



# La conception

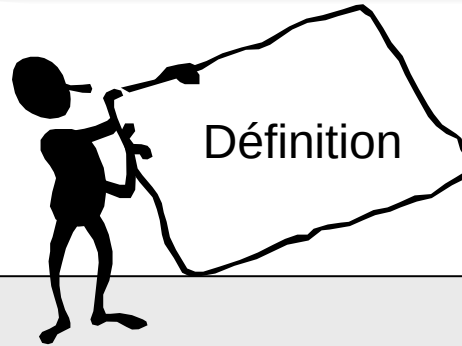
# La conception

La conception propose **une solution** au problème spécifié lors de la spécification :

- architecture de l'application (architecture logicielle et architecture physique)
- description détaillée des **modules**, des interfaces utilisateurs, des données et des fonctions

Elle donne lieu à un **document de conception**

# Définition de la conception



La conception donne une description interne de la solution, la plus explicite possible selon 2 dimensions complémentaires :

- **structure** (données, fonctions, ressources matérielles)
- **comportement** (très souvent description temporelle mais pas exclusivement)

L'objectif est de produire, à partir des spécifications fonctionnelles et opératoires, une description complète du système à concevoir.

# Le modèle structurel des fonctions

- C'est une structure construite à l'aide de fonctions internes et de relations entre ces fonctions. Une fonction est une entité assurant une activité spécifique du système.
- La décomposition en fonctions correspond à un découpage "topologique" et non à un découpage temporel des activités. Modèle hiérarchique (avec les dimensions horizontale et verticale)
- Par raffinements successifs, une fonction complexe se décompose en une structure de fonctions moins complexes, jusqu'aux fonctions élémentaires (pouvant s'exprimer par une description purement séquentielle).
- Il existe 3 types de relations entre fonctions :
  - relation de synchronisation, matérialisée par un événement
  - relation par partage de variable (variable d'état)
  - relation de transfert d'information, par échange de messages

# Le modèle structurel des données

- Il doit décrire complètement la nature des données (catégorie : variable d'état ou message, signification, structure, type de chaque composant).
- Il doit être indépendant de l'implantation.
- Il est du type relationnel et hiérarchique.

# Le modèle comportemental

- Ce modèle décrit la contribution des fonctions pour leur environnement soit leur spécification. Une fonction est une transformation des entrées en sorties.
- Tous les types de modèles temporels séquentiels sont utilisables (diagrammes à états finis, grafcet, algorithme).
- On distingue les fonctions temporaires (avec entrée d'activation par événement ou message) des fonctions permanentes (les autres).
- Il faut exprimer le maximum de parallélisme possible.

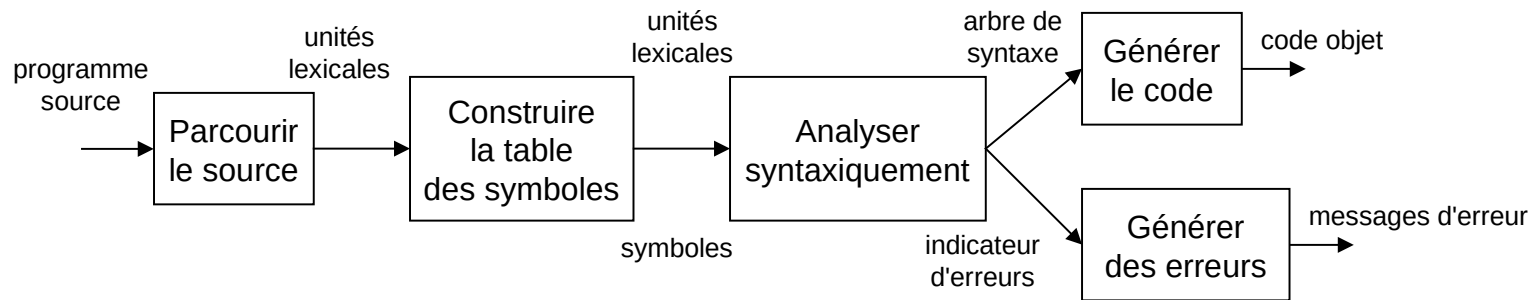
# Principes de conception

- La conception est basée sur les résultats de la spécification
- La conception est indépendante de la technologie  
Les constituants matériels ne doivent être considérés qu'au dernier moment.
- Les modèles doivent être de complexité minimale
  - Raffinement en favorisant la lisibilité et la compréhension.
  - La complexité de la solution "mise à plat" doit être minimale.
- Cohésion des constituants  
→ intégrité (les interfaces entre les composants doivent être définies), cohérence des modèles
- Compréhensibilité  
→ lisibilité, structuration, uniformité (style et format définis)
- Abstraction, raffinement, modularité, réutilisabilité, masquage de l'information

# Les 2 stratégies de conception

- La conception orientée fonction

Le système est décrit en termes de composants fonctionnels partageant une zone de données commune. L'état du système est centralisé. Ex d'un compilateur :



Dans l'approche de conception fonctionnelle, un module est un sous-système du système global. La relation de base est la relation de décomposition, qui constitue une hiérarchie. Les modules peuvent également utiliser des ressources que d'autres modules offrent. Cette relation d'utilisation est en général non hiérarchique (réseau).

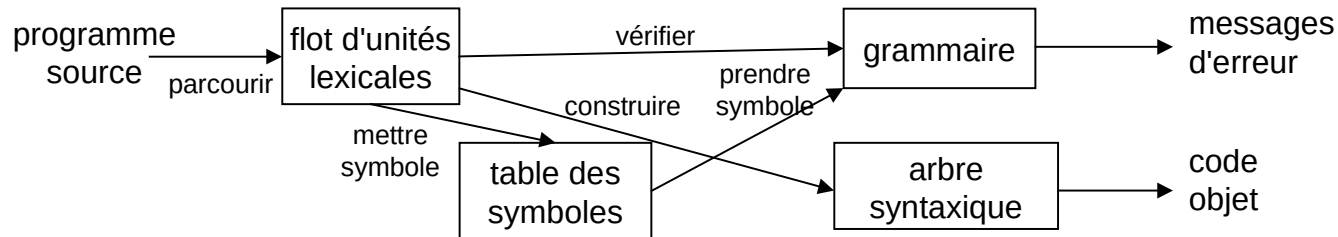
→ Cette architecture peut être remise en cause à chaque évolution des fonctionnalités. Elle est très mal adaptée aux évolutions des structures de données qui peuvent avoir un impact sur énormément de modules



# Les 2 stratégies de conception

- La conception orientée objet

Le système est vu comme un ensemble d'objets, chaque objet gérant l'information concernant son propre état. L'état du système est décentralisé. Ex d'un compilateur :



Dans l'approche orientée objets, les modules principaux correspondent aux objets du domaine de l'application. Les objets regroupent données et traitements. Ce sont des entités autonomes qui collaborent pour réaliser le système global. L'architecture est décrite au niveau des classes d'objets. La relation de base est la relation d'utilisation (par appel de service), qui constitue un réseau quelconque. On trouve aussi des relations de décomposition et d'héritage entre classes.

→ L'architecture n'est plus liée aux seuls traitements mais aux objets du domaine (données+traitements). Les évolutions sont plus locales, donc plus faciles.

# Les 2 stratégies de conception

---

En fait, les grands systèmes représentent des entités si complexes que l'on peut utiliser des stratégies différentes à une certaine étape de la conception de deux parties différentes du système.

# Modules et interfaces

Quelle que soit la stratégie de conception, on cherche à diviser un système en composants qui peuvent être **conçus indépendamment**. Pour cela, il faut que la nature de la relation d'utilisation, c'est-à-dire la nature des ressources et services qu'un module procure soit spécifiée explicitement et précisément : c'est **l'interface** du module. La manière dont ses services sont réalisés est son implantation qui est cachée. On sépare ainsi la vue abstraite d'un module nécessaire pour les clients du module (le contrat passé entre le module et ses clients), de la vue de son implantation. On peut programmer un module en ne connaissant que les interfaces des autres modules, c'est-à-dire en manipulant des abstractions. Un module bien conçu doit montrer le moins de choses à travers son interface. Le reste de l'information est caché (encapsulé) dans l'implantation. La partie cachée peut être modifiée sans aucun impact sur les modules clients à partir du moment où l'interface ne change pas.

# Autres aspects de la conception

En plus de la conception de l'application en modules, d'autres aspects doivent être considérés par les concepteurs qui nécessitent des méthodes et des techniques spécifiques :

- conception des interfaces utilisateurs
- conception des algorithmes
- conception des bases de données

# Démarche de conception orientée fonction

- Délimitation des entrées et sorties du système
- Recherche d'une première décomposition fonctionnelle
- Raffinement de chaque fonction (itération sur cette phase) par une approche basée sur les données plutôt que sur les fonctions
- Comportement de chaque fonction élémentaire
- Synthèse de la solution globale

Les données sont spécifiées au fur et à mesure de leur apparition pour chaque niveau de raffinement.

Le processus de conception peut mêler approches descendante et ascendante.

A chaque étape de raffinement, le concepteur doit réaliser les 3 phases suivantes : analyse des spécifications - construction d'une ou plusieurs solutions - vérification.

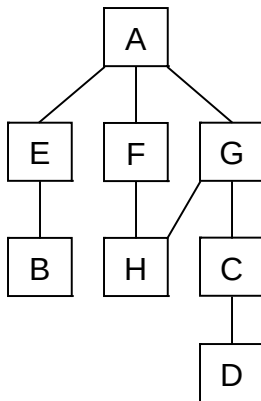
La documentation doit être structurée et complète, et décrire la solution selon une approche descendante.

# Démarche de conception orientée objet

- La notation **UML** (avec les démarches associées) est un standard utilisé pour la description d'une conception orientée objet.
- La notion de **patron** ("pattern") s'est imposée pour diffuser de bonnes pratiques sous forme de schémas de classes réutilisables pour répondre à des situations de conception qui apparaissent de manière répétitive dans beaucoup de projets. On distingue :
  - les patrons architecturaux (structure générale du système en sous-systèmes)
  - les patrons de conception ( structure d'un composant en sous-composants dans un contexte donné)
  - les idiomes (patrons de bas niveau dans un langage de programmation donné)
  - les cadres (sous-système prêt à être instancié avec des possibilités définies d'adaptation et d'extension)

# Architecture logicielle

- Elle donne la structure générale du logiciel
- Elle montre les relations hiérarchiques des modules/composants du logiciel
- Elle sert de cadre pour la conception détaillée des modules
- Si l'architecture est hiérarchique :

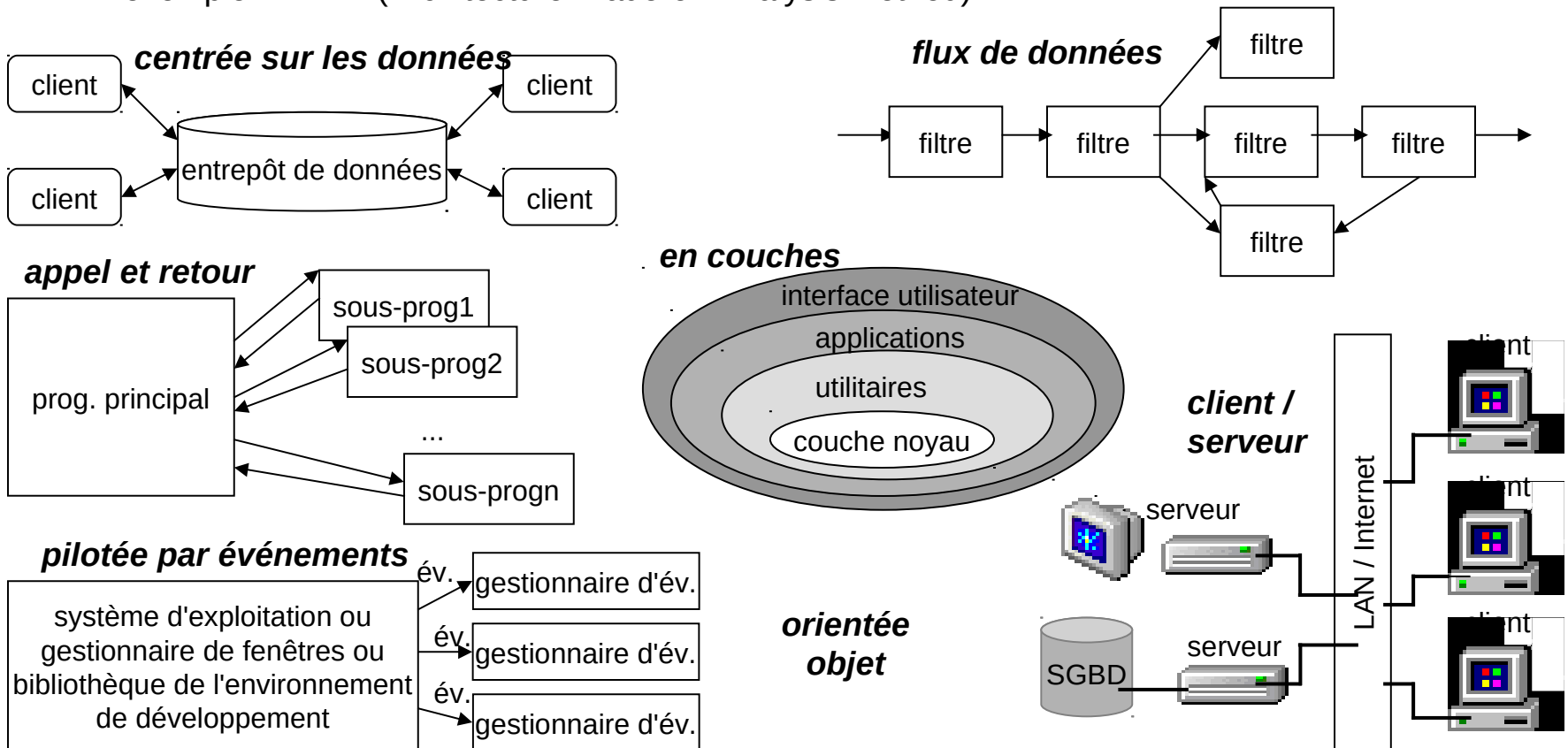


A,B, ..., H sont des modules.  
Les connexions représentent  
l'utilisation directe des  
modules par d'autres modules

- séparation horizontale : regrouper dans une branche distincte chaque fonction majeure ; le contrôle est alors distribué entre les branches horizontalement
- séparation verticale : le contrôle est distribué de haut en bas ; la prise de décision est effectuée par les modules des niveaux supérieurs, le traitement est réalisé par les modules des niveaux inférieurs

# Architectures logicielles

- Il existe une panoplie d'architectures logicielles déjà établies.
- Il existe des méthodes permettant de déterminer l'architecture qui convient, exemple : ATAM (Architecture Trade-off Analysis Method)





# Architecture physique

- Les systèmes sont de plus en plus distribués sur un réseau. Il est nécessaire de concevoir l'architecture physique de l'application. Les différents modules (liés à la présentation, à la logique applicative, aux services de base, etc.) sont répartis.
- Les architectures physiques de base sont :
  - 1 tiers : système centralisé
  - 2 tiers : système client/serveur où le serveur gère les données (BD et gestionnaire de BD) et les clients gèrent la présentation et la logique des traitements (clients lourds)
  - 3 tiers : système client/serveur tel que les clients légers ne s'occupent que de la présentation, la logique de l'application est gérée par un serveur d'application qui lui-même adresse des requêtes au serveur de données.
  - n tiers : système client/serveur tel que le serveur d'application fait appel aux services de plusieurs autres serveurs (d'applications ou de données).

# Le processus de conception

- La conception est un processus itératif → depuis la conception préliminaire jusqu'à la conception détaillée. La difficulté est de savoir quand arrêter le processus de raffinement.
- La conception préliminaire décrit la structure et le comportement d'un système en faisant abstraction des techniques et méthodes de réalisation. La conception détaillée affine, détaille, enrichit la description pour tenir compte des contraintes technologiques.
- En sortie de chaque activité de conception, on obtient une spécification de la manière dont le système doit être réalisé. L'ultime résultat est la spécification des algorithmes et des structures de données.
- En résultat de la conception, le document final doit décrire la solution et non pas la démarche suivie par les concepteurs.
- L'approche de conception est globalement descendante (décomposition fonctionnelle réursive du problème en sous-problèmes) → La dimension structurelle utilise un modèle hiérarchique.

# Conclusion sur la conception

*"La conception n'est pas une science, mais un art." (Bertrand Meyer).*

*"La conception est un processus créatif. La qualité du concepteur reste donc un facteur important d'influence sur la qualité de la solution." (Jean-Paul Calvez).*

La conception est facilitée par l'utilisation de règles (méthodologie), mais l'atout principal du concepteur reste l'expérience.