

Métamodélisation

Anne Etien

Maître de conférences

Université Lille 1

Anne.Etien@univ-lille1.fr

INTRODUCTION

Note

- Ces transparents sont très largement inspirés des cours de Eric Cariou, Pierre Alain Muller, Jean-Marc Jezequel, Mireille Blay-Fornarino, Benoît Combemale, Stéphane Ducasse...

2

Definition

Software maintenance is the modification of a software product after delivery to correct faults, to improve performance or other attributes.

ISO/IEC 14764:2006 Software Engineering —
Software Life Cycle Processes — Maintenance

3

4

Definition

Legacy software: A system which continues to be used because of the cost of replacing or redesigning it and often despite its poor competitiveness and compatibility with modern equivalents. The implication is that the system is large, monolithic and difficult to modify.

mondofacto.com/facts/dictionary

5

Disponibilité sur le long terme

- Cycle de vie de l'Airbus A300
 - Le programme a commencé en 1972, la production a été arrêtée en 2007
 - 2007-1972 = **35 années**
 - Le support durera jusqu'en 2050
 - 2050-1972 = **78 années !!**



6

Logiciel patrimonial

```
public class VerboseParser extends VerboseParser {
    public static final String DEFAULT_CODE_VERSION = "Java
    ...
    * Action: The version of Java expected by the parser
    protected String codeVersion = null;
    ...
    * Action: whether to generate local information (local to a type)
    protected boolean localInfo = true;
    ...
    * The arguments that were passed to the parser
    * Need to initialize the source file name
    private Collection<String> arguments;
    private Collection<String> arguments;
    ...
    * Java parser, provided by JDT
    private ASTParser fParser = null;
    public VerboseParser() {
        super();
    }
    ...
    protected SourceLanguage getLanguage() {
        return this.getSourceLanguage();
    }
    public void setArguments(Collection<String> args) {
    ...
}
```

1 page \approx 60 lignes de code (LOC)

Recto verso = 120 LOC

7

Logiciel patrimonial

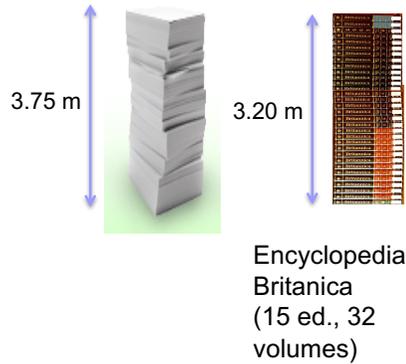


10 feuilles = 1200 LOC
(1.2 KLOC)

8

Logiciel patrimonial

Windows NT 3.1 (1993)
4 to 5 MLOC

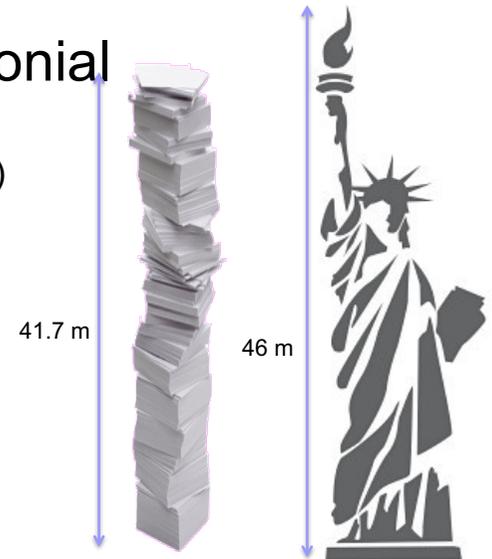


9

Logiciel patrimonial

Windows NT 3.1 (1993)
4 to 5 MLOC

Windows server 2003
50 MLOC



10

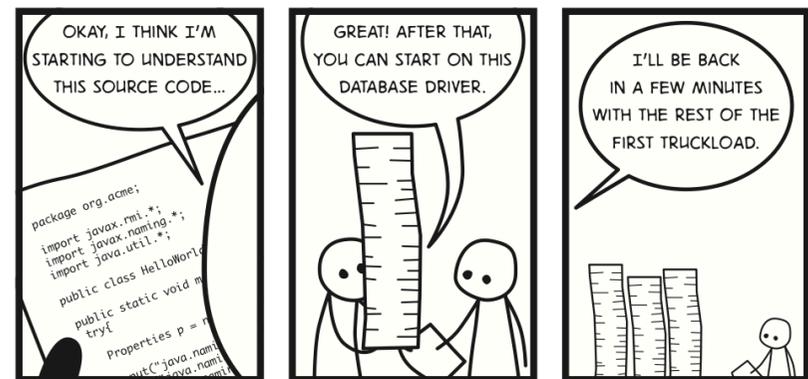
Logiciel patrimonial

- Linux kernel 3.6
→ 16 MLOC
- MacOS X 10.4
→ 86 MLOC
- Debian 5.0
→ 324 MLOC



11

Complexité des logiciels



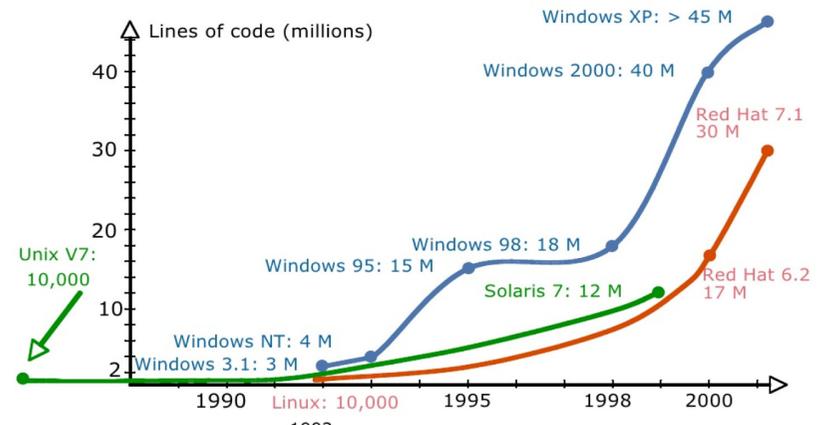
12

Complexité des logiciels

1 000 000 lignes de code
* 2 = 2 000 000 secondes
/ 3600 = 560 heures
/ 8 = 70 jours
/ 20 = 3,5 mois

13

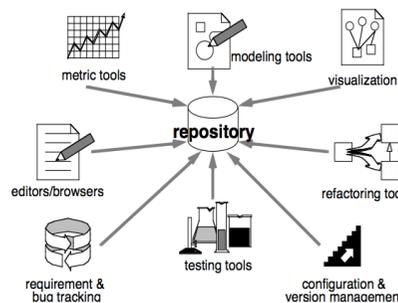
Complexité des logiciels



14

Problème d'intégration d'outils

- Réingénierie vs. forward ingénierie
 - Les outils de forward ingénierie sont choisis délibérément
 - Les outils de réingénierie doivent intégrer ce qui existe déjà.
- L'intégration d'outils est plus difficile en réingénierie
- Les outils doivent travailler ensemble :
- Données partagées ⇒ entrepôt de données
- Activités synchronisées ⇒ API
- Différents vendeurs ⇒ interopérabilité des standards



15

Solution alternative

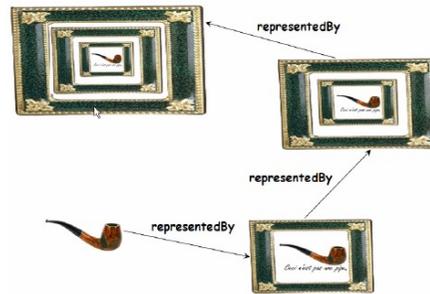
Une autre solution qui permette :

- Prendre du recul
- Conforme à la réalité
- D'aller plus vite
- Dans le but de :
 - Comprendre le logiciel
 - Identifier les problèmes
 - Proposer des solutions / évolutions

16

Les choses et leurs représentation

- Les choses
 - Réelles, virtuelles
 - Rares, chères, fragiles, dangereuses, inaccessibles, lointaines, trop nombreuses...
- Les concepts pour penser les choses
 - Plus facile, moins cher, moins dangereux



Le tableau de Magritte
La trahison des images revisité

17

Besoin de modèle

- Abstraction
- Réutilisation
- Meilleur compréhension
- Rapidité de développement
- Modularité
- Interopérabilité
- Indépendance vis-à-vis des outils

18

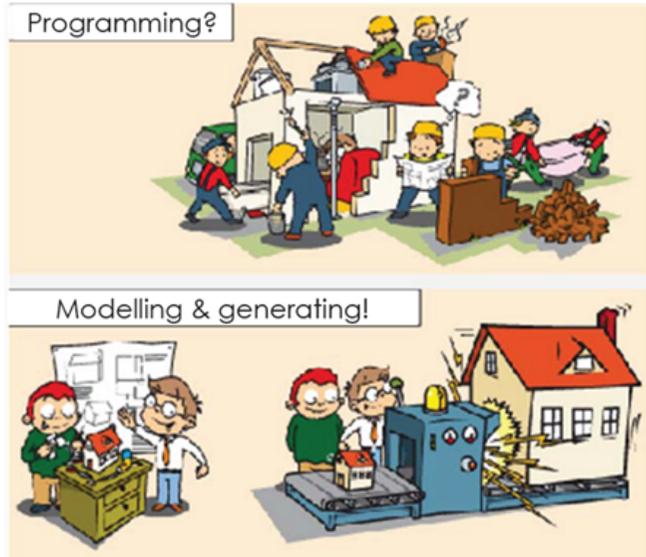
Besoin de modèle

- Un modèle est une description, une spécification partielle d'un système
 - Abstraction de ce qui est intéressant pour un contexte et dans un but donné
 - Vue subjective et simplifiée d'un système
- But d'un modèle
 - Faciliter la compréhension d'un système
 - Simuler le fonctionnement d'un système
- Exemples
 - Modèle économique
 - Modèle démographique
 - Modèle météorologique

19

Le développement comme la rétro ingénierie utilisent des modèles.

20



21

Principes de l'IDM

- IDM : Ingénierie Dirigée par les Modèles
- Séparation des préoccupations
 - 2 principales préoccupations
 - Métier : le cœur de l'application, sa logique
 - Plate-forme de mise en œuvre
 - Mais plusieurs autres préoccupations possibles
 - Sécurité
 - Interface utilisateur
 - Qualité de service
 - ...
- Chaque préoccupation est modélisée par un ... modèle
- Intégration des préoccupations
 - Par transformation/fusion/tissage de modèles
 - Conception orientée aspect

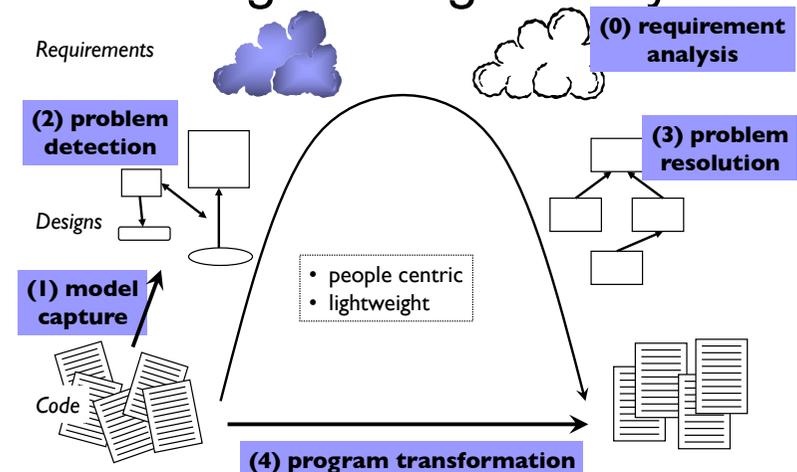
22

Remarques

- La modélisation n'est pas une discipline récente en génie logiciel
- Les processus de développement logiciel non plus
 - RUP, Merise ...
- C'est l'usage de ces modèles qui change
- Le but de l'IDM est
 - De passer d'une vision plutôt **contemplative** des modèles
 - A but de documentation, spécification, communication
 - A une vision réellement **productive**

23

The Reengineering Life-Cycle



© S. Demeyer, S. Ducasse, O. Nierstrasz

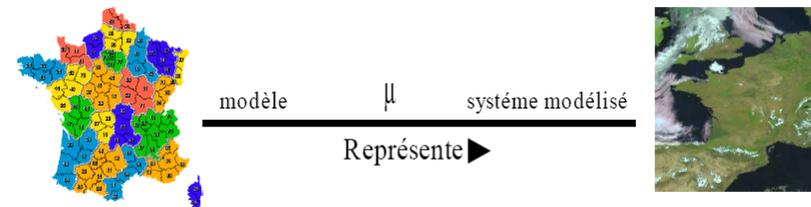
24

MODÈLE

25

Modèle

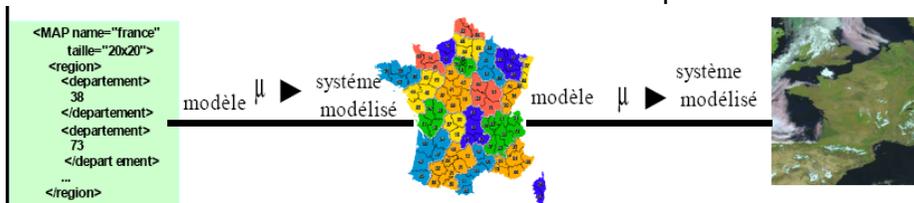
- Différence entre spécification et description
 - Spécification d'un système à *construire*
 - Description d'un système *existant*
- Relation entre un système et un modèle
 - ReprésentationDe (notée μ)



26

Modèle

- Un modèle représente un système modélisé
 - De manière générale, pas que dans un contexte de génie logiciel ou d'informatique
 - Un modèle peut aussi avoir le rôle de système modélisé dans une autre relation de représentation



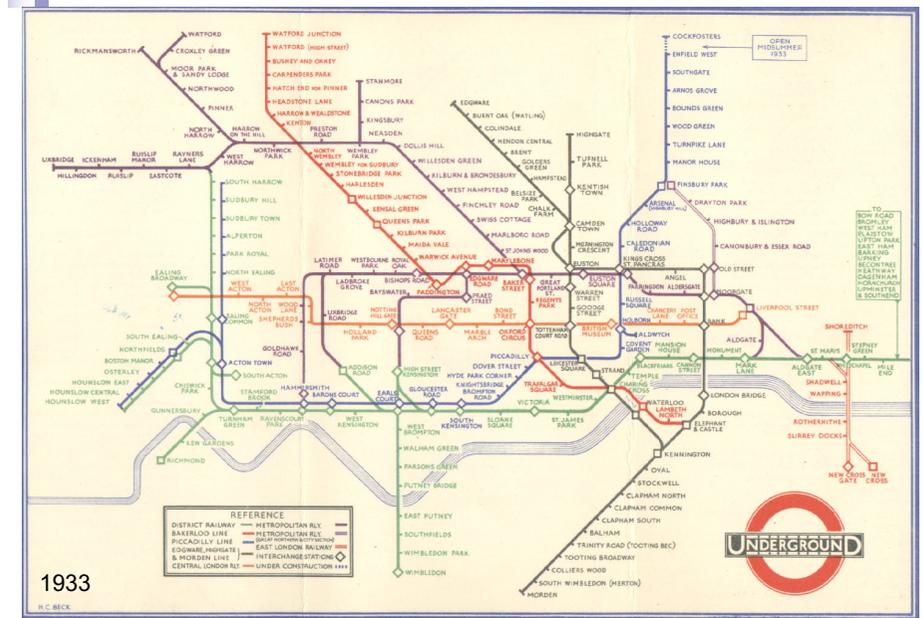
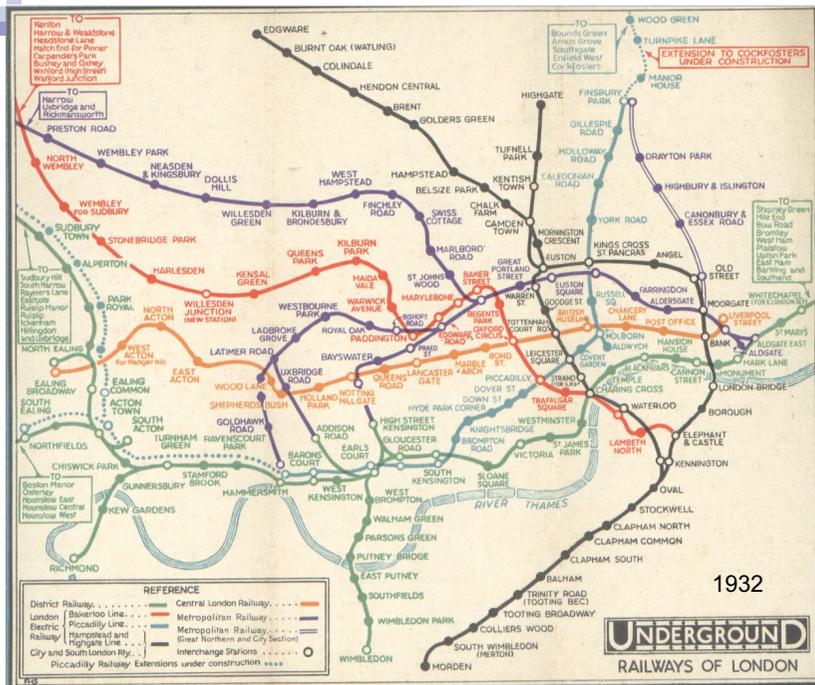
Modèle XML de la carte de la France administrative qui est un modèle de la France « réelle »

Modèles - contextes - vues



- Une France, mais plusieurs modèles de la France chacun représentant un point de vue du même système complexe

28



Modèles - contextes - vues



Un modèle sans légende n'est pas lisible

Modèle

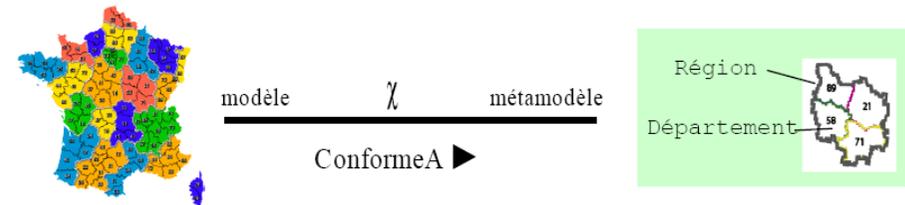
- Un modèle est écrit dans un langage qui peut être
 - Non ou peu formalisé, la langue naturelle
 - Le français, un dessin ...
 - Formel et bien défini, non ambigu
 - Syntaxe, grammaire, sémantique
 - On parle de méta-modèle pour ce type de langage de modèle
- Pour les modèles définis dans un langage bien précis
 - Relation de conformité
 - Un modèle est conforme à son méta-modèle
 - Relation EstConformeA (notée χ)

MÉTAMODÈLE

33

Métamodèle

- Un modèle est conforme à son métamodèle



34

Métamodèle

- Cette relation de conformité est essentielle
 - Base de l'IDM pour développer les outils capables de manipuler des modèles
 - Un métamodèle est une entité de première classe
- Mais pas nouvelle
 - Un texte écrit est conforme à une orthographe et une grammaire
 - Un programme Java est conforme à la syntaxe et la grammaire du langage Java
 - Un fichier XML est conforme à sa DTD
 - Une carte doit être conforme à une légende
 - Un modèle UML est conforme au métamodèle UML

35

Spécification de métamodèle

- But : définir un type de modèle avec tous ses types d'éléments et leurs contraintes
- Plusieurs approches possibles
 - Définir un métamodèle nouveau à partir de « rien », sans base de départ
 - Modifier un métamodèle existant : ajout, suppression, modification d'éléments et des contraintes sur leurs relations
 - Spécialiser un métamodèle existant en rajoutant des éléments et des contraintes (sans en enlever)
 - Correspond par exemple au mécanisme de profil UML

36

Exemple : bases de données relationnelles

Employé

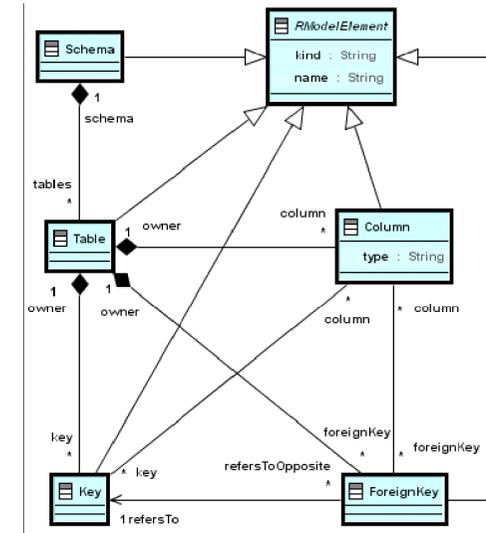
NoEmp	Nom	Année	NoDep
2067	Dupont	1965	06
0456	Martin	1981	03

Département

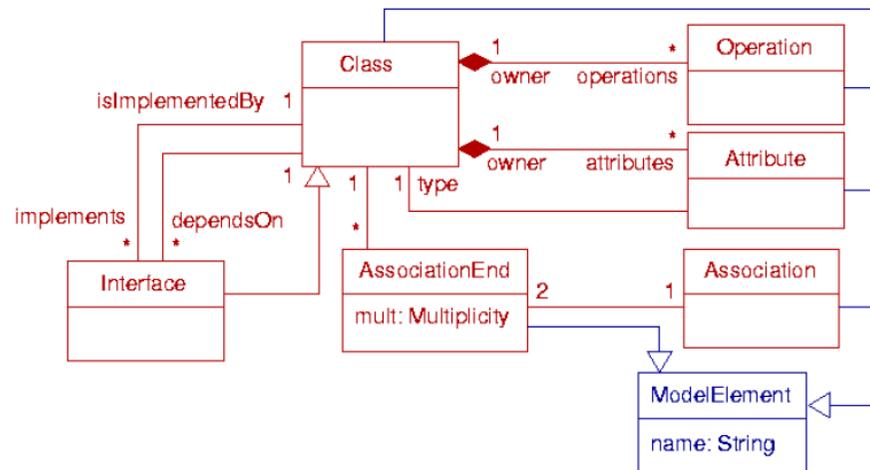
NoDep	Intitulé	Taille
03	Comptabilité	6
06	Informatique	10

Schéma de bases de données : niveau M1

Métamodèle de BDR



Méta-métamodèle



MODELES ET REVERSE ENGINEERING



Comprendre un logiciel

- Concepts manipulés
- Problèmes potentiels