

Analyse et Conception

7. Formalismes de Spécification



© Petko Valtchev

Université de Montréal

Octobre 2003

2

Méthode Z

Sommaire

- Méthodes formelles, classification
- La méthode Z, motivation
- Bases des notations
- Développer une spécification
- Preuves de correction
- Bilan sur les MF



© Petko Valtchev

Université de Montréal

Octobre 2003

3

- Les méthodes de spécification des besoins peuvent être catégorisées selon leur « degré de formalité »
 - **Méthodes informelles** (notations simples)
 - langage naturel,
 - diagrammes et tables simples
 - **Méthodes semi-formelles** (notations diagrammatiques complexes, vérifications possibles)
 - analyse structurée (**DFD, DE-A, DTE**)
 - analyse orientée-objet (**UML**)
 - **Méthodes formelles** (langages formelles de description avec une syntaxe et une sémantique clairement définies et donc vérifiables)
 - Basées sur des résultats mathématiquement fondés
 - **Z, VDM, B,**
 - **LOTOS**



Méthode Z Langages formels de spécif.

Langages formels de spécification catégorisés en *deux grandes catégories*:

- Notations basées sur la **théorie des modèles**: adaptées aux spécifications de systèmes **d'information**:
 - **Focus**: les états **cohérents** du système et les opérations **appropriées**?
 - *Ex. le langage Z, Vienna Development Method (VDM)*
- Notations basées sur des **algèbres de processus** : adaptées aux spécifications de systèmes **concurrentes**:
 - **Focus**: **correction** des protocoles de **communication** inter-processus
 - *Ex. Communicating Sequential Processes (CSP), CCS, LOTOS*



Méthode Z Aspects d'un langage formel

● Syntaxe

- définit le langage des **notations** particulières utilisées pour la spécification:
 - *alphabet*,
 - *symboles des opération, etc.*
- détermine les règles de **bonne formation**.

● Sémantique

- définit un « **univers d'objets** » ou « **domaine d'interprétation** » dans lequel les énoncés sur le système seront interprétés,
- établit le lien entre chaque énoncé et les parties de ce domaine,
- détermine les **énoncés valides**:
 - **de base** (axiomes),
 - **dérivés** (théorèmes).
- détermine les **règles d'inférence**: production d'un *énoncé valide* (théorème) à partir d'un ensemble d'*énoncés valides* (théorèmes ou axiomes).



Méthode Z

Sommaire

- Méthodes formelles, classification
- La méthode Z, motivation
- Bases des notations
- Développer une spécification
- Preuves de correction
- Bilan sur les MF



Méthode Z

Historique

- **Formalisme orienté-modèle**: exprime les propriétés souhaitables d'un système en s'appuyant sur des résultats provenant de *la théorie des ensembles*.
- **Origine**: développé par l'équipe de J.-R. Abrial à l'Université d'Oxford, dans les années 80.
- **Principe de modularité** : décomposition de la spécification en parties de taille raisonnable, les **schémas**, portant sur un seul aspect du système à la fois.
- **Schéma** (1ère approximation) = structure encapsulant la description:
 - des **états possibles** d'une partie ou du système entier : collections de **types** et des **variables** de ces types,
 - des **relations** qu'entretiennent les variables,
 - des effets des **actions** qui portent sur les variables.



© Petko Valtchev

Université de Montréal

Octobre 2003

8

Méthode Z

La Spécification en Z

« Une **spécification Z** se présente sous la forme d'une collection de **schémas** décrivant les éléments du système. »

- **Ex.** Schéma décrivant l'enregistrement d'une réservation de billet pour un spectacle.
- Aspects **Statiques**:
 - États du système
 - Propriétés invariantes pour un état et pour tous les états
- Aspects **Dynamiques**:
 - Opérations qui modifient l'état,
 - Opérations qui consultent l'état (sans le modifier),
 - Relations entre les entrées et les sorties.



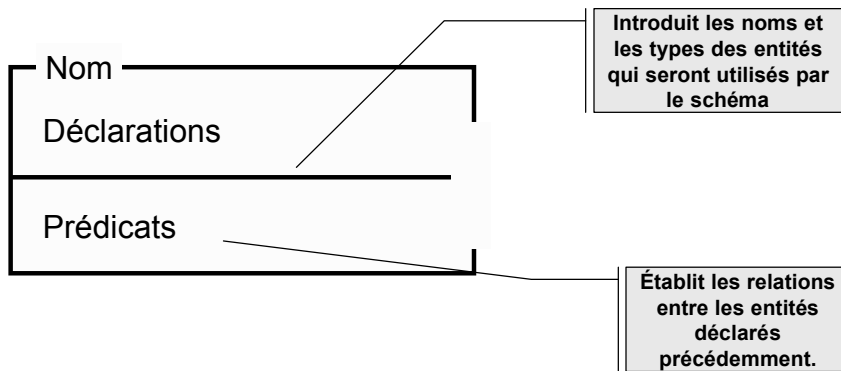
© Petko Valtchev

Université de Montréal

Octobre 2003

9

« Un **Schema** : unité syntaxique qui comporte trois parties principales:
nom, déclarations et prédicats. »



- Déclarations de **variable** de la forme:
 - < identificateur > : < type >
- Les prédicats précisent les **propriétés** des **variables** et les **relations** entre **variables**
- Un schéma est utilisé afin de décrire un **état** ou une **opération**
 - *Dans la description d'un état:*

Les **variables** déclarées représentent les **composantes de l'état**,
Les **prédicats** expriment les **propriétés invariantes** de l'état.
 - *Dans la description d'une opération:*

Les déclarations incluent les **composantes** de l'**état initial** et celles de l'**état final**, ainsi que tous les **entrées** et les **sorties** de l'opération.
Les **prédicats** expriment les **relations** entre les **entrées** et les **sorties**, et entre l'**état initial** et l'**état final**.



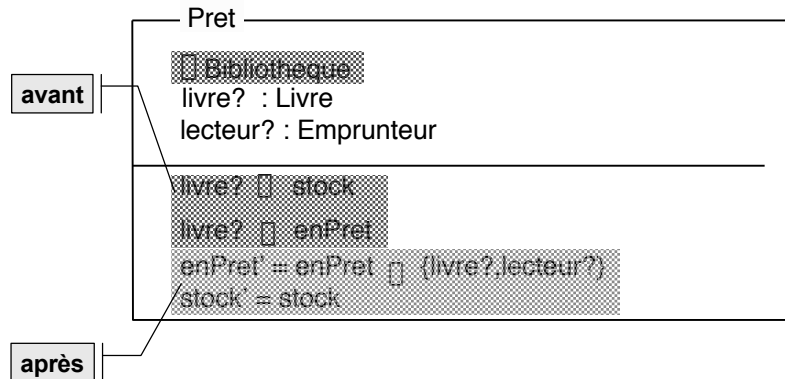
Méthode Z

Schéma de « Prêt »

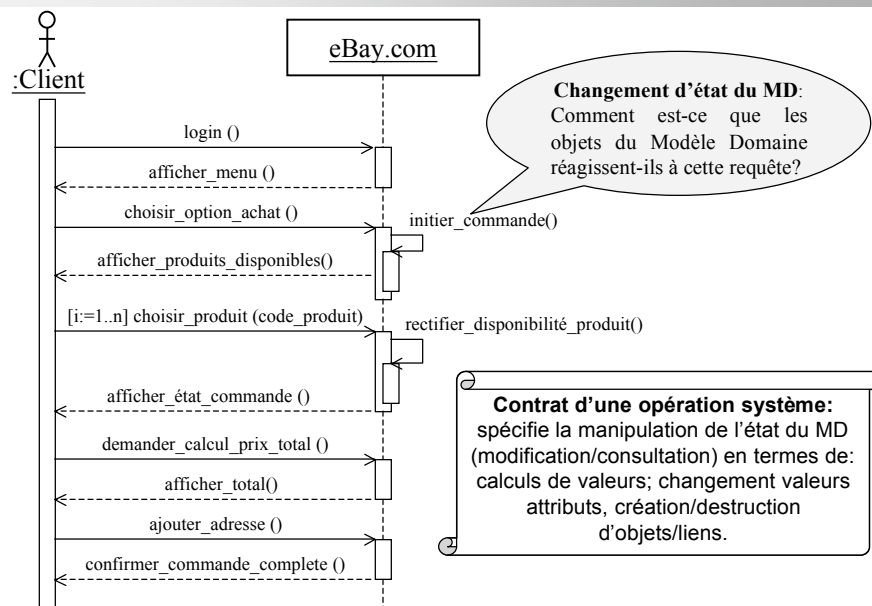
Contexte: Le SI de la *bibliothèque*,

Exemple: Le schéma de l'opération « prêt de livre »

1. décrit le changement de l'état du système suite à l'exécution de l'opération,
2. distingue l'état « avant » de l'état « après ».

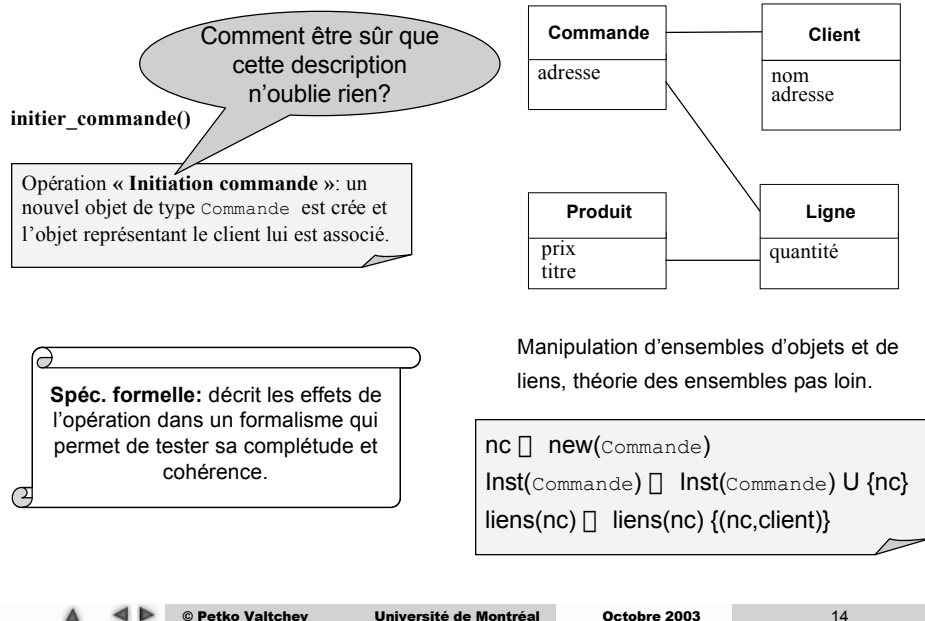


Méthode Z A quoi bon? - Les contrats OS



Méthode Z

Les Contrats Formalisés



Méthode Z

Sommaire

- Méthodes formelles, classification
- La méthode Z, motivation
- Bases des notations
- Développer une spécification
- Preuves de correction
- Bilan sur les MF

- Chaque **spécification** Z commence avec la déclaration de tous les types qui seront utilisés dans les déclarations des schémas de la spécification.

$[TYPE1, TYPE2, \dots]$

- Ex. $[Course, Student, \dots]$
- Déclarations de **variable** de la forme
 - *Identificateur* : *Type*
- Ex. *bestStudent* : *Student*



- **Type en Z** : interprété comme un **ensemble**. L'ensemble de toutes les valeurs possibles pour un type est appelé **carrier set** (ensemble-porteur).
- **Le schéma comme un type** :
 - Après son introduction, un **schéma** peut-être employé comme un type.
- **Types simples** :
 - **Prédéfinis**: Ex. entier (*N*), chaîne (*String*), etc.
 - **Définis par l'utilisateur**: Ex. Jour, Mois, etc.
- **Types composites** :
 - **Ensembles d'ensembles**: $\mathbb{P}$
 - **Produits Cartésiens**: $\square$
 - **Schémas**



« **Les prédicats** précisent
les **propriétés** des **variables** et les **relations** entre **variables**. »

Syntaxe :

Langage standard de la *théorie des ensembles*

- **Constantes:** entiers ($\dots, -1, 0, 1, 2, \dots$), ensemblistes ($\emptyset$), etc.
- **Opérations et prédicats sur les entiers:** $+$, $-$, $*$, div , mod , $>$, $<$, etc.
- **Opérations ensemblistes:** $=$, $\subseteq$, $\supset$, $\cap$, $\cup$, etc.
- **Connecteurs, quantificateurs:** $\neg$, $\wedge$, $\vee$, $\forall$, $\exists$, etc.



Syntaxe (suite):

Opérateurs sur fonctions et relations:

- **relation** ($X \subseteq Y$), **fonction totale** ($X \subseteq Y$) ou **partielle** ($X \twoheadrightarrow Y$),
- **domaine** ($\text{dom } R$), **co-domaine** ($\text{ran } R$), **identité** ($\text{id } R$),
- **composition** ($Q \circ R$), **image** ($x \mapsto y$),
- **restrictions:**
 - **domaine** ($Q \subseteq R$),
 - **co-domaine** ($R \subseteq Q$),
- **surcharge** ($Q \rightarrow R$),

