Diginamic Formation

« Dynamisez votre carrière digitale ! »

Modélisation et Architecture fonctionnelle

- Présenté par Christophe GERMAIN

Présentation

Digimamic Formation

« Dynamisez votre carrière digitale ! »

- Petit tour de table

Diginamic Formation

« Dynamisez votre carrière digitale ! »

- Généralité sur l'objet
- UML, vue d'ensemble.
- Modélisation des structures élémentaires statiques
- Des exercices
- Modélisation des structures élémentaires dynamiques
- Modélisation des structures élémentaires fonctionnelles
- UML et le processus Unifié
- Comprendre les besoins

- Modèle des cas d'utilisations
  - Etude de cas - exercice
- De l'initialisation à l'élaboration
  - Etude de cas - exercice
- Modèle du domaine
- Des besoins à la conception
- Conception objet et responsabilités
- De la conception au code

Diginamic Formation
« Dynamisez votre carrière digitale ! »

Un constat du marché

- Bonne nouvelle pour les professionnels du logiciel
  - Le logiciel fait tourner une économie moderne.
    - Permet au gouvernement de diriger
    - À la société de communiquer
  - Les systèmes informatisés que la société exige
    - Croissent en nombre, en taille, en complexité, en importance Le développement l'applications aujourd'hui

---

- Mauvaise nouvelle
  - **La croissance toute azimut conduit les développeurs au-delà des limites en matière de savoir-faire.**
  - **L'adaptation des anciens systèmes aux nouvelles technologies compte un grand nombre de difficultés techniques.**
  - **Le problème s'agrandie quant les sociétés exigent une plus grande productivité et une qualité accrues parallèlement à un dév. plus rapide**
  - **Enfin, l'offre en personnel de développeur qualifié ne suit pas la demande.**

Diginamic Formation

« Dynamisez votre carrière digitale ! »

Un constat des entreprises

- Une application peut comprendre des milliers de lignes de code
  - Un système implique alors la participation de plusieurs programmeurs
    - Il faut structurer les systèmes en composants pour faciliter la coopération des programmeurs

---

- Une application peut avoir une durée de vie de plusieurs décennies
  - Coût de remplacement ou adaptation trop élevé
  - les programmeurs sont séparés en temps
    - Connaissance du fonctionnement interne des composants est perdue
    - Nécessité de maintenir une documentation

Diginamic Formation

« Dynamisez votre carrière digitale ! »

Et la « Crise du Logiciel » (2004)

- Cette crise existe encore aujourd'hui
  - L'amélioration des méthodes de développement est dépassée par les besoins d 'applications de plus en plus complexes.
    - Peu de développement voient le jour
    - Start-up ferment
    - Les banques ne développent plus en interne
  - Toujours des difficultés de construire une application
  - Toujours des erreurs dans les spécifications ou dans le code
    - Besoins des clients sont mal compris ou changent pendant la programmation ou la vie du système
    - Erreur de programmation ou choix de structure de données (p.ex: bogue de l'an 2000, Ariane)

Diginamic Formation

« Dynamisez votre carrière digitale ! »

La crise : quelques symptômes

- Une mauvaise interprétation des demandes des utilisateurs finaux;
- une incapacité à tenir compte des changements du cahiers des charges ;
- des modules qui ne fonctionnent pas bien ensemble ;
- des programmes difficiles à maintenir et à faire évoluer ;
- la découverte tardive de défauts sérieux dans le projet ;

- la très faible qualité du logiciel ;
- des logiciels aux performances insuffisantes ;
- un manque de cohésion dans les équipes de développement, au point qu'il est impossible, *a posteriori*, qui à modifié quoi,quand, où et pourquoi ;
- un processus de construction et de livraison anarchique.

Diginamic Formation
« Dynamisez votre carrière digitale ! »

Les causes profondes

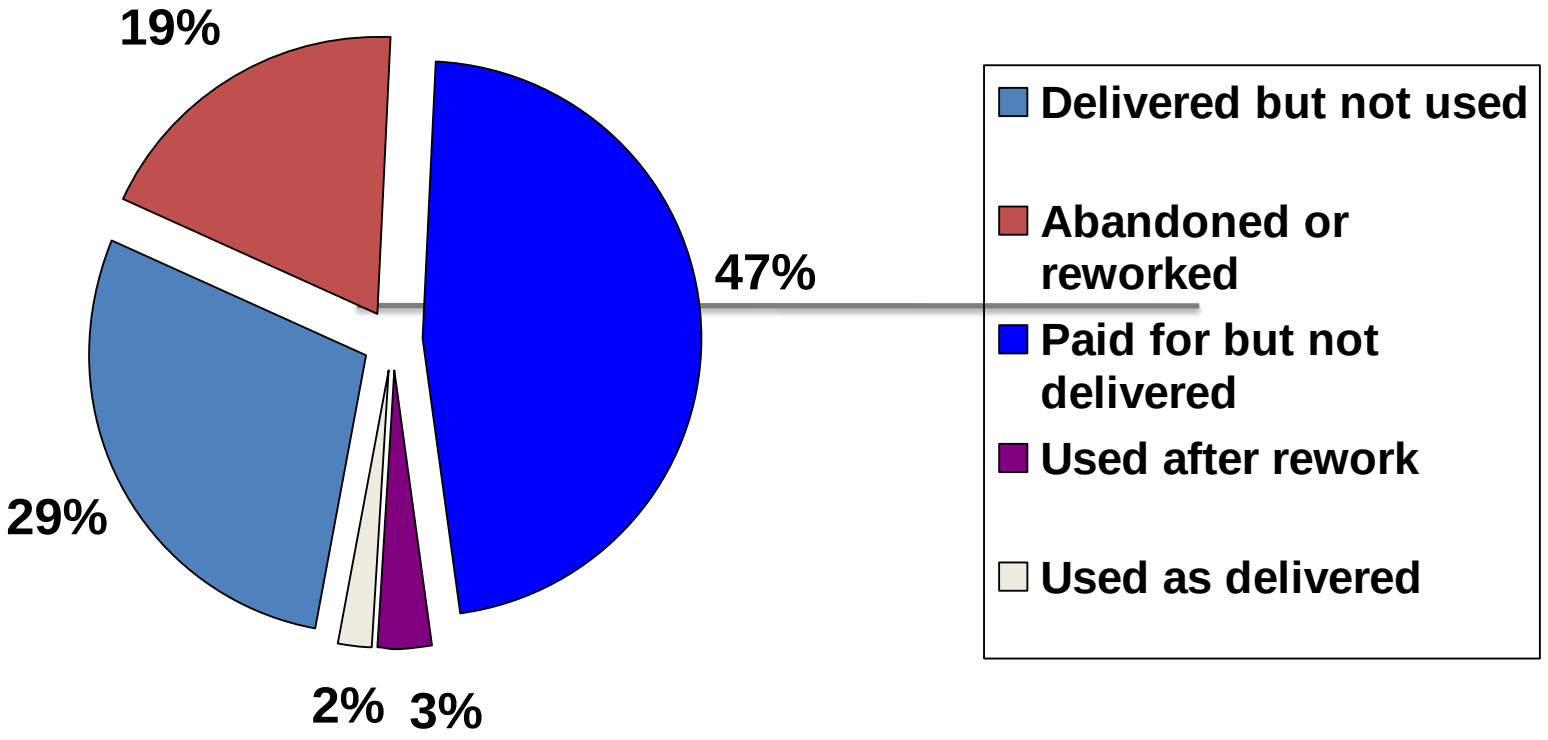
- une gestion du cahier des charges au coup par coup ;
- une communication ambiguë et imprécise ;
- des architectures fragile ;
- une complexité effroyable ;
- des incohérences entre le cahier des charges logiciel, les modèles de conception et l'implémentation ;

- un nombre insuffisant de tests ;
- une insouciance vis-à-vis des risques ;
- une absence de la propagation des changements ;
- une trop faible automatisation des tâches.

Diginamic Formation
« Dynamisez votre carrière digitale ! »

Coûts des logiciels

Source: US Gov. Accounting Report, (in B. Cox, OO Prog.)

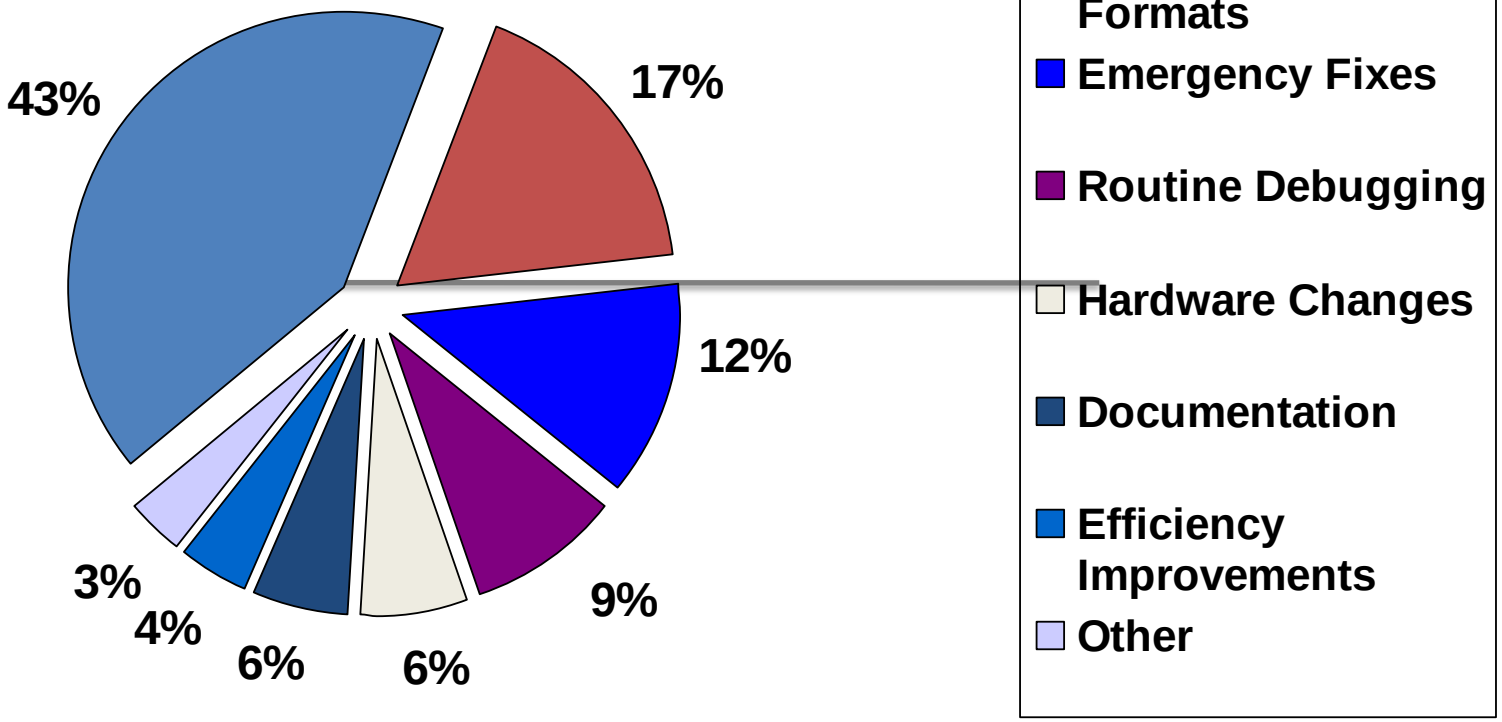


Diginamic Formation

« Dynamisez votre carrière digitale ! »

La maintenance en chiffres

Source:Lientz
(in B. Meyer, OO Soft. Constr.)



Diginamic Formation
« Dynamisez votre carrière digitale ! »

Retour en 1980

- Deux événements se sont produits dans les années 1980:
  - Les ordinateurs sont devenus de plus en plus puissants et économiques.
  - Les logiciels sont devenus de plus en plus gros (complexe) et coûteux

Il fallait trouver une solution

L'industrie ce tourne alors vers le paradigme Orienté Objet

Diginamic Formation
« Dynamisez votre carrière digitale ! »

L'orienté objet réduit la complexité

- La modélisation O.O. reproduit la façon dont les humains appréhendent le monde réel
  - Les objets sont des abstractions d'entités du monde réel
  - Ils sont regroupés en types (classes) ayant des caractéristiques semblables
  - Ils entretiennent des relations entre eux
  - Ils performent ou subissent des actions (ils ont un comportement)

Diginamic Formation
« Dynamisez votre carrière digitale ! »

Évolution des méthodes

- Modélisation centrée sur la sortie des résultats (1950s et 1960s)
- Modélisation centrée sur les processus (1970s)
- Modélisation centrée sur les entités (1980s)
- Modélisation orientée-objet (1990s)

Diginamic Formation
« Dynamisez votre carrière digitale ! »

Le + de la modélisation O. O.

- La modélisation centrée sur les processus est excellente pour représenter la dynamique d'un système, mais inefficace pour représenter les entités

- La modélisation centrée sur les entités est excellente pour représenter les propriétés statiques d'un système, mais inefficace pour en représenter les fonctionnalités

➔ La modélisation orientée objet combine les forces des deux approches précédentes; elle est excellente pour représenter les entités *et* le comportement

Diginamic Formation
« Dynamisez votre carrière digitale ! »

Modélisation orientée objet

- La modélisation O.O découle directement de la modélisation E.-R;

- Un objet est la même chose qu'une entité à l'exception qu'il possède un comportement, ce qui n'est pas représenté par le modèle E.-R.

➔ Cette caractéristique confère à la modélisation O.O. un avantage indéniable: celui de bien gérer les changements

Diginamic Formation
« Dynamisez votre carrière digitale ! »

Un exemple s'impose

- Problème:

  Un instructeur doit indiquer à ses étudiants comment se rendre à leur prochaine salle de cours

- **Solution selon la décomposition fonctionnelle:**
  1. **Obtenir une liste de tous les étudiants de la classe**
  2. **pour chaque personne:**
  3. **trouver la prochaine salle de cours**
  4. **trouver la localisation de cette salle**

  ---

  5. **identifier le chemin pour s'y rendre**
  6. **fournir l'information**

- **Solution selon la modélisation objet:**
  1. **L'instructeur affiche un plan montrant la localisation de l'actuelle salle et celles des autres salles de cour.**
  2. **Il avise les étudiants et leur dit d'utiliser cette information pour trouver leur salle de cours respective.**

Diginamic Formation

« Dynamisez votre carrière digitale ! »

Différence entre les 2 approches:

- Dans le premier cas,
  - l'instructeur est responsable de donner des directions explicites à chaque étudiant
- Dans le second cas,
  - il fournit des instructions générales à l'ensemble des étudiants et s'attend à ce que chacun trouve lui-même son chemin

---

➔ La différence réside dans le transfert de responsabilité

Diginamic Formation
« Dynamisez votre carrière digitale ! »

Conclusion

1. Les étudiants, et non l'instructeur, sont responsables d'eux-mêmes
2. L'instructeur peut parler à différents types d'étudiants (gradués ou non gradués) de la même façon
3. L'instructeur n'a pas besoin de connaître le chemin détaillé que doit suivre chaque étudiant

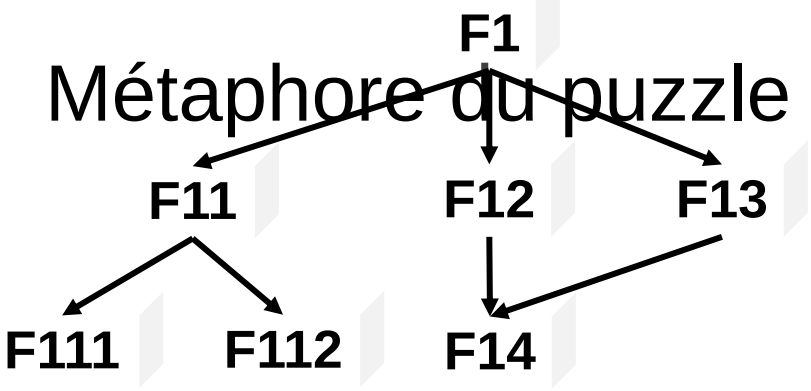
---

Fin de l'exemple

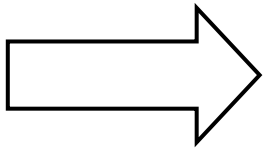
Diginamic Formation

« Dynamisez votre carrière digitale ! »

Métaphore du puzzle

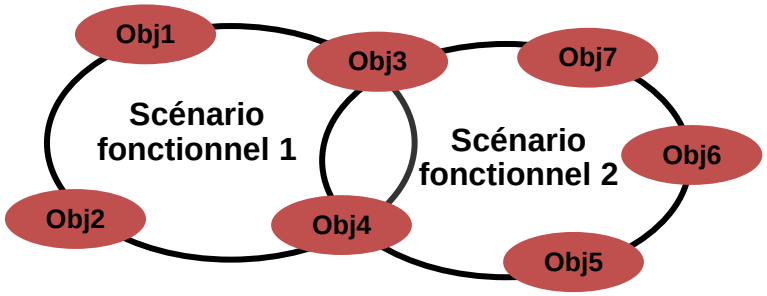


Approche fonctionnelle

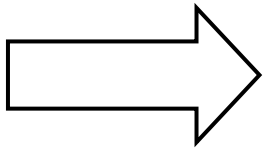


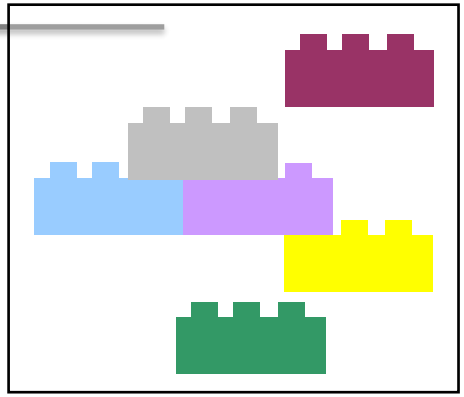
Projet

Un puzzle de constituants non réutilisable



Approche objet





Un ensemble de brique logicielle à composer

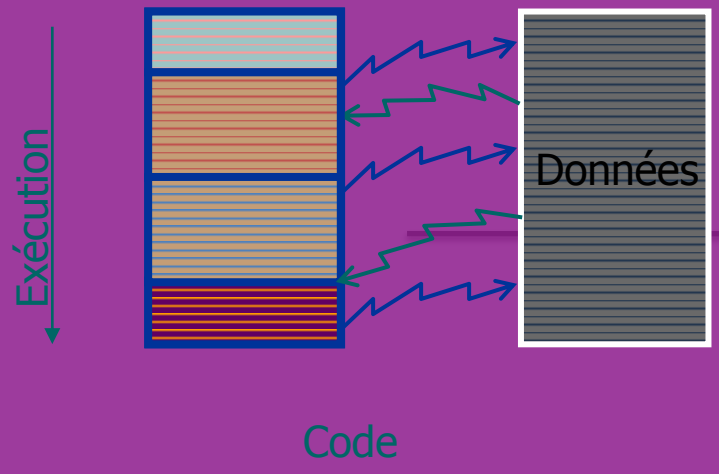
Diginamic Formation

« Dynamisez votre carrière digitale ! »

Autres comparaisons

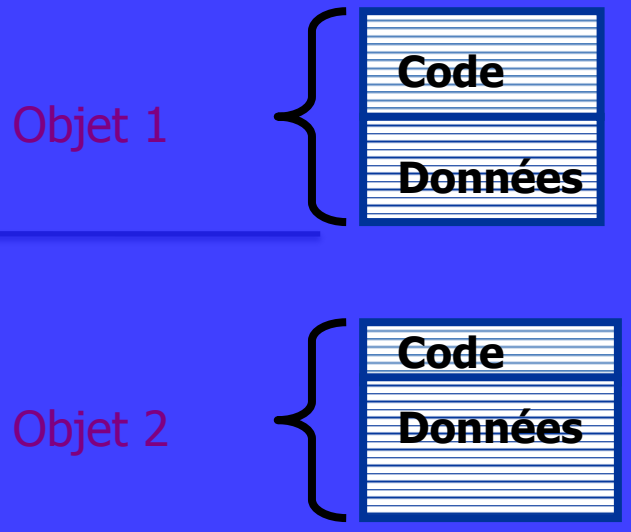
Programmation procédurale

= code et données sont distincts



N'importe quelle partie du code peut modifier n'importe quelle donnée si elle est accessible

objet = **Structure contenant des données + le code capable de les manipuler**



Les données ne doivent pas être accessible autrement qu'avec le code de l'objet dans lequel elles résident

Diginamic Formation
« Dynamisez votre carrière digitale ! »

Un concept central : l'objet

- Comme une entité, un objet a un nom, possède des attributs associés à des données et entretient des relations avec d'autres objets

- Il possède en plus un comportement exprimé par un programme qui permet d'accéder ou de modifier les données qui lui sont associées

- Un objet *encapsule* à la fois les données qui le concernent et les fonctions requises pour les modifier

Diginamic Formation
« Dynamisez votre carrière digitale ! »

Paradigmes de programmation

- De la programmation fonctionnelle à la programmation par objets
  - Programmation fonctionnelle
    - Style de programmation qui consiste à composer des fonctions qui prennent des arguments et retournent des valeurs
  - Programmation impérative
    - Style de programmation ayant un effet de bord sur les variables du programme
  - Programmation procédurale
    - Extraction d'une partie du programme utilisée plusieurs fois et appelée procédure
  - Programmation modulaire
    - Les procédures ayant des fonctionnalités communes sont groupées en *modules*
  - Programmation orienté-objet
    - Un programme est défini en terme d'*objets*

Diginamic Formation
« Dynamisez votre carrière digitale ! »

Questions autour du paradigme

- Qu'est-ce qu'un paradigme?
  - Un paradigme est un mode de pensée qui permet de structurer nos connaissances, notre apprentissage et notre compréhension (Brown, 2002)
- Qu'est-ce qu'un changement de paradigme?
  - Un changement de paradigme se produit lorsque nous adoptons une façon radicalement différente d'organiser en totalité ou en partie notre compréhension du monde
- Changements de paradigme dans l'informatique
  - Au cours des 30 dernières années, le domaine des systèmes d'information et celui du logiciel ont connu des changements de paradigme
  - Ces changements se sont manifestés à plusieurs niveaux:
    - Dans les méthodes de modélisation
    - Dans les systèmes de gestion de bases de données
    - Dans les langages de programmation

Diginamic Formation
« Dynamisez votre carrière digitale ! »

Le paradigme orienté-objet ?

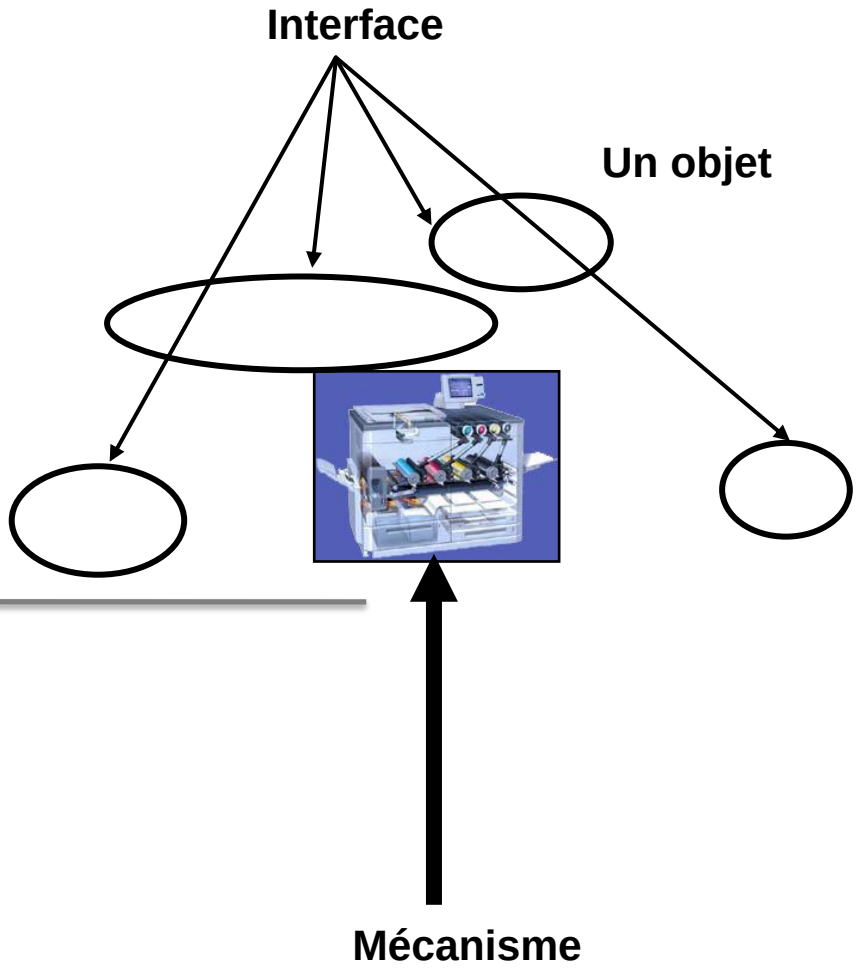
- Définition :
  - C’est une nouvelle façon de penser à des problèmes en utilisant des modèles organisés autour de concepts du monde réel
  - La composante principale est l’objet qui combine la structure des données et le comportement en une seule entité
  - Les objets sont des composantes qui peuvent être réutilisés dans des problèmes différents

Diginamic Formation

« Dynamisez votre carrière digitale ! »

Définition du paradigme objet

- Un objet est une entité capable d'exécuter un travail.
- Pour utiliser un objet, il faut en avoir un disponible.
- On utilise alors son interface qui est un ensemble de manettes et d'indicateurs permettant de configurer et d'utiliser l'objet; elle permet aussi de récupérer les résultats.
- En tant qu'utilisateur, la mécanique ne nous intéresse pas;
- elle est une boite noire que seul le créateur de l'objet peut comprendre, modifier, réparer.



Diginamic Formation
« Dynamisez votre carrière digitale ! »

Les 3 paradigmes objets de base

1. *L'Héritage*, au minimum "héritage de type", normalement "héritage de classe", permet de définir ce qu'un objet étend / modifie d'un autre. Cette modification concerne son interface uniquement (*Héritage de type*) ou son implémentation (Héritage de classe).

2. *L'Encapsulation*, qui permet de regroupement de plusieurs items dans un container (physique ou logique), et qui est très souvent lié au *masquage d'information*, afin de rendre visible uniquement les caractéristiques de conception stable d'un concept. (typiquement au travers d'accesseurs).

3. *Le polymorphisme*, qui permet de travailler avec plusieurs types d'objets comme s'ils n'en représentaient qu'un. On notera que le type de polymorphisme employé dans la programmation objet est avant tout de type *universel*, plus précisément un *polymorphisme d'inclusion*, donc basé sur l'héritage.

Diginamic Formation
« Dynamisez votre carrière digitale ! »

Des méthodes de modélisation

- L'apparition du paradigme objet à permis la naissance de plusieurs méthodes de modélisation
  - OMT, OOSE, Booch, Fusion, …
- Chacune de ces méthodes fournie une notation graphique et des règles pour élaborer les modèles
- Certaines méthodes sont outillées

Diginamic Formation

« Dynamisez votre carrière digitale ! »

L’importance de la modélisation

- Nous construisons des modèles pour les systèmes complexes parce que nous ne sommes pas en mesure d’appréhender de tels systèmes dans leur intégralité.

Diginamic Formation
« Dynamisez votre carrière digitale ! »

Pourquoi modéliser ?

- Parce que de la contrainte naît l'efficacité
  - *Modéliser les applications avant de programmer facilite les choses.*
    - *Anticipation du résultat du codage,*
    - *Facilite l'étude et la documente,*
    - *Est un outils majeur de la communication,*
    - *Facilite la traçabilité*

  - *Modéliser en utilisant des méthodes de conception prévient de douloureux errements*
    - *Il permet de décrire la cible d'une étape en cours.*
  - *Modéliser impose un commun accord sur l'architecture*
    - Permet le travail en équipe et facilite l'industrialisation.

Diginamic Formation
« Dynamisez votre carrière digitale ! »

Un bon modèle, c'est quoi?

- Un bon modèle est une simplification de la réalité
- Un bon modèle devrait
  - utiliser une notation standard
  - être compris des clients et utilisateurs
  - permettre aux ingénieurs logiciel de bien saisir le système
  - procurer une vue abstraite du système

- Les modèles sont utilisés pour:
  - faciliter la création de designs
  - permettre l'analyse et la révision des ces designs
  - constituer la base de la documentation du système

Diginamic Formation
« Dynamisez votre carrière digitale ! »

Les 4 principes de modélisation

1. Le *choix des modèles* à créer a une forte influence sur la manière d'aborder un problème et sur la nature de sa solution.
2. Tous les modèles peuvent avoir *différents niveaux de précision*.
3. Les meilleurs modèles *ne perdent pas le sens de la réalité*.
4. Parce qu'aucun modèle n'est suffisant à lui seul, il est préférable de *décomposer un système important en un ensemble de petits modèles presque indépendants.*

Diginamic Formation
« Dynamisez votre carrière digitale ! »

Et toujours la crise du logiciel

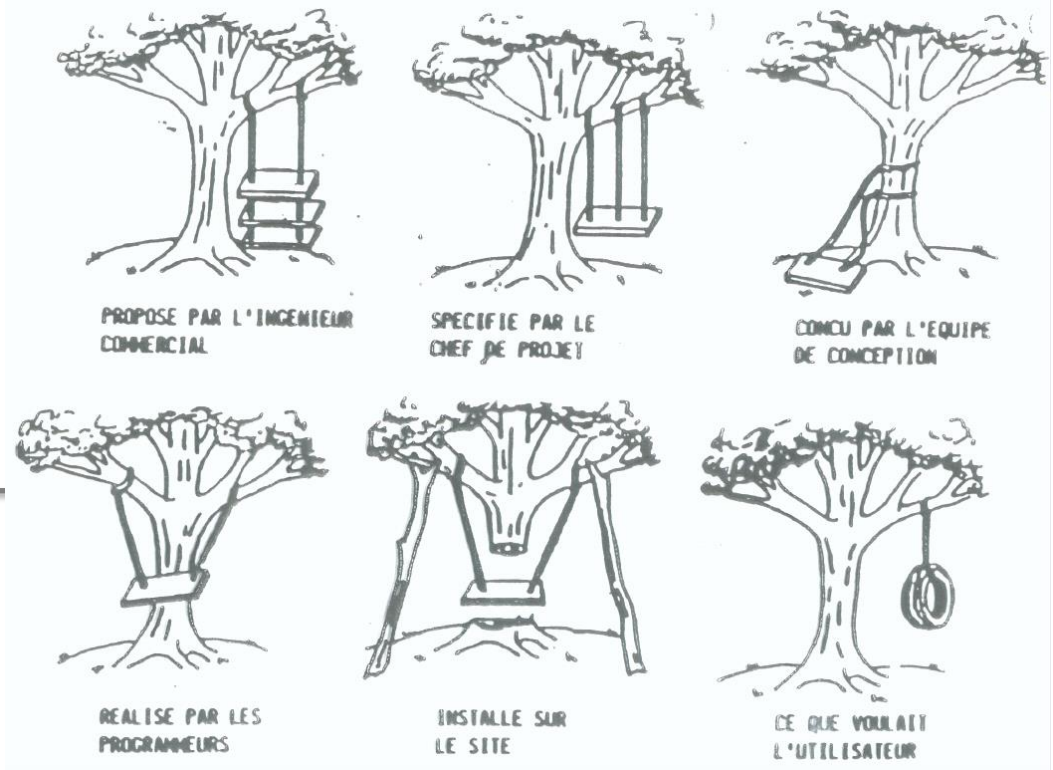
- Premier problème : la complexité
  - Selon Scientific American (Septembre 94), en moyenne,
    - les grands projets prennent 50% plus de temps que prévu,
    - les trois quarts des grands projets subissent des échecs en phase opérationnelle
    - un quart d'entre eux sont abandonnés.
  - Mais tous les grands projets souffrent des mêmes défauts.
    - Simplement les projets logiciels sont par essence très complexes, parce que les instructions de base de la machine sont très simples et qu'il faut en assembler énormément pour effectuer une tâche complexe.
      - Exemples,
        - logiciel d'un central téléphonique comporte environ 5. ^6 lignes de code (1998)
        - Windows 2000 en comporte environ 11.^6 .
- Le risque d'échec n'est pas le seul problème à mettre en avant, même si c'est le plus crucial.

Diginamic Formation

« Dynamisez votre carrière digitale ! »

Le problème des spécifications

- Le second problème concerne la spécification du système à implémenter.
  - le client ne sait pas ce qu'il veut ou ce dont il a besoin.
  - Ou pire croie savoir ce dont il a besoin,
  - Conséquences le client est devenu un acteur important dans les causes d'échecs.
  - Et plus les changements sont tardifs dans le cycle de développement, plus ils coûtent cher à réaliser.



Diginamic Formation

« Dynamisez votre carrière digitale ! »

Et pourtant, trop de méthodes

- Entre 89 et 94 : le nombre de méthodes orientées objet est passé de 10 à plus de 50
- Toutes les méthodes avaient pourtant d'énormes points communs (objets, méthode, paramètres, …)
- Au milieu des années 90, G. Booch, I. Jacobson et J. Rumbaugh ont chacun commencé à adopter les idées des autres. Les 3 auteurs ont souhaité créer un langage de modélisation unifié

Naissance d'UML en 1995

Unified Modeling Language

Diginamic Formation
« Dynamisez votre carrière digitale ! »

UML n'est pas une méthode

- Voici les travaux qui ont influencé le développement d'UML :
  - Booch, catégories et sous-systèmes,
  - Embley, classes singleton et objets composites,
  - Fusion, description des opérations et numérotation des messages,
  - Gamma et al., frameworks, patterns et notes,
  - Harel, automates (statecharts),
  - Jacobson, cas d'utilisation (usecases),
  - Meyer, préet postconditions,
  - Odell, classification dynamique, éclairage sur les événements,
  - OMT, associations,
  - ShlaerMellor, cycle de vie des objets,
  - WirfsBrock, responsabilités (CRC).

- UML se veut donc être un langage pour décrire des modèles :
  - générique,
  - expressif,
  - flexible (configurable et extensible),
  - disposant d'une syntaxe et d'une sémantique,
  - unificateur des expériences passées.

Diginamic Formation

« Dynamisez votre carrière digitale ! »

Résumons

Une crise du logicielle

- **Complexité ↗**
- **Durée de vie ↗**
- **Cycle de vie des projets ↘**
- **Coût ↗**

Langages objets disp. ↗
Equipement PC en entreprise ↗
Modèle Client/Serveur ↗
Arriver de COM et DCOM

Internet
Modèle N-tier
XML
Service Web

Naissance du paradigme Objet
1980 - 1987

Trop de méthodes
1987 - 1994

Naissance d'UML
1994 - 2xxx

Naissance d'UP
2000 – 2xxx

Lois de Moore

- **Coût ordinateur ↘**
- **Puissance ↗**

- **Nouvelle méthodes de conception, OMT, OOSE, E-R, …**
- **Outils Rad**
- **Notion de composant**

Web de 3 ième génération
Economie digitale

Diginamic Formation

« Dynamisez votre carrière digitale ! »

Points clés

- L'industrie du logiciel à façon est toujours en crise.
- Un changement de paradigme de monde fonctionnel vers l'orienté objet.
- Ce paradigme fondée sur l'objet.
  - une entité, avec un nom, des attributs associés à des données et entretient des relations avec d'autres objets
  - Il possède en plus un comportement exprimé par un programme qui permet d'accéder ou de modifier les données qui lui sont associées
  - Un objet encapsule les données et les fonctions, peut hériter d'un autre, et peut être polymorphe

---

- Un paradigme plus complexe nécessitant un modèle
- Un bon modèle est une simplification de la réalité
- La modélisation permet
  - Anticipation du résultat du codage,
  - Facilite l'étude et la documente,
  - Est un outils majeur de la communication,
  - Facilite la traçabilité.

- **Les modèles sont utilisés pour:**
  - **faciliter la création de designs.**
  - **permettre l'analyse et la révision des ces designs.**
  - **constituer la base de la documentation du système.**
- **Les 4 principes de la modélisation**
  - **Le choix du modèle.**
  - **Le niveaux de précisions.**
  - **Ne pert pas le sens des réalités.**
  - **Doit permettre la décomposition en modèle plus petit.**

- **Le problème des spécifications**
  - **Le client**
  - **La gestion du changement**

Diginamic Formation

« Dynamisez votre carrière digitale ! »

Plus d'information …

- OMG http://www.omg.org/
- UML http://www.uml.org/
- UML Rational https://www.ibm.com/docs/en/rsm/7.5.0?topic=tours-create-diagrams-using-rational-uml-modeling-tools