditions d'exécution et d'enchaînement de traitements pour caractériser le comportement du système

Modèles : modèle conceptuel des traitements [analytique] (ou réseaux de Pétri, actigramme), [cycle de vie d'un objet]

Application découpée en chaînes fonctionnelles (ou fonctions)

Représentation de l'ensemble des données

Modélisation de toutes les informations (et de leurs structures) devant être manipulées (et donc stockées)

Modèles : modèle entité-association (ou modèle relationnel, modèle navigationnel, modèle objet, datagramme, logique des prédicats, réseaux sémantiques, etc.)

Contraintes d'intégrité : conditions à satisfaire pour les données mémorisées par le système d'information

Statique : vérifiées à tout moment

Dynamique : caractérise la validité des changements d'états du système d'information

Remarque : certaines contraintes sont déjà incluses dans les modèles

Validations formelles des traitements et des données

Complétude des traitements

L'ensemble des traitements décrits correspondent à la définition

Cohérence des traitements

Statique : pas de contradiction

Dynamique : pas d'inter-blocage, terminaison

Complétude des données

Pas d'oubli (respect de la norme décrivant le modèle)

Cohérence des données

Conformité à la norme : pas d'ambiguïté, pas de contradiction, pas de redondance, désagrégation/décomposition

Synthèse des différents schémas (de communication, des traitements, des données) garantissant la cohérence du schéma conceptuel

Toute communication s'appuie (si besoin) sur un traitement

Tous les traitements assurent les communications de l'organisation avec son environnement et en son sein

Aucun traitement ne fait référence à une donnée n'existant pas

Toutes les données sont manipulées par au moins un traitement

Confrontation avec les utilisateurs

Schéma conceptuel

Progiciel ou Développement spécifique

*Solution informatique*

Objectif

Adapter la solution fonctionnelle à un choix technique particulier

Progiciel vs développement spécifique

Achat d'un progiciel standard

Plus économique

Présent sur de nombreux segments de marché

Produit déjà testé

S'assurer de la réelle adaptation aux besoins

Complexité du paramétrage

Peut nécessiter de recourir à un spécialiste

Existe-t-il un service après-vente, qui soit de plus viable à long terme?

Développement spécifique

Solution parfaitement adaptée aux besoins

Deux approches : traditionnelle ou génie logiciel

Deux étapes

Étape logique : choix d'organisation (*Qui fera quoi ? Où ? Quand ?*)

Étape physique : choix techniques (*Comment : avec quels moyens matériels et logiciels ?*)

Représentation des traitements

Étape logique : prise en compte des contraintes des utilisateurs faisant intervenir le temps et le lieu des traitements

Prise en compte du temps (de chaque traitement) : date de début au plus tôt, date de fin au plus tard, durée, date de début effective, etc.

Prise en compte du lieu (de chaque traitement) : communication entre les acteurs, poste de travail effectuant le traitement, traitement manuel ou informatisé (interactif ou différé)

Procédures de fonctionnement en mode dégradé (données détruites, lieu ou ressource indisponible)

Fonction découpée en chaînes de traitements

Modèles : modèle organisationnel des traitements ou schéma de circulation de l'information/document

Étape physique : fait intervenir les contraintes de ressources nécessaires et utilisées et affecter les responsabilités des traitements

Contraintes de ressources : regroupement de traitements successifs, éclatement d'un traitement

Détermination des responsables

Chaîne de traitements découpée en unités de traitements

Utilisation de fichiers permanents ou temporaires

Typologie des traitements : saisie, contrôle, mise à jour (création, modification, suppression d'une occurrence), consultation, calcul, édition, sélection [multiple], projection, tri, fusion, séparation (ou partition), interclassement (ou union), mise en attente, prise d'attente, création ou suppression de rubriques

Modèles : description de documents (écrans et imprimés), pseudo-code, algorithmes généraux ou détaillés

Remarques

Il s'agit d'un fonctionnement donné d'une organisation

Non remise en cause des traitements du niveau conceptuel mais possible adaptation aux différentes contraintes

Exemple : un traitement parallèle de l'analyse fonctionnelle est sérialisé lors de l'analyse organique afin de prendre en compte des contraintes de temps, lieu, ressources, etc.

Représentation des données

Étape logique : prise en compte des besoins d'utilisation des informations

Définition des modes d'accès aux données

Modèle : modèle navigationnel (ou modèle relationnel)

Étape physique : prise en compte des contraintes physiques liées en particulier aux matériels et logiciels utilisés

Description des données par rapport à leur implantation (Système de Gestion de Fichiers ou Système de Gestion de Bases de Données) et à un matériel et un logiciel déterminés

Bases de données, tables, fichiers (séquentiel, séquentiel indexé, à accès direct), piles, listes, etc.

« Masques » des fichiers

Calculs d'activité afin de déterminer les schémas, vues, index, clusters, redondances, etc. les plus efficaces (optimisations en temps et/ou espace)

Modèles : modèle relationnel ou description des fichiers/rubriques

Structure d'accueil

Mémoire, processeur, réseau, langage, progiciel, etc.

Interface homme-machine

Ergonomie, langage de communication

Méthode de conception

Analyse descendante

Programmation

Programmation structurée, objet (encapsulation, héritage, polymorphisme, etc.)

Exemple : assembleur, langage de 3^{ième} génération, Système de Gestion de Fichiers ou extensions pour manipuler des bases de données (hiérarchique, réseau, relationnelle, etc.)

Planning

Budget de programmation

Objectif

Passer à l'ère industrielle de la production du logiciel,
en développant des méthodes et des techniques permettant de réaliser à moindre coût des logiciels performants et fiables

Concevoir (le produit)

Résultat d'une analyse ou d'une étude de marché
Fournir un ensemble de spécifications détaillées
Choisir une interface utilisateur : mode texte ou graphique

Fabriquer

Principe : décomposer en composants plus simples, mettre au point un processus d'assemblage
Pour chaque composant identifié

- Utiliser un composant standard (déjà disponible)

 - SGF, SGBD, bibliothèque mathématique, bibliothèque de classes, applets JAVA, contrôles VBX ou ActiveX, etc.

- Le fabriquer soi-même

 - Choisir des outils adaptés : langage de programmation, atelier de génie logiciel, générateur d'applications, outils de développement rapide

 - Se procurer les équipements correspondants : configuration suffisamment performante et proche de celle de la future mise en exploitation

 - Disposer de personnels formés

- En sous-traiter la fabrication

 - Lorsque les coûts sont trop importants

 - Par une entreprise spécialisée

Implémenter les traitements

- Décomposition modulaire : unités de programme de petite taille, tous les éléments concourent à la réalisation de la même tâche (forte cohésion), indépendance et autonomie (couplage faible)

Implémenter les données

- Données internes (ou transitoires) : variables de programmation, types abstraits de données

- Données externes (ou permanentes) : base de données

Méthode

- Programmation descendante : décomposition des modules en objets plus simples jusqu'à l'obtention d'opérations élémentaires (primitive du langage ou que l'on sait déjà faire)

- Programmation objet

Langage de programmation

- Choix d'un paradigme : procédural, déclaratif, fonctionnel, L4G, objet, etc.

- Identification des besoins : objets, systèmes répartis, bases de données, systèmes concurrents, etc.

- Identification des moyens : disponibilité du produit sur les plateformes cibles, personnel formé

Choix des outils

- Outils de développement rapide : pour du prototypage car les performances sont souvent insuffisantes

- Générateurs de code : description de haut niveau des traitements à réaliser, code généré en L3G

- Outils spécialisés : SGBD, gestionnaire réseau, architecture client/serveur, etc.

Tester

- Jeux d'essais : jeux de données couvrant tous les cas possibles, générateurs de tests

- Simulation du fonctionnement : injection de données

- Tests en grandeur nature (par les utilisateurs finals)

Prouver/Valider

- Méthodes mathématiques de preuve de programmes

- Preuve des spécifications formelles du logiciel

- Utilisation d'outils de validation

- Vérifier l'adéquation aux besoins

Évaluer les performances

- Calculs des complexités a priori (et s'assurer que les charges des machines suffiront)

- Tests en grandeur nature (dans l'environnement final, dans les conditions réelles d'exploitation)

Assurer la fiabilité

- Plus aucune erreur majeure

- Risque d'erreurs (mineures) résiduelles

- Révisions successives du logiciel (versions alpha, bêta, release, mises à jour mineures et majeures)

Fournir une documentation

Technique

- Durant tout le développement : communication entre sous-équipes, rédigé quotidiennement par les développeurs

- Pour la maintenance : recherche ultérieure des causes d'erreurs

Utilisateur

- Mode d'emploi (produit sur mesure) : précis, technique, sans fioritures

- Communication (produit grand public) : rédigé par des professionnels

Proposer un service après-vente

- Maintenance sur site, ligne directe, service payant

- Formation

... le tout à moindre coût

MERISE/2 : nouvelle génération de MERISE, autour de six axes directeurs

Conservation des points forts de MERISE (garantissant une compatibilité ascendante)

Amélioration des points clés

Affinement d'un système en sous-systèmes, diagramme de flots de données, modèle entité-association étendu, représentation systématique de l'action des traitements sur les données, distinction entre les niveaux organisationnel et logique, démarche par étapes, etc.

Extension vers des aspects plus techniques

Technologie actuelle : poste/station de travail intelligent/puissant/multi-tâches/graphique/multi-fenêtres, traitement coopératif, SGBD relationnel, réseaux locaux fiables et performants, réseaux distants en progrès

Technologie à venir : SGBD réparti, SGBD orienté-objet, architecture d'application à composants hétérogènes, travail en coopération de personnes géographiquement distantes

Adaptation aux standards (typologie de projets et de démarches)

Obtention d'un système modulaire, adaptable et intégrable

Adoption de modèles et techniques reconnus internationalement

Outillage (logiciels d'aide à la conception et au développement)

Historique

Fin 1989, lancement à l'initiative de la Direction Technique Recherche et Développement de SEMA Group, avec R. COLLETTI, R. LETOUCHE, G. PANET et H. TARDIEU

Premier semestre 1991 : dépôt légal

Entre 1991 et 1993 : projets pilotes

Bibliographie

G. PANET et R. LETOUCHE, MERISE/2 : modèles et techniques MERISE avancés, Les Éditions d'Organisation, 1994

L'architecture de MERISE/2

S'organise autour des activités de pilotage, de développement (cycles de vie, modèles et techniques, livrables) et de qualité

Propose une typologie des projets

- Identifier les besoins du projet

- Analyser les enjeux et les risques liés au projet

- Classifier le projet dans une typologie de projets

- Utiliser la démarche de développement correspondante (cycle de vie, modèles et techniques, livrables)

Offre une démarche

- Décrivant, suivant l'étape du cycle de vie, un niveau d'abstraction plus ou moins détaillé sur une couverture du domaine d'étude plus ou moins large

- Privilégiant le point de vue de l'utilisateur (ergonomie, convivialité, etc.) notamment par des techniques de maquettage

- Rendant l'interface utilisateur indépendante du noyau de l'application

- etc.

MERISE/2 couvre toutes les phases du cycle de développement du logiciel en plusieurs étapes

- Schéma directeur - Étude d'opportunité

- Étude préalable

- Conception d'ensemble

- Conception fonctionnelle détaillée

- Conception technique détaillée

- Réalisation

- Mise en œuvre

MERISE/2 propose quatre niveaux d'abstraction

Conceptuel (Quoi?)

- Description des données et des traitements pour prendre en compte les règles de gestion (indépendamment de l'organisation interne et des choix ou contraintes techniques)

Organisationnel (D'où? Qui? Quand?)

- Intégration des préoccupations et des choix d'organisation (pour une organisation cible) en précisant l'utilisation des données et des traitements ainsi que la localisation des acteurs et des sites

Logique (Quand? Où? Comment?)

- Conception du logiciel correspondant aux parties à automatiser du système prenant en compte de l'état de la technologie (description d'une application développée dans le contexte des interfaces graphiques et des architectures d'applications modernes)

Physique (Comment?)

- Intégration des préoccupations et des choix techniques déterminant l'implantation des données et la mise en place des traitements

MERISE/2 modélise trois axes du système d'information

- Statique (ce qu'est le système) : description des données

- Cinématique ou dynamique (comment le système se comporte) : description des traitements (et de la succession de transformation sur les données)

- Architecture (ce que fait le système) : description des activités

Interfaces	Application			
	Statique	Cinématique	Architecture	
MC	MCD	CVO , MCTA	MFC	Règles de gestion
MOT	MOD , vues	CVO , MOTA	MFO	Règles d'organisation
maquettes IHM	MLD-MLDR	MLT-MLTR	SALMI , SAL-SALR	Primitives
	MPD	MPT		

MC : modèle de contexte

MCD : modèle conceptuel des données

CVO : cycle de vie des objets

MCTA : modèle conceptuel des traitements analytique

MFC : modèle de flux conceptuel

MOT : modèle organisationnel des traitements

MOD : modèle organisationnel des données

MOTA : modèle organisationnel des traitements analytique

MFO : modèle de flux organisationnel

MLD : modèle logique des données

MLDR : modèle logique des données réparties

MLT : modèle logique des traitements

MLTR : modèle logique des traitements répartis

SALMI : schéma d'architecture logique des moyens informatiques

SAL : schéma d'architecture logique

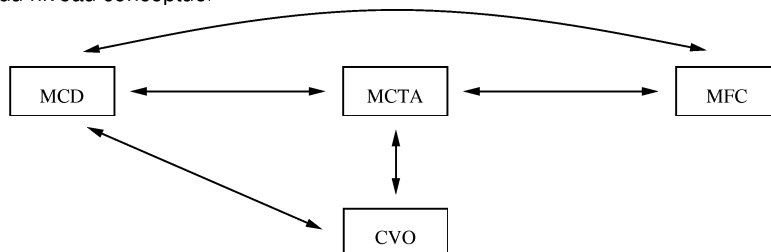
SALR : schéma d'architecture logique répartie

MPD : modèle physique des données

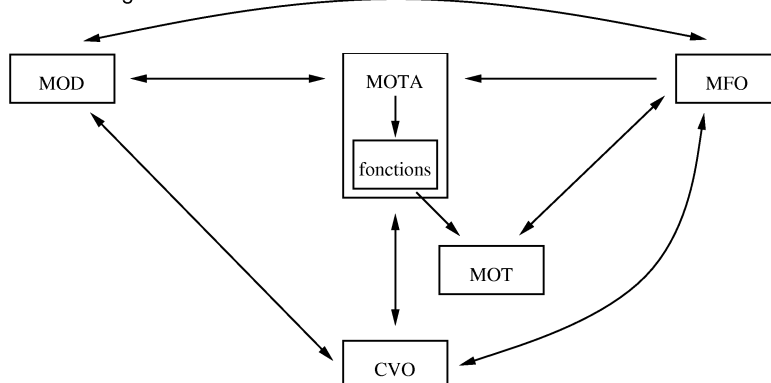
MPT : modèle physique des traitements

Enchaînements des modèles aux niveaux conceptuel et organisationnel

Modèles du niveau conceptuel



Modèles du niveau organisationnel



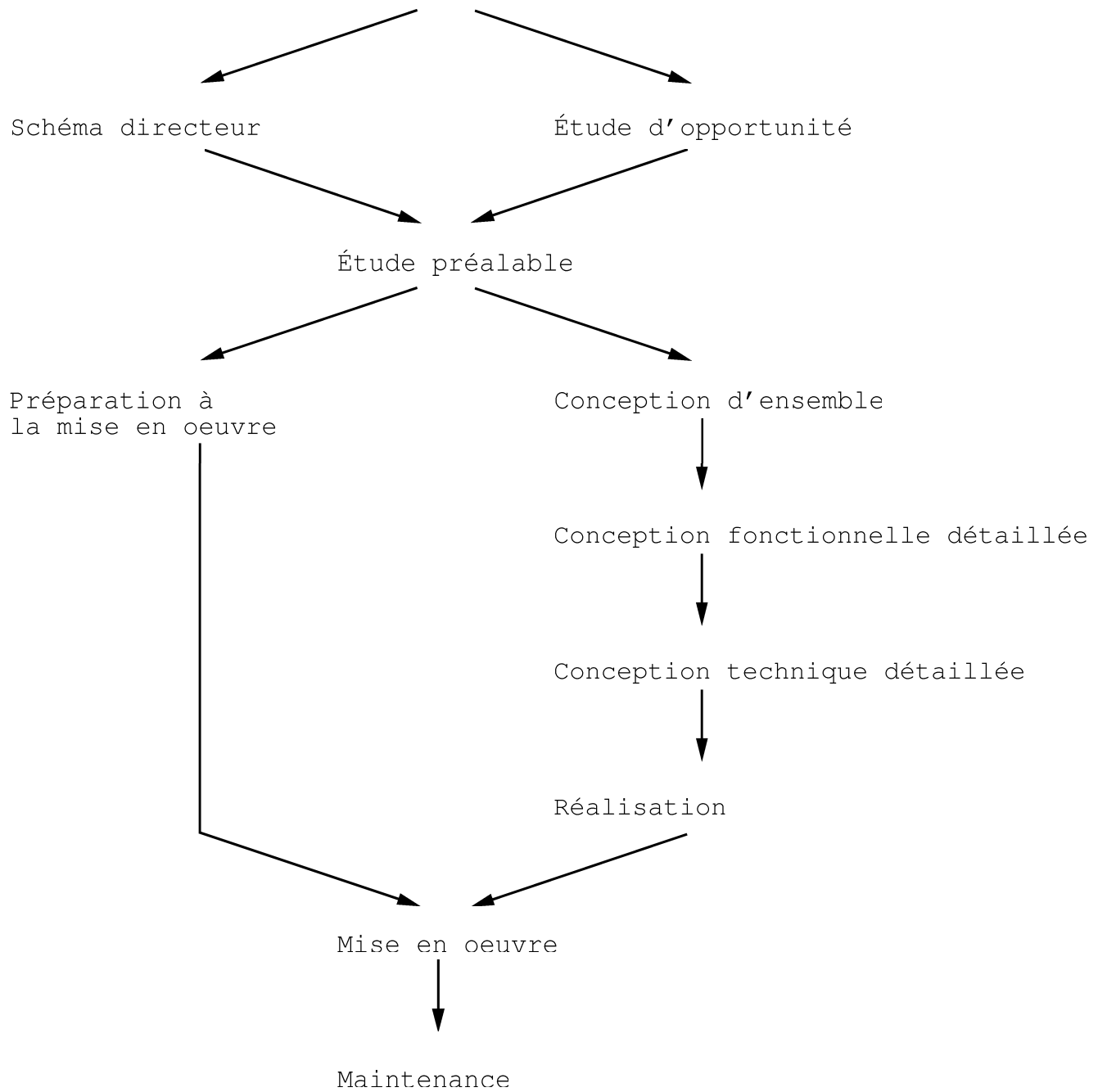
Deux exemples de choix des modèles selon la typologie des projets

Grand projet d'architecture classique *DataBase/DataCommunication* : MC, MCD, CVO, MCTA, MFC, règles de gestion ; MOD, CVO, MOTA, MFO, règles d'organisation ; MLD, MLT, primitives

Petit projet à architecture répartie : MC, MCD, CVO, règles de gestion ; MOD, CVO, MOTA, MFO, règles d'organisation ; MLD, MLDR, MLT, MLTR, primitives

CYCLE DE VIE DE RÉFÉRENCE

4



Étude d'opportunité

- Identifier le maître d'ouvrage et le maître d'œuvre
- Procéder à une première analyse
- Procéder à une première évaluation des enjeux
- Classer le projet dans une typologie de projet
- Inscrire le projet dans un plan annuel

Étude préalable

- Évaluer les enjeux (avantages stratégiques, rentabilité, contraintes réglementaires, risques)
- Vérifier sa faisabilité organisationnelle et sa faisabilité technique
- Proposer des solutions fonctionnelles et techniques (globalement et sur un sous-ensemble), et des trajectoires de migration possibles
- Estimer les coûts et délais du développement et de la mise en œuvre
- Permettre la prise de décision sur la suite à donner

Conception d'ensemble (ou conception générale)

- Obtenir une description fonctionnelle (conceptuelle et organisationnelle) et technique complète du futur système
- Arrêter les principes majeurs concernant les contraintes à respecter
- Définir les solutions de secours et dégradées éventuelles
- Définir complètement la trajectoire de migration
- Faire approuver l'ensemble par le maître d'ouvrage

Conception fonctionnelle détaillée

- Spécifier dans le détail et de façon exhaustive l'ensemble des données et traitements informatiques visibles par l'utilisateur
- Vérifier les faisabilités organisationnelle et technique
- Actualiser le dossier économique
- Faire approuver l'ensemble par le maître d'ouvrage

Conception technique détaillée

- Décrire complètement le logiciel (données et traitements) dans l'environnement technique cible
- S'assurer des performances et de la sécurité
- Vérifier ou actualiser les estimations des ressources nécessaires
- Préparer la réalisation

Réalisation

- Fournir des logiciels opérationnels

Préparation à la mise en œuvre (par le maître d'ouvrage)

- Assurer les travaux préparatoires à la mise en œuvre du système

Mise en œuvre (par le maître d'ouvrage)

- Qualifier le nouveau système
- Mettre en service le nouveau système

Étude d'opportunité

Identification des besoins

Ébauche des solutions

Synthèse

Livrable : Rapport d'étude d'opportunité

Étude préalable

Lancement

Recueil, bilan, orientations

Conception fonctionnelle et technique préalable

Appréciation (des solutions)

Décision

Livrables : Note de lancement - Plan d'assurance qualité (version 1) - Plan de développement (version 1) - Dossier de recueil, bilan, orientations - Dossier de conception fonctionnelle et technique préalable - Dossier d'appréciation - Rapport de synthèse

Remarques :

Étape considérant globalement tout le domaine concerné et dans le détail les sous-ensembles représentatifs
Maquettage sur les sous-ensembles critiques

Modèles complets (MC, MFC) ou limités aux sous-ensembles représentatifs (MCTA, MCD, CVO, règles de gestion, MFO, MOTA, MOD, règles d'organisation, SALMI, SAL, SALR, MLD, MLDR)

Conception d'ensemble (ou conception générale)

Lancement

Conception fonctionnelle d'ensemble

Conception technique d'ensemble

Actualisation du dossier d'appréciation et du plan de développement

Décision

Livrables : Note de lancement - Plan d'assurance qualité (version 2) - Plan de développement (version 2) - Dossier de spécifications fonctionnelles générales - Dossier de spécifications techniques générales - Dossier d'appréciation actualisé - Rapport de synthèse

Remarque : modèles complets (MC, MFC, MCD, MCTA, CVO, règles de gestion, MFO, MOTA, MOD, SALMI, SAL, SALR, MLD, MPT) ou limités aux sous-ensembles critiques (règles d'organisation, MLT, MLDR)

Conception fonctionnelle détaillée

Lancement

Conception fonctionnelle détaillée des postes de travail et de l'ergonomie

Conception fonctionnelle détaillée du logiciel en fonctionnement normal

Conception fonctionnelle détaillée du logiciel en fonctionnement dégradé ou de secours

Conception fonctionnelle détaillée du logiciel d'interfaces et de reprise des données

Validation technique

Actualisation du dossier d'appréciation et du plan de développement

Décision

Livrables : Note de lancement - Plan d'assurance qualité (version 4) - Dossier de spécifications fonctionnelles détaillées des postes de travail et de l'ergonomie - Dossier de spécifications fonctionnelles détaillées des procédures en fonctionnement normal - Dossier de spécifications fonctionnelles détaillées des procédures en fonctionnement dégradé ou de secours - Dossier de spécifications fonctionnelles détaillées des procédures d'interfaces et de reprise des données - Dossier de validation technique - Dossier d'appréciation actualisé - Plan de développement (version 4) - Rapport de synthèse

Remarque : tous les modèles des niveaux conceptuel, organisationnel, logique ainsi que le MPT sont alors complets (tandis que le MPD n'est pas encore commencé)

Conception technique détaillée

Lancement

Conception de l'architecture logique répartie du logiciel

Conception de l'architecture physique du logiciel

Conception technique détaillée des objets et composants logiciels

Préparation des tests internes

Actualisation du dossier d'appréciation et du plan de développement

Décision

Livrables : Note de lancement - Plan d'assurance qualité (version 5) - Dossier de conception de l'architecture logique répartie du logiciel - Dossier d'architecture physique du logiciel - Dossier de spécification technique détaillée du logiciel - Plan de tests interne - Dossier d'appréciation actualisé - Plan de développement (version 5) - Rapport de synthèse

Réalisation

Lancement

Codage, documentation, tests unitaires

Assemblage, tests d'intégration

Qualification fonctionnelle et technique interne (par le maître d'œuvre)

Soumission à la qualification fonctionnelle et technique sur plate-forme de tests (par le maître d'ouvrage)

Prise en compte et traitement des non conformités, avec auquel cas retour à une phase ou à une étape précédente

Qualification sur plate-forme de tests (il s'agit de la réception provisoire)

Actualisation du dossier d'appréciation

Bilan (par le maître d'œuvre)

Livrables : Note de lancement - Plan d'assurance qualité (version 6) - Objets et composants primaires - Documentations techniques de maintenance et de mise en exploitation - Rapports de qualifications interne et sur plate-forme de tests - Procès verbal de réception provisoire

Préparation à la mise en œuvre (par le maître d'ouvrage)

Lancement

Élaboration du plan de mise en œuvre et du plan d'assurance qualité de la mise en œuvre

Choix, commande, installation et tests des moyens techniques et logistiques

Définition et préparation de la mise en place des structures et postes

Élaboration du plan de communication et assurer l'information

Rédaction du plan de tests externes, des manuels utilisateurs et d'exploitation

Préparation des supports de formation et assurer la formation

Préparation de la migration

Livrables : Note de lancement - Plan d'assurance qualité (version 3) - Plan de mise en œuvre - Dossier d'organisation - Plan de communication - Plan de tests externes - Manuels utilisateurs - Supports de formation - Manuels d'exploitation - Dossier de migration

Mise en œuvre (par le maître d'ouvrage)

Basculement

Qualification (sur site pilote)

Montée en régime et exploitation sous contrôle

Réception définitive

Déploiement sur les autres sites

Élaboration du bilan du projet

Livrables : Rapport de qualification sur site pilote - Procès verbaux de réception définitive - Rapport de bilan

OMT : Object Modeling Technique

Bibliographie

J. RUMBAUGH et al., Object oriented modeling and design, Englewood cliffs, Prentice hall, 1991 (tomes 1 et 2)

Historique

Méthode inventée dans le Centre de Recherche et Développement de General Electric (fin des années 1980)

Succès actuellement en Europe et en France

Projets de fusion avec la méthode (H)OOD de G. Booch

OMT utilise tous les concepts de l'approche objet

Objet : objet, agrégation, actif/passif, concurrence

Relation entre objets : association, composition, classification (généralisation/spécialisation)

Communication entre objets : message, événement/état/transition

Architecture système : décomposition en sous-systèmes et en modules

Encapsulation des données et des services

OMT couvre toutes les phases du cycle de développement du logiciel

Analyse : modélisation du monde réel en vue de sa compréhension

Construction des modèles objet, dynamique et fonctionnel

Intégration des trois modèles

Conception : définition d'une architecture système, solution informatique au problème

Conception système

Conception préliminaire

Réalisation : choix du SGBD et du langage de programmation

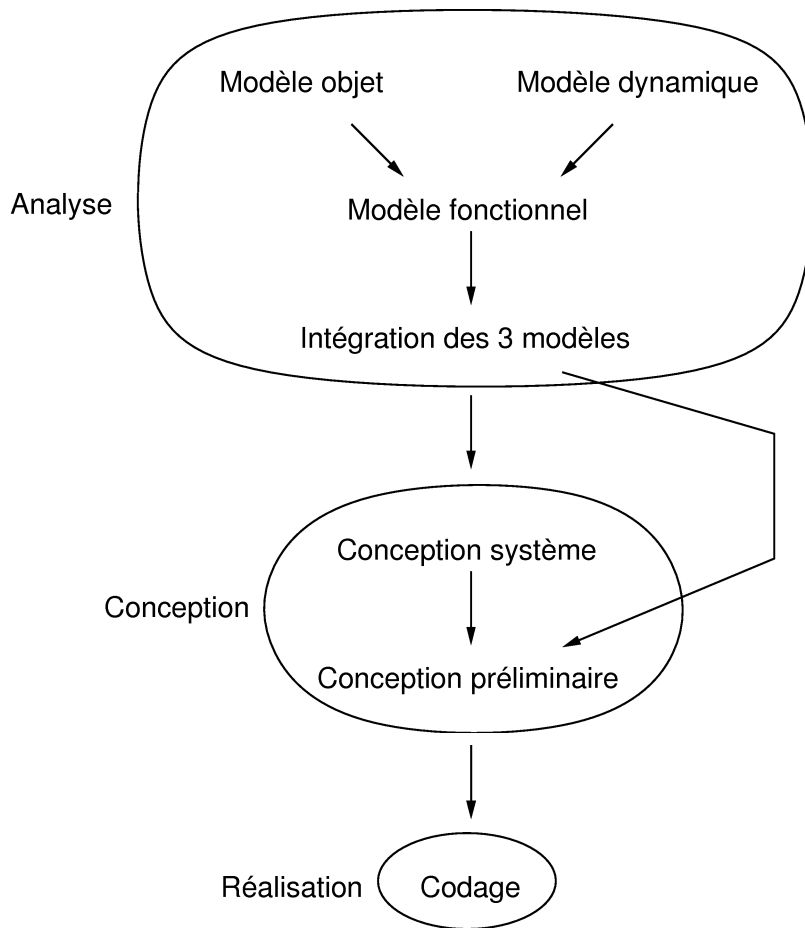
Codage

OMT utilise trois modèles

Modèle objet : description de la structure statique du système

Modèle dynamique : modélisation de l'interaction entre le système et l'extérieur et entre les objets du système

Modèle fonctionnel : spécification des fonctionnalités du système



Modèles \ Phases	Analyse	Conception
Objet	structure statique dictionnaire des données	architecture du système conception objet
Dynamique	contrôle externe du système	contrôle interne du système
Fonctionnel	décomposition fonctionnelle diagrammes de flots de données de haut niveau	décomposition des processus

Rôle

Modélisation du problème en vue de sa compréhension

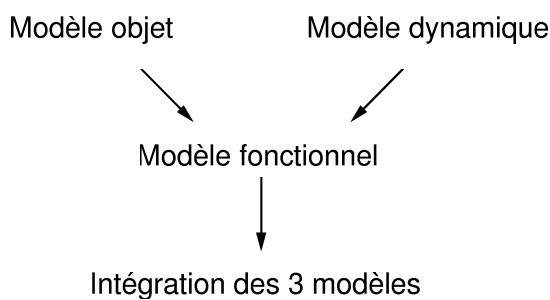
Entrées

Description du problème, entretiens avec les utilisateurs, connaissance du domaine

Sorties

Modèles objet (structure statique), dynamique (réaction aux événements extérieurs) et fonctionnel (processus de transformation de données)

Étapes



Construction du modèle objet

Rôle : vue statique de l'application durant l'analyse

Construction du modèle dynamique

Rôle : vue dynamique de l'application durant l'analyse

Construction du modèle fonctionnel

Rôles : description des dépendances fonctionnelles, des transformations des données, et des opérations réalisant une fonctionnalité

Remarque : créé après les modèles objet et dynamique

Intégration des trois modèles

Remarque

La création des trois modèles n'est pas linéaire :

d'abord travailler au niveau du domaine,
et ensuite au niveau de l'application

Construction du modèle objet

- Constituer un dictionnaire des données

 - Description de chaque entité du modèle

 - Donner des exemples de classes et d'associations

- Trouver les classes (et les généralisations)

 - Une classe du domaine représente une entité physique ou conceptuelle du domaine concernée par l'application ou une entité externe qui communique avec l'application

 - Une classe de l'application représente une entité du monde réel, une interface utilisateur, un superviseur, un contrôleur de périphérique spécifique, etc.

 - Généraliser : placer les attributs, opérations et associations communs au plus haut niveau de la hiérarchie des sous/super-classes

- Trouver les associations (et les assemblages)

 - Une association représente une dépendance structurelle entre les classes

- Déterminer les attributs des classes et des associations

- Créer les modules

 - Structurer le modèle en catégories de classes

- Vérifier le modèle

 - Rechercher les classes, associations et attributs omis, erronés ou en trop

 - Tester le modèle objet à partir des interrogations qui peuvent être faites sur le système

Construction du modèle dynamique

- Créer les scénari pour chaque cas d'utilisation du système

 - Cas d'utilisation : acteurs extérieurs interagissant avec le système, différentes utilisations du système pour chaque acteur

 - Scénari : bon fonctionnement, interaction avec l'extérieur, cas limites, mauvais fonctionnement

- Identifier les événements externes (et les événements d'erreurs)

 - Paramétrer les événements et en identifier les source et cible

- Réaliser les diagrammes de flots d'événements pour l'application

 - Un diagramme de suivi d'événement pour chaque scénario

 - Diagrammes d'événements entre objets et entre classes

- Faire un diagramme d'états pour chaque classe du modèle objet

 - (mais uniquement une liste des événements en entrée et en sortie pour les classes qui répondent à un événement indépendamment de leur histoire passée)

- Vérifier les diagrammes d'états

 - Vérifier les scénari

 - S'assurer de la cohérence de l'application en concurrence

Construction du modèle fonctionnel

- Faire le diagramme de contexte

 - (le système est considéré dans sa totalité, avec ses valeurs en entrée et en sortie)

- Réaliser la décomposition fonctionnelle

 - Décomposer l'application en catégories fonctionnelles

 - Réaliser l'arbre de décomposition fonctionnelle

 - Donner les spécifications textuelles des fonctionnalités (entrée, sortie, tâches)

 - Décomposer chaque catégorie fonctionnelle

- Construire les diagrammes de flots de données de haut niveau

 - Faire un diagramme de flots de données pour chaque fonctionnalité de haut niveau

 - Spécifier les données en entrée et en sortie des actions et activités

- Définir les processus

 - Spécifier les processus

 - Décomposer les processus complexes (des diagrammes de flots de données de haut niveau jusqu'à l'obtention d'opérations rattachées à des objets)

Intégration des trois modèles

- Exécuter les scénari sur les 3 modèles

- Définir les opérations

 - Pour le modèle objet, une lecture ou écriture d'attributs ou de liens

 - Pour le modèle dynamique, soit un événement, soit une action ou une activité

 - Pour le modèle fonctionnel, un processus

- Relier les trois modèles

 - Pour le modèle objet : un diagramme d'états pour chaque classe active

 - Pour le modèle dynamique : encapsulation des objets en définissant leurs attributs et leurs opérations, une transition (événement, action, activité) devient une opération

 - Pour le modèle fonctionnel : décomposition des processus jusqu'à l'obtention de fonctions réalisables

Rôles

- Définir une solution au problème
- Définir une architecture du système
- Prendre en compte les détails internes au système
- Enrichir les trois modèles issus de l'analyse

Entrées

- Les modèles objet, dynamique et fonctionnel issus de l'analyse

Sorties

- Les modèles objet, dynamique et fonctionnel enrichis

Étapes

Conception système



Conception préliminaire

La conception système

Rôle : choisir une solution

La conception préliminaire

Rôles : description complète des classes et des associations, écriture des algorithmes des méthodes réalisant les opérations

Stratégie : intégration au modèle objet des informations des trois modèles issus de l'analyse et du résultat de la conception système

La conception système

Estimer les besoins en ressources et performances (pour le choix d'une solution)

Décomposer le système en sous-systèmes

Un sous-système est un ensemble de classes et d'associations ayant une interface avec le reste du système

Architecture d'un système : des sous-systèmes de haut niveau (système spécifié) aux sous-systèmes de bas niveau (ressources utilisées)

Type d'architecture : organisation verticale ou horizontale, architecture fermée ou ouverte pour une organisation horizontale

Stratégie : déterminer la frontière du système et décomposer le système en sous-système

Gestion de la concurrence

Objet actif (c'est à dire autonome) ou passif

Responsabilité des objets : placer par exemple les objets concurrents dans des sous-systèmes différents, définir les échanges des objets ayant des activités exclusives

Gestion des données (persistantes)

Choix sur la mise en œuvre du contrôle

Contrôle interne ou externe

Procédural ou gestion d'événements séquentiels ou concurrence

La conception préliminaire

Définir les opérations et les attributs à partir des modèles objet, dynamique et fonctionnel

Vérifier que les opérations couvrent les événements externes et l'arbre de décomposition fonctionnel

Choisir les algorithmes (et les structures de données) des opérations

Ajouter les classes nécessaires à la réalisation (encapsulées dans des modules)

Réutiliser des classes (généricité)

Réaliser le contrôle externe : mise en œuvre du modèle dynamique

Exemple de mise en œuvre procédurale : une procédure exécutant séquentiellement le chemin principal d'un diagramme d'état

Affiner l'héritage

Réorganiser les classes (avec leurs attributs et opérations)

Concevoir les objets et les associations (afin d'enrichir le modèle objet avec des objets de réalisation)

Ajouter des informations redondantes pour optimiser les accès

Transformer des associations en attributs (représentant les liens)

Déterminer la portée (publique ou privée) des attributs et opérations des objets

Créer et enrichir les modules

Un sous-système représente un ensemble de modules

Rôle

Programmer les décisions de conception

Caractéristiques (d'un langage orienté objet)

Héritage

Généricité

Encapsulation

Typage

Polymorphisme

Modularité

Communication

etc.

Préceptes (de la programmation orientée objet)

Réutilisabilité

Robustesse

Évolutivité

Maintenabilité

etc.

Étapes

Définition des classes (avec leurs attributs et opérations)

Gestion de la création/suppression des objets (permanents ou non, de façon explicite ou implicite)

Appels des opérations (avec leurs arguments en précisant leurs modes de passage)

Héritage (statique ou à l'exécution)

Association et assemblage : pointeur sur un objet ou ensemble de pointeurs sur objets, dépendances possibles