

LINGE1322: Informatique : analyse et conception de systèmes d'information

Notes

Année académique 2014-2015

Part I

Introduction à l'informatique de gestion

1 Problématique de l'informatique de gestion

Différence entre connaissance, information et donnée :¹

- **Donnée:** une donnée est le résultat direct d'une mesure. Une donnée seule ne permet pas de prendre une décision sur une action à mener.
- **Information:** une information est une donnée à laquelle un sens et une interprétation ont été donnés. Une information permet à un responsable opérationnel de prendre une décision sur une action à mener.
- **Connaissance:** La connaissance est le résultat d'une réflexion sur les informations analysées en se basant sur :
 - ses expériences, ses idées, ses valeurs, les avis des autres personnes consultées pour l'occasion
 - sa propre expertise et celle de ses pairs

La connaissance permet aux responsables de confronter les informations au contexte de l'organisation et à d'autres contextes externes à l'organisation afin d'avoir une meilleure connaissance et une interprétation élargie des phénomènes mis en lumière par ces informations.

2 Les systèmes d'information

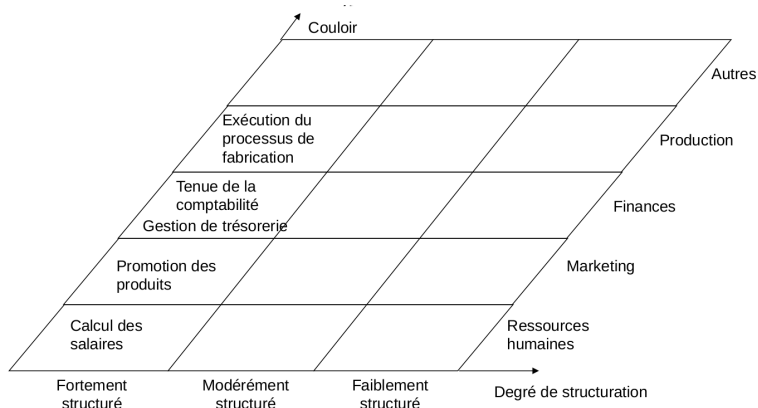
Notion de gestion intégrée

Pyramide des trois niveaux

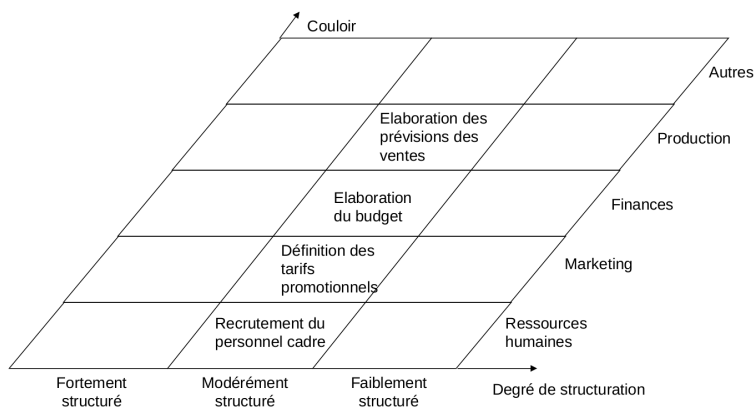
- Niveau stratégique
- Niveau décisionnel
- Niveau opérationnel

¹Source : http://www.itilfrance.com/index.php?pc=pages/docs/itilv3-03/117-02.inc&pg=menu_itilv3.inc&pt=Concept-cl%E9%20Donn%E9es-%3E%20Information%20-%3E%20Connaissance%20-%3E%20Sagesse

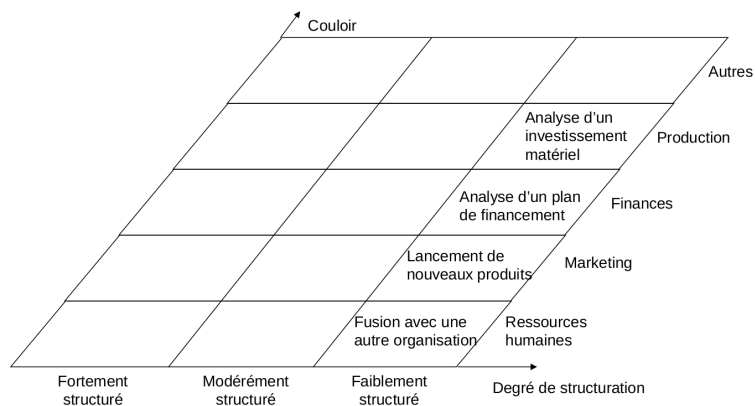
2.1 Niveau opérationnel



2.2 Niveau décisionnel



2.3 Niveau stratégique



Comparaison des 3 niveaux :

	Opérationnel	Décisionnel	Stratégique
Structuration	Forte	Modérée	Faible
Exigence	Différée	Légèrement différée	Temps réel
Interactivité	Faible	Modérée	Élevée
Types de problèmes	Administratifs, gestion courante	Productivité, efficacité	Amélioration de la qualité
Terme	Court	Moyen	Long

3 Fichiers

3.1 Typologie des fichiers

- La mémoire est la notion de base d'un SI
C'est un ensemble structuré de fichiers reliés par des relations
- Opération : création, suppression, modification, consultation
- Types
 - Fichiers permanents
 - Fichiers de mouvement
 - Fichiers de manoeuvre
 - Fichiers archives

3.2 Structure des fichiers

- Fichier : ensemble structuré d'enregistrements
- Enregistremet : ensemble structuré de champs
 - Peut être logique ou physique
 - Possède éventuellemnt une structure décrite par
 - * Nom
 - * Taille
 - * Type de données

3.3 Organisation et exploitation des fichiers

- Organisation : disposition physique
- Exploitation : manière d'accéder aux données

4 Modèles de base de données

4.1 Principes et objectifs des bases de données

Objectifs

- On doit chercher des enregistrements suivants un ou plusieurs critères ou clés dans un ou plusieurs fichiers
- Exemple: ouvrages d'une bibliothèque

Principes

- Intégration, interrogation, partageabilité, sécurité

Exemple : catalogue électronique de produits

Solutions possibles :

- **Solution statique** : codée directement en HTML
- **Solution dynamique sur fichiers** : générée automatiquement sur un, plusieurs fichiers
- **Solution dynamique sur base de données** : idem, avec un système de gestion de base de données
- **Solution par construction** : par l'utilisateur
- **Solution décentralisée**

4.2 Problèmes liés à l'utilisation de bases de données

- Stockage : indépendant des applications
- Cohérence : assurée pour toutes les données stockées
- Intégrité : assurée pour toutes les données stockées
 - moyens de sauvegarde (totale, incémentale)
 - moyens de régénération (ancienne, trace)

Part II

Le modèle du domaine

1 Introduction au modèle du domaine

Le diagramme des classes est

- Un modèle du domaine
Un modèle qui définit les objets que l'utilisateur final peut voir , accéder et manipuler à travers le système avec les relations entre ces objets
- Un modèle structurel
Une vue d'un système qui met l'accent sur la structure des objets, avec leurs classificateurs, leurs relations, leurs attributs et leurs opérations
- Un modèle du domaine est plus qu'un modèle de données (modèle de données \in modèle du domaine)

2 Identification des classes

Classe : description d'un ensemble d'objets qui ont même

- Sémantique
- Structure

Une classe constitue un canevas de sémantique et de structure à partir duquel des objets individuels peuvent être instantiés

- Classe : voiture
- Objets : ma propre voiture, la voiture du voisin

Description d'un ou de plusieurs objets de manière uniformisée, avec moyens de gérer les objets de cette classe.

Objet : une abstraction que quelque chose présent dans le domaine d'intérêt qui doit refléter une capacité du système d'information à maintenir cette information et à interagir avec elle.

Diagramme de classe : moyen de décrire les classes en vue de leur implémentation

- Est le modèle le plus fréquent
- Fournit les fondements nécessaires pour les liens avec les autres modèles
- Ne décrit que la partie structurelle, statique

Règles pour identifier des objets et classes :

- Rôles joués dans l'organisation
(client de la banque, ...)

- Choses ou événements à conserver
(connexion à un site web, ...)
- Appareils physiques
(ordinateur personnel)
- Règles de gestion
(calcul du taux d'intérêt, ...)
- Sites physiques
(chantiers versus siège, ...)

3 perspectives de Fowler

- Conceptuelle : le diagramme représente le domaine étudié indépendamment de tout système
- Spécification : quelles sont les conditions, l'interface offerte avec l'extérieur
- Implémentation : chaque classe est implémentée directement

Comment trouver les classes ?

- Observer les pratiques et comportements
- Interviewer les intervenants
- Consigner dans un scénario textuel
- Vérifier avec les classes préalablement identifiées
 - En vue de préserver la cohérence
 - En vue de maximiser la réutilisabilité
 - En vue d'optimiser la structuration
- Considérer d'autres SI
- Lire les manuels et règles de gestion
- Prototyper le système

3 Identification des attributs

Attribut

- Propriété commune des objets d'une classe
- Toute donnée d'un objet d'une classe possédant sa propre valeur
- Doit représenter un concept tangible, concret, atomique avec une valeur, un ensemble de valeurs
 - Bons exemples : nom, prénom, adresse
 - Mauvais exemples : esthétique, ce qu'on fait dans la journée
- Doit être placé dans la classe qui lui correspond, qui la décrit le mieux
- Dans une structure d'héritage, les attributs partagés par un certain nombre de sous-classes doivent être logés au niveau d'une super-classe
 - Les attributs génériques dans la super-classe
 - Les attributs dans la sous-classe

Déclaration d'un attribut [visibilité] nom [card] [:type] [= valeur initiale][{props...}]

Utilisation plus limitée que dans une modélisation de type entité-association

Type

- A réserver à des types primitifs (entier, réel) ou des “structures de données” simples (string, date)
- Ne pas utiliser de structures de type tableau, liste, ... Sauf exception...

Nature

- Obligatoire / facultatif
- Simple / répétitif
- Élémentaire / décomposable

Visibilité des attributs

- Public (+) : visible et utilisable par toute autre classe (utilisation très limitée)
- Protégé (#) : visible et utilisable par toute spécialisation de la classe
- Privé (-) : visible uniquement par la classe elle-même
- Dérivé (/) : calculé à partir d'autres attributs
- Autre (?) : inconnu
- Non spécifie () : indéterminée

Recommandations d'identification

- Questions anthropomorphiques
 - Comment suis-je décrit en général ?
 - Comment suis-je décrit dans le domaine d'intérêt ?
 - Comment suis-je décrit vis-à-vis des responsabilités assignées au SI ?
- Questions de responsabilités
 - Que dois-je savoir ?
 - Quel états dois-je sauvegarder dans le temps ?
 - Quels sont les états significatifs ?

Identification d'une classe

- Identifiant : moyen de désigner de manière univoque n'importe quel objet d'une classe
- Aucune paire d'objets d'une même classe ne partage le même identifiant
- Attribut simple : noma, n° chassis, code, référence
- Attributs multiples : n°chassis, plaque immatriculation
- Attributs groupés : nom + prénoms + localité

4 Identification des méthodes

Une méthode

- Exprime un comportement particulier dont une classe peut faire preuve
- Exprime ce qu'une classe peut faire en offrant une série de services à l'extérieur, pour les autres classes
- Synonymes = fonction, opération, service, ...
- Autorise le changement d'état de tout objet par l'appel de méthode

Un état

- Est représenté par les valeurs collectives d'un ensemble d'attributs d'une classe d'objets
- Tout changement résulte d'une invocation de méthode

Méthodes simples

- **Méthodes CRUD**
 - Create : crée et initialise un nouvel objet
 - Read : accède, obtient la valeur d'un attribut quelconque d'un objet
 - Update : assigne une valeur à un attribut quelconque d'un objet
 - Delete : déconnecte, rend à nouveau disponible un objet
 - ==> Devraient toujours être présentes
- Connect : appelle une méthode d'un autre objet
- Méthodes complexes
 - Calculate : calcule un résultat à partir des valeurs des attributs d'un objet
 - Monitor : surveille l'état d'un système ou appareil extérieur
 - * Est approprié pour tous les appareils d'acquisition et de restitution des données connectés au SI
 - * Est approprié pour tous les systèmes de contrôle
 - * Est souvent complété par
 - Initialise : initialise le système de surveillance
 - Terminate : clôture le système de surveillance
 - Listen : écoute les événements qui y surviennent
 - Re-initialise : réinitialise le système suite à un problème

2 stratégies d'identification

- Introspection : on observe comment les objets changent au cours du temps
- Interconnexion : on regarde comment un objet est connecté à d'autres objets durant sa durée de vie

Déclaration d'une méthode [visibilité] nom [(params)] [: type] [{props ...}]
params := [in | out | inout] nom [: type] [=default] [{props ...}]

5 Identification des relations

Notation des relations

- En principe, une relation est implicitement bidirectionnelle, mais on peut la diriger

Multiplicité des relations [min][..][max]

- 0 ou plus : 0..* ou *
- 1 à 40 : 1..40
- 1 ou plus : 1..*
- Exactement 5: 5

Notes :

- 0..0 n'a pas de sens (c'est comme si il n'y avait pas de relations)
- 0..1 : équivalent à un booléen
- 1..1 ou 1 : exactement 1, possible mais peu intéressant
- m..n : cas le plus général ou $m \leq n$

4 types fondamentaux de relations

- **Dépendance** : pour exprimer une dépendance fonctionnelle entre deux classes
 - Une dépendance exprime qu'une classe en utilise une autre
 - Tout changement dans une classe affecte l'autre
- **Généralisation** : pour exprimer l'héritage d'une classe à l'autre
 - Se traduit par une relation d'héritage
 - Définit une relation "est-un" entre classes
 - Une classe peut hériter des attributs et/ou des méthodes de l'autre classe
 - La généralisation implique la possibilité de substitution
- **Association** : pour exprimer les relations proprement dites, au sens large
- **Aggrégation** : pour exprimer le lien tout-partie

Héritage

- Constitue une hiérarchie de classes dont les nouvelles classes forment des spécialisations de classes déjà existantes
- Permet une meilleure structuration et compréhension des classes
- Améliore la réutilisabilité et l'extensibilité
- 2 types d'héritage
 - **Héritage simple** : une sous-classe hérite d'une et d'une seule super-classe
 - **Héritage multiple** : une sous-classe hérite simultanément de plusieurs super-classes

Hiérarchie de classes Les classes, organisées en hiérarchie suivant le relation d'héritage entre elles, définissent un arbre d'héritage

Relation “tout-partie” Une relation tout-partie indique qu’une entité est composée d’une ou de plusieurs parties qui sont elles-mêmes des instances de cette entité ou d’une autre.

- Permet la représentation d’hierarchie de composition
- Exemple: une voiture est composée d’un moteur, d’un châssis, de sièges, de roues, ...
- Considérer chaque objet comme un tout
 - Pour chaque composant potentiel, se poser la question :
 - * Appartient-il au domaine d’intérêt ?
 - * Convient-il aux responsabilités du SI ?
 - * Capture-t-il juste une valeur ou bien est-il un concept véritable ?
 - Si valeur, insérer un attribut
 - Si concept, insérer une classe
 - * L’abstraction est-elle utile ?
 - Considérer aussi chaque objet comme une partie

Relation de composition “est partie de” (is part of)

- Les classes et les objets qui en sont instantiés peuvent être imbriqués dans d’autres classes
- Exemple :
 - Le corps humain est composé de deux poumons
 - La classe “poumon” est partie de la classe “corps humain” avec 2 instances
 - Toute instance de corps humain a donc 2 instances de poumon

Classe d’association

Pour qualifier la nature d’une association

6 Résumé synoptique

Validité d’un diagramme de classe

- Complétude : toute l’information relative au domaine d’intérêt est-elle présente ?
- Cohérence : toute l’information présente est-elle représentée de manière compatible avec la réalité et avec le reste du diagramme ?
- 7 péchés de spécification de Meyer
 - Bruit
 - Silence
 - Contradiction
 - Surspécification
 - Ambiguïté
 - Redondance
 - Incohérence

Patron de conception

- Un patron de conception est une structure récurrente qui peut être réutilisée dans différents cas en conservant la même structure, mais adaptée au cas particulier

- Un patron aide à structurer l'approche pour un nouveau cas à partir de cas précédents
- Patrons
 - Pattern 1 : entité-membre
 - * Ce patron décrit une situation générale dans laquelle des membres sont groupés dans une entité de travail
 - Pattern 2 : Acteur-participant
 - * Un acteur peut jouer un certain rôle en tant qu'acteur dans une organisation
 - Pattern 3 : Participant-Transaction
 - * Une transaction implique des participants
 - Pattern 4 : Transaction-Lieu
 - * Une transaction se situe dans un lieu
 - Pattern 5 : Transaction-Item
 - * Une transaction est constituée d'une série d'items spécifiques
 - Pattern 6 : Associé - autre associé
 - * Des associés sont des objets qui se connaissent l'un l'autre, mais sans avoir besoin de connaître des données sur cette association ou d'historique (pas de classe d'association)
 - Pattern 7 : Hiérarchie de composants

Part III

Modèle de structuration des données

1 Définition et objectifs du schéma de décomposition du projet

Définition

Ensemble de concepts et de règles destinés à structurer un problème global en une hiérarchie d'agrégats de traitements

Objectifs

- Représenter les traitements à différents niveaux de détails
- Procéder à une structuration d'un projet par raffinements successifs
- Décomposer de manière arborescente

2 Concepts de base

Le concept de projet

- Définition
 - Partie du problème global qui fait l'objet d'une étude d'opportunité et d'une analyse conceptuelle
- Critères d'identification
 - Différence d'objectifs opérationnels de l'organisation
 - Différence d'objectifs de gestion

Le concept d'application

- Définition
 - Traitement quasi autonome par rapport aux autres, qui consitue une unité de planningdans la gestion du projet
- Critères d'identification
 - Interactionfaible avec les autres applications
 - Ponctualité de la communication
 - Échange d'agrégats informationnels
 - Homogénéité des messages
 - Homogénéité des flux de données

2.1 Le concept de phase

- Définition
 - Traitement possédant une unité spatio-temporelle d'exécution, dans une cellule d'activités
- Critères d'identification
 - Dus à l'unité spatiale (inter, manuelle, auto, méca)
 - * Changement de lieu
 - Poste de travail
 - Autre endroit (siège agence)
 - Parcours
 - * Changement de ressources
 - Personne, matériel, information
 - Dus à l'unité temporelle
 - * Existence d'une interruption
 - * Existence d'un point d'attente
 - décision (ligne correcte) / accumulation (p. chaque)
 - * Permanence de l'unité d'exécution
 - * Périodicité de l'exécution
- Phase interactive = tâche pour l'utilisateur
 - Exemples :
 - * Enregistrer un bon de commande client
 - * Imprimer le courrier de masse

Représentation graphique

manuelle

PréparationBonDeCommande

interactive

EnregistrementBonDeCommande

automatique

ImpressionStatistiquesBonCde

mécanique

MiseSousPliCourrierCentral

Présentation standard des phases

N°	Nom de la phase	Préd	Définition	Justification	Nature
1	Constit-dossier	-	Constitution du dossier dans l'agence		Manuelle

- **N°** : numéro de la phase dans l'application
- **Nom de la phase** : dénomination abrégée de la phase
- **Préd** : prédécesseur(s) éventuel(s)
- **Définition** : description détaillée de la phase
- **Justification** : liste des critères d'identification de phases concernés
- **Nature** : manuelle, interactive, automatique ou mécanique

2.2 Le concept de fonction

Définition et objectifs

- Correspond au niveau le plus élémentaire
- Résulte de la décomposition d'une phase
- Est associée à un objectif et un comportement considérés comme élémentaires par l'organisation
- Traduit un objectif de gestion élémentaire
- Effectue un échange indécomposable avec l'utilisateur
- Correspond à une action minimale et cohérente

Critères d'identification

- Critère d'agrégabilité
 - Chaque fonction doit correspondre à un sous-objectif de la phase
- Critère d'ergonomie
 - Chaque fonction devrait être vue comme un service
 - L'échange de données doit être indécomposable (exécution sans interruption)
- Critère de cohérence
 - Chaque fonction doit préserver l'intégrité des données

3 Rédaction d'un schéma de décomposition

3.1 Critères d'identification / "trucs"

Pour découvrir une phase

- Changement de lieu
 - Parcours, passer du siège à l'agence, changer de bureau, aller au poste d'impression
- Changement de ressource
 - De type personne : une autre personne

- De type matériel :
 - * Tout changement de nature (manuelle / interactive / automatique / mécanique)
 - * Terminaux, imprimanté, réseau Internet
- De type d'information : un autre type de dossier
- Point d'attente de type décision
 - Décision humaine
 - Ligne correcte, commande enregistrée, si ..., acceptée, il peut dans ce cas ..., il approuve
- Point d'attente de type accumulation
 - Pour chaque produit, lorsque tout est arrivé, on attend un paquet pour l'envoyer, déclenche après réception de 10 paquets
- Différence de périodicité
 - À chaque réapprovisionnement - chaque fois qu'un produit est disponible
 - En fin de semaine - au moins 20 min. en attente

Pour découvrir une fonction

- Identifier les conditions minimales qui :
 - Réalisent un sous-objectif de la phase
 - Conservent la cohérence du SI
- Identifier les actions élémentaires sur le SI (créer supprimer, modifier, consulter, ...)
- Identifier les entrées nécessaires pour effectuer ces actions
- Toutes les entrées de la fonction sont-elles des entrées de la phase ?
 - Si oui => fin
 - Si non
 - * Identifier une fonction
 - Dont l'objectif est de produire cette entrée
 - Qui conserve la cohérence du SI
 - * Reprendre deux points plus haut

3.2 Validation d'un schéma de décomposition

Règles de complétude

- CP11. Tout traitement doit posséder
 - Un nom
 - Une définition
 - Un objectif
 - Un niveau donné dans l'arborescence
 - Une relation d'appartenance avec un supérieur (sauf...)
 - Une relation d'appartenance avec un inférieur
- CP2. Le schéma doit vérifier la nomenclature

- Tout projet comprend au moins une application
- Une application appartient à un et un seul projet
- Une application comprend au moins une phase
- Toute phase appartient à une seule application
- Toute phase comprend au moins une fonction
- Toute fonction appartient à une seule phase

Règles de cohérence

- CH1. Unicité des noms et absence d’homonymes
 - Tout traitement doit avoir un nom unique
- CH2. Absence de synonymes
 - Pas de traitement présentant une même sémantique mais portant des noms différents (exemple : quit / exit)

Part IV

Modèle de la dynamique de traitements

1 Définition et objectif du schéma de la dynamique des traitements

Définition

- Modèle représentant les conditions d’exécution et d’enchaînement des traitements en vue de caractériser le comportement du système informatique

Objectifs

- Préciser la cause : la survenance d’un événement
- Préciser l’effet : le déclenchement de processus
- Utiliser les structures algorithmiques de base
- Préciser 2 niveau d’enchaînement
 - Application en phases, phase et fonction

2 Concepts de base

2.1 Le concept de message

- Représente une collection structurée d’informations véhiculées dans le système informatique
- Possède un contenu informationnel
- Est associé à une présentation physique
- Est caractérisé par un type
 - Externe : échangé avec l’environnement (dest/prov)
 - Interne : échangé à l’intérieur du système

- Est représenté graphiquement par



2.2 Le concept de processus

- Représente l'exécution d'un traitement dans le système informatique
- Possède deux changements d'état
 - Le déclenchement : création du processus
 - La terminaison : destruction du processus
 - Est représenté graphiquement pas (un rectangle tout bête)
- Possède une nature
 - Manuelle
 - Interactive
 - Automatique
 - Mécanique

PréparationBonDeCommande

EnregistrementBonDeCommande

ImpressionStatistiquesBonCde

MiseSousPliCourrierCentral

2.3 Le concept d'événement

- Représente un changement d'état qui survient à un moment donné de l'évolution du système informatique
- Correspond à un stimulus auquel le système doit réagir
- Possède deux survenances possibles
 - À la génération d'un message
 - À un changements d'état du processus
 - * Déclenchement
 - * Terminaison
- Est représenté graphiquement par une flèche

2.4 Les mécanismes dynamiques

1. Message externe

- Le message externe “commande client” forme un événement qui en génération déclenche le processus “enregistrement commande”

2. Événement temporel

- Le message horaire “il est 17h30” forme un événements temporel qui en génération déclenche le processus “expédition courrier”

3. Enchaînement séquentiel :

- 1e forme
 - Le processus “enregistrement commande” en terminaison déclenche le processus “préparation réquisition”
- 2e forme
 - Le processus “enregistrement commande” génère l’événement “commande enregistrée” qui en génération déclenche le processus “préparation réquisition”

4. Enchaînement parallèle

- 1e forme
 - Le processus “réception livraison” en terminaison déclenche le processus “contrôle qualité” **ET** le processus “contrôle quantité”
- 2e forme
 - Le processus “réception livraison” génère l’événement “livraison à contrôler” qui en génération déclenche le processus “contrôle qualité” **ET** le processus “contrôle quantité”
- 3e forme
 - Le processus “réception livraison” génère les événements “qualité à contrôler” **ET** “quantité à contrôler” qui en génération déclenchent respectivement les processus “contrôle qualité” et “contrôle quantité”

5. Enchaînement convergent

- 1e forme
 - Le processus “enregistrement versement” **OU** le processus “enregistrement paiement” en terminaison déclenche le processus “impression solde”
- 2e forme
 - Le processus “enregistrement versement” **OU** le processus “enregistrement paiement” génère l’événement “solde à imprimer” qui en génération déclenche le processus “impression solde”
- 3e forme
 - L’événement “versement enregistré” généré par le processus “enregistrement versement” **OU** l’événement “paiement enregistré” généré par le processus “enregistrement paiement”, en génération déclenche le processus “impression solde”

6. Enchaînement synchronisé par conjonction

- 1e forme
 - Le processus “contrôle qualité” **ET** le processus “contrôle quantité” contribuent au point de synchronisation “fin contrôle” qui en réalisation déclenche le processus “enregistrement livraison”

- 2e forme
 - L'événement "bordereau 1" généré par le processus "contrôle qualité" **ET** l'événement "bordereau 2" généré par le processus "contrôle quantité" contribuent au point desynchronisation "fin contrôle" qui en réalisation déclenche le processus "enregistrement livraison"
7. Enchaînement synchronisé par disjonction
- 1e forme
 - Le processus "enregistrement commande" **OU** le processus "sélection commande" contribuent à la réalisation du point de synchronisation "termination commande" qui en réalisation déclenche le processus "traitement commande"
 - 2e forme
 - L'événement "bon réquisition" **OU** l'événement temporel "fin de journée" contribuent au point de synchronisation "constitution série" qui en réalisation déclenche le processus "préparation série"
8. Enchaînement groupé
- L'événement "bon réquisition" contribue au point de synchronisation "constitution série" qui en réalisation déclenche le processus "préparation série"
9. Enchaînement synchronisé: attente par conjonction
- L'événement "livraison" en génération déclenche le processus "contrôle qualité" **ET** le processus "contrôle quantité" qui contribuent au point de synchronisation "fin contrôle" qui en réalisation déclenche le processus "enregistrement livraison"
10. Enchaînement synchronisé : attente par accumulation
- L'événement "bon commande" en génération déclenche le processus "vérification ligne" qui en terminaison contribue au point de synchronisation "fin vérification" qui en réalisation déclenche le processus "mémoire commande"
11. Enchaînement synchronisé...
- avec modalité de mémorisation
 - Le processus "préparation courrier" **ET** l'événement "il est 15h", en terminaison, respectivement en génération, contribuent au point de synchronisation "début expédition" qui en réalisation déclenche le processus "expédition courrier"
 - Modalité de mémorisation : génération de "il est 15h" mémorisée pendant 2 heures
 - ... sans modalité de mémorisation
 - Le processus "préparation courrier" **ET** l'événement "il est 15h", en terminaison, respectivement en génération, contribuent au point de synchronisation "début expédition" qui en réalisation déclenche le processus "expédition courrier"
 - Modalité de mémorisation : génération de "il est 15h" non mémorisée
12. Mécanisme de sélection : déclenchement conditionnel
- Le processus "préparation commande" en terminaison déclenche le processus "enregistrement commande" si la condition "identification possible" est vérifiée
13. Mécanisme de sélection : contribution conditionnelle
- Le processus "enregistrement réquisition" en terminaison contribue au point de synchronisation "constitution série" si la condition "réquisition possible" est vérifiée

14. Enchaînement conditionnel

- 1e forme
 - Le processus “vérification commande” en terminaison déclenche le processus “enregistrement commande” si la condition “commande valide” est vérifiée, le processus “rejet commande” si la condition “commande valide” n’est pas vérifiée
- 2e forme
 - Le processus “vérification commande” en terminaison génère l’événement “commande à enregistrer si la condition “commande valide” est vérifiée, l’événement “rejet commande” si la condition “commande valide” n’est pas vérifiée
- 3e forme
 - Le processus “vérification commande” en terminaison génère le message “commande vérifiée” qui déclenche le processus “enregistrement commande” si le résultat de ce message est positif et “rejet commande” si son résultat est négatif

15. Mécanisme de duplication : déclenchement multiple

- Le processus “enregistrement réapprovisionnement” en terminaison déclenche le processus “enregistrement réquisition” pour chaque “commande différée”

16. Mécanisme de duplication : contribution multiple

- L’événement “bon à expédier” en génération contribue à plusieurs réalisations du point de synchronisation “début expédition” pour lequel le processus “enregistrement livraison” contribue en terminaison

3 Rédaction d’un schéma de la dynamique

3.1 Validation d’un schéma de la dynamique

Règles de complétude

- Règles de complétude pour les processus
 - CP1. Tout processus de niveau phase doit posséder une nature
 - CP2. Tout processus non initial doit posséder un événement déclencheur
 - CP3. Tout processus non terminal doit posséder une terminaison
- Règle de complétude pour les messages
 - CP4. Tout message non initial doit posséder une génération (exception : messages horaires)
- Règles de complétude pour les points de synchronisation
 - CP5. Tout point de synchronisation doit être spécifié par une condition de réalisation (exception: ET par défaut)
 - CP6. Tout point de synchronisation doit posséder une terminaison
- Règles de complétude pour les sélections
 - CP7. Tout point de sélection doit être spécifié par une condition de sélection
 - CP8. Tout point de sélection associé à un événement doit posséder une terminaison pour les deux alternatives possibles
- Règle de complétude pour les duplications
 - CP9. Tout point de duplication doit être spécifié par une expression de duplication

Règles de cohérence

- CH1. La réalisation de tout point de synchronisation doit pouvoir être réalisée par une combinaison d'événements initiaux
 - Contradiction entre sélection et synchronisation par conjonction
 - Contradiction entre duplication et synchronisation par accumulation
 - Contradiction entre deux synchronisations
 - Contradiction entre deux contributions à une même synchronisation
- CH2. Les événements initiaux d'un sous-schéma doivent correspondre aux événements déclencheurs des processus décrits dans le sous-schéma

3.2 Démarche de rédaction

- Démarche pour une application
 - À partir de la décomposition de l'application en phases, on représente les phases
- Démarche pour une phase
 - À partir de la décomposition de la phase en fonctions, on représente toutes les fonctions, on spécifie les messages en entrée/sortie

Part V

Modèle du flux des données

1 Définition et objectifs du diagramme des flux

Définition Modèle dépeignant un portrait partiel du fonctionnement d'un système informatique

Objectifs

- Représente graphiquement la production, la circulation et la destination des messages
- Donne une vue organisationnelle synthétique
- Offre une base commune de dialogue entre concepteurs, informaticiens et utilisateurs
- Permet de quantifier l'enchaînement de processus
- S'effectue au niveau phase uniquement

2 Concepts de base

2.1 Les concepts réutilisés

- Le concept de message
- Le concept de processus (au niveau phase)

2.2 Le concept d'unité organisationnelle

- Définition
 - Élément de la structure organisationnelle où s'exécutent des processus et où sont localisés les responsables de leur exécution
- Objectif
 - Permet de situer dans le diagramme des flux les différents lieux où les phases s'exécutent
- Exemples
 - Département, service, atelier, magasin, poste de travail

2.3 Le concept d'environnement

- Définition
 - Est constitué de tout ce qui n'appartient pas au système informatique et qui interagit avec lui via des messages externes
- Objectif
 - Permet de délimiter les frontières d'un système
- Exemples
 - Les clients et les fournisseurs avec lesquels une société interagit

3 Rédaction d'un diagramme des flux

3.1 Démarche de rédaction d'un diagramme

- RS1. Remplacer un enchaînement conditionnel par un enchaînement
- RS2. Remplacer un enchaînement synchronisé par un enchaînement séquentiel
- RS3. Remplacer un enchaînement synchronisé par un enchaînement convergent
- RS4. Remplacer un enchaînement multiple par un enchaînement séquentiel

3.2 Validation d'un diagramme des flux

- Règles de complétude
 - CP1. Toute phase d'un diagramme des flux doit
 - * Avoir un nom
 - * Recevoir un message en entrée
 - * Générer un message en sortie, sauf, éventuellement, une phase terminale
 - * Être générée par une et une seule unité organisationnelle
 - CP2. Tout message doit
 - * Avoir un nom
 - * Posséder une origine (phase/unité/environnement)
 - * Posséder une destination (phase/unité/environnement)
 - CP3. Toute unité organisationnelle doit

- * Avoir un nom
 - * [Etre responsable d'au moins une phase
- CP4. Toute composante de l'environnement doit
 - * Avoir un nom
 - * Générer ou recevoir au moins un message externe
- Règles de cohérence
 - CH1. Absence d'homonymes
 - * Chaque unité organisationnelle doit avoir un nom unique
 - * Chaque message doit avoir un nom unique
 - * Chaque phase doit avoir un nom unique
 - * Chaque composante de l'environnement doit avoir un nom unique
 - CH2. Absence de boucle
 - * Une phase ne peut pas générer un message à destination d'elle-même
 - * Il ne peut pas y avoir de circuit dont on ne peut sortir

3.3 Démarche de rédaction d'un diagramme

- Idée :
 - Partir du schéma de la dynamique
 - Répartir les phases dans les
 - * Unités organisationnelles
 - * Composantes de l'environnement
 - Simplifier l'enchaînement au moyen de 4 règles de simplification

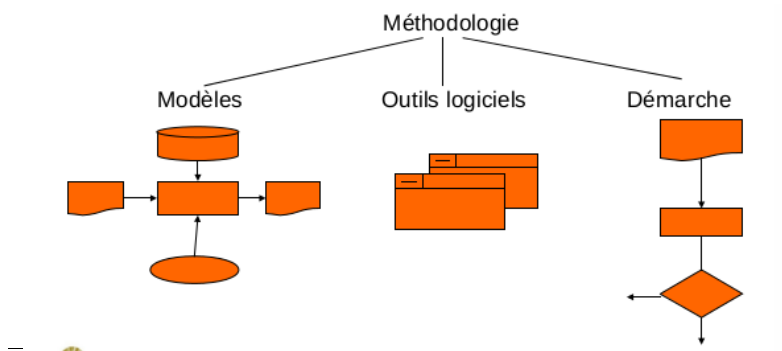
4 Le diagramme de flux

- Addition de messages horaires
- Quantification temporelle du diagramme
 - Le diagramme de flux est structuré suivant deux axes :
 - * X = unités organisationnelles parcourues dans l'ordre du flux (on commence en général par l'environnement)
 - * Y = temps
 - Quantification par valeur ou par contraire
 - * Égalité, inégalité, intervalle, ensemble
 - Portant sur
 - * La durée d'exécution ; une phase ou plusieurs phases
 - * Le délai
 - * La fréquence d'exécution
- Structuration des unités organisationnelles
 - Doit suivre la structure organisationnelle
 - Dans un diagramme de flux simple, on considère toujours que le flux couvre une organisation et son environnement

- * On ne modélise donc pas les flux extérieurs, si ce n'est sous une forme simplifiée
- L'environnement peut être ou non
 - * Groupé : une seule colonne, mais avec des cartouches différents suivants les entités
 - * Décomposé: une colonne par entité
- Exceptionnellement, deux colonnes peuvent se partager un traitement
- Plusieurs messages initiaux, terminaux sont possibles
- Représentation graphique des flux
 - Avec mise en évidence de la destination
 - * Le message appartient à l'unité destinataire
 - * Utile lorsque l'on veut distinguer les destinataires
 - (par exemple, lettre à l'assuré, au courtier, au service concerné)
 - Avec mise en évidence du destinataire
 - * Le message appartient à l'unité destinataire
 - * Utile lorsque plusieurs instances du même message sont envoyées à différents destinataires
 - (par exemple, copie assuré, copie courtier, copie compagnie)
- Représentation graphique adaptée
 - Seulement avec les messages importants
 - Seuls les flux demeurent inchangés
 - Avec icône représentant les ressources mobilisées
 - Avec représentation graphique de la nature de la tâche

5 Démarche globale

- Constitution d'une méthodologie
 - Articulation en 3 pôles



- Premier pôle d'une méthodologie : les modèles
 - * Exprime en concepts et règles une vue partielle de la réalité
 - * Se rapporte à un aspect du modèle général
 - * Se décline en modèles particuliers
 - * Forme un outil de standardisation et de communication
 - * Est soumis à des règles de
 - Complétude

- Cohérence
- Deuxième pôle : les outils logiciels
 - * Aident le concepteur à
 - Réaliser
 - Standardiser
 - Communiquer
 - Valider les modèles
 - * Sont des outils CASE
 - Passifs (éditeurs de spécifications, de modèles)
 - Actifs (maquetteurs, prototypeurs, simulateurs)
 - Automatique (sans intervention)
 - Assisté (avec intervention du concepteur)
- Troisième pôle : la démarche
 - * Consiste en un ensemble structuré d'étapes
 - * Comporte un ensemble de règles et de propositions générales pour mettre en oeuvre
 - Les modèles
 - Les outils logiciels
 - * Est régie par un guide méthodologique