

Méthodes fonctionnelles : SADT

Bernard ESPINASSE

Professeur à l'Université d'Aix-Marseille

2009

Plan

- Introduction
- Principes de base
- Représentations graphiques
- Actigrammes & Datagrammes
- Conventions simplificatrices et particularités
- Equipe SADT - Cycle auteur – lecteur

Introduction

- **SADT : Structured Analysis and Design Technics**
- SADT : technique structurée d'analyse et de modélisation
- **D.T. Ross (1972)**
- **marque déposée de SOFTECH (USA)**, introduit en France par IGL (1982)
- dérivé de SADT : **IDEF0**
- **bibliographie complémentaire :**
 - D.T.Ross, Structured Analysis : A Language for Communicating Ideas, IEEE Transactions, Software engineering, Vol SE-3, No. 1
 - D.T. Ross, K.E. Schoman, Structured Analysis for Requirements Definition, IEEE Transactions, Software Engineering, Vol SE-3, No. 1
 - SADT: un langage pour communiquer, Eyrolles, 1993 (3^eédition)
 -
- **domaines d'utilisation :**
 - télécommunications
 - avionique
 - systèmes d'armes ...

Objectifs de SADT

• Objectifs de la méthode :

- spécification fonctionnelle de systèmes complexes
- permettre les échanges aisés avec l'utilisateur
- favoriser le travail en équipe (communication)
- couplage avec une méthode de conception (MACH)

Il s'agit de poser un problème et non de le résoudre

Spécification : le quoi ?

Conception : le comment ?

• éviter certains problèmes inhérents à la phase de spécification :

- omissions
- contradictions
- redondances
- manque de clarté
- insuffisance de communicabilité
- "flous" des analystes
- ...

Avantages de la méthode

- permettre **l'analyse** et la **compréhension** de **systèmes complexes**
- être un **outil de communication** pour :
 - l'équipe (les analystes) : répartir le travail et coordonner les efforts
 - le client (expression des besoins)
 - la hiérarchie (direction et suivi du projet) : maîtrise de la réalisation et du suivi
- permettre des **spécifications** :
 - de **qualité**
 - **précises**
 - **complètes**

Attention :

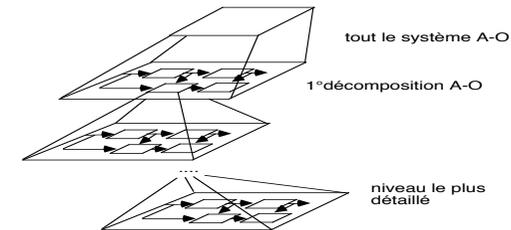
- ne dispense pas l'activité créatrice
- ne remplace pas l'analyste
- ne propose pas de formalisme permettant des vérifications sémantiques

Principes de base

- **approche descendante, modulaire, hiérarchique, structurée :**
un système complexe est décomposé en éléments (fonctions) figurant l'ensemble du modèle
- **approche du problème par un modèle de celui-ci :**
un modèle est constitué de **diagrammes** et de **textes**
- **distinction entre modèle du problème & modèle de la solution**
- **plusieurs vues et modèles possibles :**
 - modèle des activités
 - modèle des données
- **langage graphique associé :**
 - fonctions, relations,
 - structure hiérarchique
- **favoriser le travail en équipe : l'équipe SADT**
 - modularité
 - structure
- **obligation d'une forme écrite des choix :**
 - cycle auteur-lecteur, revue, archivage

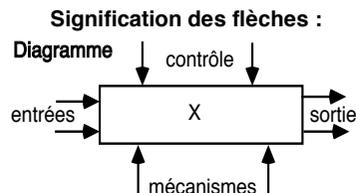
Concepts utilisés

- un modèle SADT représente :
 - les **données** (objets du domaine)
 - les **activités** (opérations)
- **dualité activités - données :**
 - **actigramme** (privilégie les activités)
 - **datagramme** (privilégie les données)
- **hiérarchie de diagrammes : diagrammes Père - Fils** (3 à 6 boîtes maxi par diagramme)
 - un diagramme fils ne doit contenir que des éléments appartenant au diagramme de son père
 - un diagramme fils représente toute la boîte père et rien que la boîte père



Représentation graphique

- **Règles générales :**
 - les **flèches** et les **boîtes** sont **identifiées**
 - la **position** des flèches sur une boîte est **significative**
- **Actigramme :**
 - boîte = fonction d'un niveau donnée (= verbe d'action)
 - flèche = contraintes interboîte
 - ne représente pas un flux de commande
 - ne porte pas de signification séquentielle
- **Datagramme :**
 - boîte = classe de données (= nom)
 - flèche = activités agissant sur cette classe

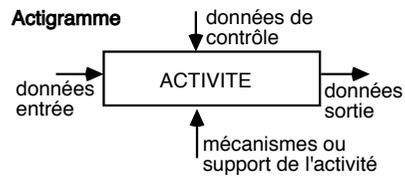


Actigrammes et Datagrammes

- **Actigrammes**
 - **entrées** : données transformées par l'activité en sortie
 - **sorties** : données créées par l'activité
 - **contrôles** :
 - données dont la présence contraint l'activité
 - données non modifiées par l'activité
 - **mécanismes** : processeur qui effectue l'activité (personne, machine ...) au moyen de :
 - **Règles sur les actigrammes :**
 - une boîte possède au moins une donnée de contrôle déclenchant l'activité et au moins une donnée de sortie
 - si une donnée apparaît comme contrôle et entrée on la place de préférence en contrôle
 - un mécanisme peut être défini par un autre modèle
- **Datagrammes**
 - **entrées** : activités génératrices de données
 - **sorties** : activités utilisatrices de données
 - **contrôles** : activités dont le résultat influe la création ou l'utilisation des données
 - **mécanismes** : unités de stockage des données

Actigrammes et Datagrammes

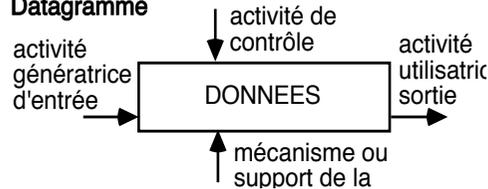
Signification des flèches :



- les données en entrées sont transformées en données de sortie par la fonction représentée par la boîte
- le contrôle agit sur la manière dont la transformation est faite
- le mécanisme indique ce qui supporte la fonction (individu, organisme, machine,...)
- remarque :

boîte = verbe d'action | flèches = nom

Datagramme



- activité (ventes, stockage, affichage, ...)
- activité d'entrée : modification sur les données
- activité de sortie : celle qui va les utiliser
- activité de contrôle : influe sur la manière dont une donnée est utilisée ou créée
- le mécanisme exprime le dispositif de mémorisation de données
- remarque :

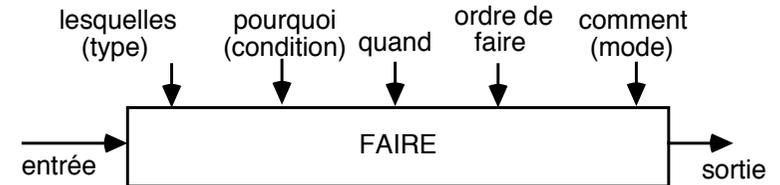
boîte = nom | flèches = verbe d'action

Actigrammes et Datagrammes : le contrôle

- concerne les boîtes d'activités

- un contrôle :

- n'est pas modifié par l'activité,
- mais il la **déclenche** ou la **contraint**

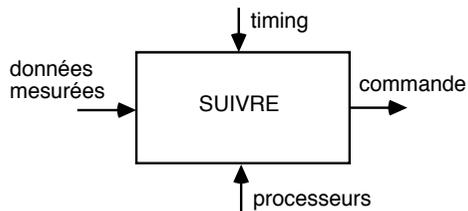


- une boîte d'activité doit avoir **au moins un contrôle** qui déclenche l'activité

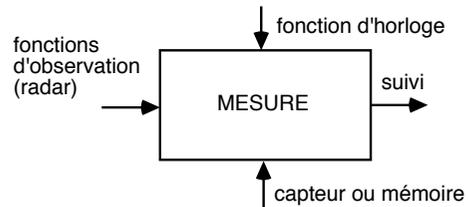
- si une **donnée** est à la fois un **contrôle** et une **entrée** on la représente comme un **contrôle**

- la **séparation contrôle / entrée** apparaîtra au niveau inférieur

Actigrammes et Datagrammes : exemples

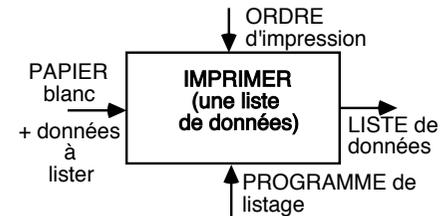


Actigramme

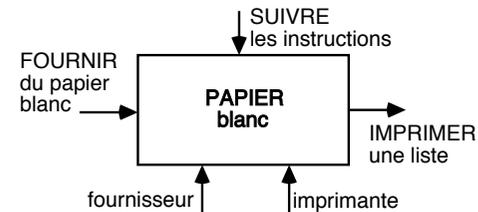


Datagramme

Actigrammes et Datagrammes : exemples



Actigramme



Datagramme

- les **activités** sont libellés par des **VERBES** commentés si besoin

- les **données** sont libellés par des **NOMS** qualifiés si besoin

Règles sur les diagrammes (actigrammes et datagrammes) :

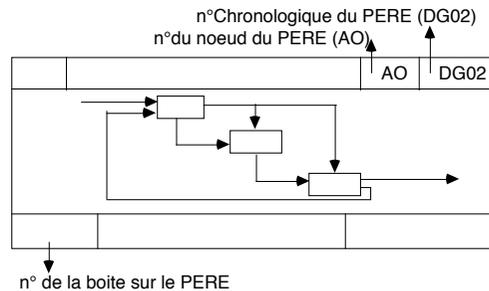
• règles de hiérarchie :

- les flèche externes d'une boîte **père** doivent se retrouver en flèches externes du diagramme **fils**

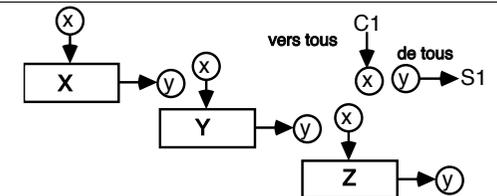
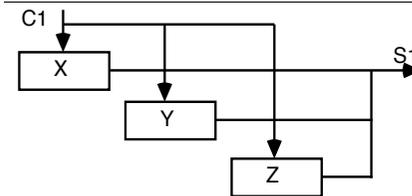
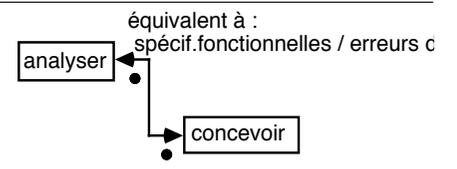
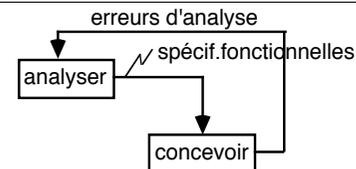
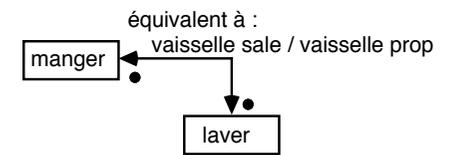
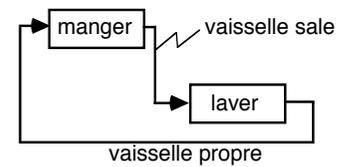
- le **nombre** de "boite" d'un diagramme est **limité de 3 à 6**

• règle de construction : on doit se laisser guider par l'enchaînement des données et non par la chronologie des opérations

• informations générées par le FILS pour le PÈRE :



Conventions simplificatrices



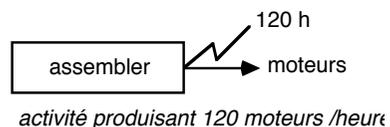
Remarques sur le formalisme SADT

Boîtes: Labels de propriété

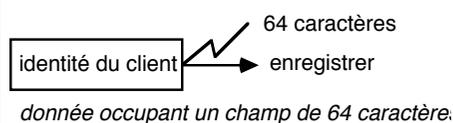
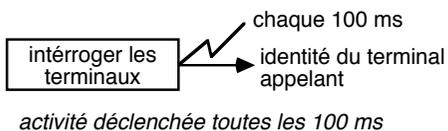
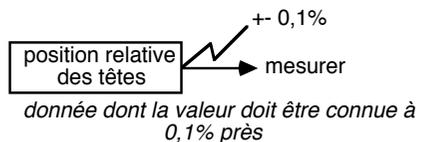
- les boîtes, les flèches, les notes ou les commentaires en disent beaucoup mais sont parfois éloignés de l'activité ou de la donnée qu'ils renseignent.
- on a souvent besoin d'une information courte, souvent numérique, associée à une activité ou à une donnée ; les labels de propriété expriment cette information

Exemples :

Actigrammes



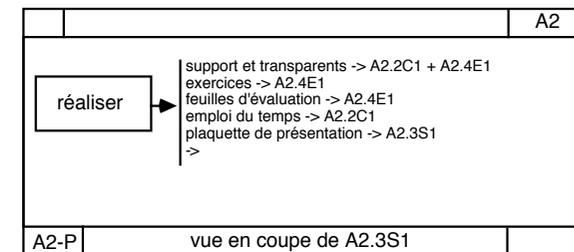
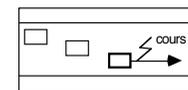
Datagrammes



Remarques sur le formalisme SADT

Les diagrammes pour explication seulement (P.E.S)

- ne font pas réellement partie du modèle
- clarifie ou illustre un aspect particulier du sujet
- représente copie simplifiée du diagramme, avec uniquement les flèches les plus importantes

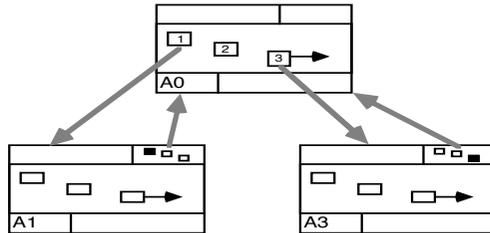


le PES A2-P explique comment le "cable" A2.3S1 "Cours" se décompose et quelles fonctions utilisent ses parties

Remarques sur le formalisme SADT

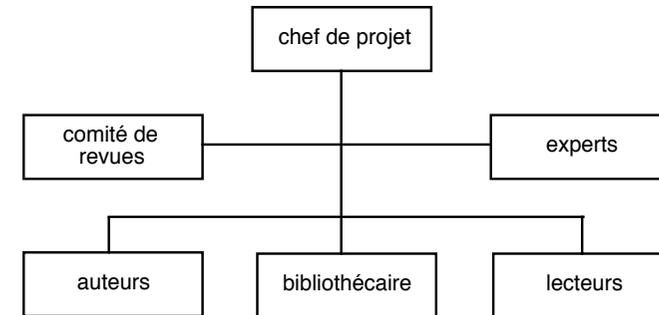
Listes hiérarchiques et numérotation des diagrammes

- le système de référence utilisé par SADT est basé sur l'**arbre hiérarchique** que constituent les diagrammes (actigrammes A et datagrammes D)
- hiérarchie exprimée en affectant à chaque actigramme et datagramme un **Numéro de Noeud**
- **par convention** : le système global et son environnement par un diagramme formé d'une seule boîte identifiable par le numéro de noeud **A - 0** (lire A moins 0).
- chacune des boîtes décomposée en un diagramme portant le **numéro de noeud** A1, ... An.
- par la suite, le numéro de noeud de chaque diagramme s'obtiendra en accolant au numéro de noeud du diagramme père celui de la boîte dont il fournit la décomposition



L'Equipe SADT

- le modèle SADT :
 - est produit pour être **communiquer** mais également pour être diffusé
 - n'a de sens que s'il est mis en jeu par l'**équipe SADT (cycle Auteur-Lecteur)**
- en créant cette **circulation de l'écrit**, on espère faire prendre conscience, avant la diffusion des diagrammes, des nécessités d'une prise en compte du **destinataire**, en faisant apparaître par exemple, les ressources du **questionnement** :



L'Equipe SADT

- **Auteurs** : analystes étudiant les besoins et les contraintes fonctionnelles du produit avant de les proposer sous forme de modèles SADT
- **Commentateurs (Lecteurs)** : analystes commentant et critiquant par écrit le travail des auteurs.

Les auteurs sont systématiquement lecteurs les uns des autres.

- **Experts** : spécialistes fournissant aux auteurs au cours d'interviews, des informations sur les facilités, les contraintes, les besoins du produit.
- **Comité Technique** : équipe technique de haut niveau, dont le rôle est de critiquer l'analyse à chaque étape importante de la décomposition. Ce comité résout les problèmes techniques ouлет recommande une décision à la gestion du projet.
- **Bibliothécaire** : a la responsabilité de la tenue à jour du fichier central, des pièces du produit, de la bonne circulation des documents de lecture et de leur archivage.
- **Chef de projet** : a la responsabilité technique du projet.

Cycle auteur-lecteur

- **l'auteur** des diagrammes soumet à un **lecteur**
 - les **diagrammes**
 - les **textes** explicatifs annexes
 - un **glossaire** des termes employés (dictionnaire de données)
- le **lecteur** :
 - **vérifie** la **syntaxe**
 - **vérifie** la **hiérarchie**
 - **analyse** la **modélisation** proposée
 - **émet** des **critiques** sur cette modélisation (commentaires écrits)
- **l'auteur** porte à son tour par écrit ses réactions aux remarques et aux suggestions faites par le lecteur: si désaccord, l'auteur et le lecteur en discuteront (résultats de la discussion par écrit)

Cycle auteur-lecteur

- un tel cycle de critique et d'approbation se déroule selon 2 axes :
 - celui de la hiérarchie des **diagrammes** constituant le modèle, et
 - celui de l'**ensemble des personnes impliquées** et cela jusqu'au **consentement final**.
- cette procédure documentée permet de savoir **pourquoi** des décisions particulières ont été prises et ce qui les a influencées.
- SADT conduit à la **création** et à la **mise à jour permanente d'un modèle**, évite d'avoir en fin de projet une phase de **documentation** trop importante.

Évolutions de SADT

- **Actigrammes** :
 - prendre en compte les aspects de synchronisation (réseau de PETRI)
- **Datagrammes** :
 - liaison avec des modèles existant type entite-relation (MERISE)
- **qualimétrie** dans les systèmes de spécification
- **outils de mise en oeuvre** :
 - ASA (société VERILOG)
 - ...
- **SA-RT**
- **IDEF0**
- ...