

# Spécification formelle

## MÉTHODES DE SPÉCIFICATION

École Supérieure de Technologie et d'informatique  
3<sup>ème</sup> année Ingénierie des Systèmes Intelligents

E. Menif Abassi

## Plan du module

2

- I. Introduction aux méthodes formelles
- II. Méthodes de spécification**
- III. Théorie des ensembles
- IV. Langage Z

# Plan du chapitre

3

1. Spécification
2. Intégration des méthodes formelles dans la spécification
3. Classification des méthodes formelles de spécification

# Plan du chapitre

4

1. Spécification
  - a. Qu'est-ce que la spécification?
  - b. Pourquoi spécifier?
  - c. Que faut-il spécifier?
  - d. Quand faut-il spécifier?
  - e. Méthodes de spécification
2. Intégration des méthodes formelles dans la spécification
3. Classification des méthodes formelles de spécification

# Plan du chapitre

5

1. Spécification
  - a. Qu'est-ce que la spécification?
  - b. Pourquoi spécifier?
  - c. Que faut-il spécifier?
  - d. Quand faut-il spécifier?
  - e. Méthodes de spécification
2. Intégration des méthodes formelles dans la spécification
3. Classification des méthodes formelles de spécification

E. Menif Abassi

Spécification formelle

ENICAR

# Spécification

6

- a. Qu'est-ce que la spécification?
  - ▣ Produit: Constitue un canal de communication entre les clients et les concepteurs du système. Il décrit les objets du système et leurs comportements individuels et collectifs
  - ▣ Processus:
    - Spécification ⇒ Ingénierie des exigences (Requirement Engineering)
    - Établit ce que le système doit faire (**QUOI**) et les contraintes sous lesquelles il doit opérer: une description de **haut niveau d'abstraction** des services que doit offrir le système
    - Processus itératif incrémental

E. Menif Abassi

Spécification formelle

ENICAR

# Spécification

7

## a. Qu'est-ce que la spécification?

### ▣ Activités du processus:

#### 1. Identification des parties prenantes

- une partie prenante est toute entité qui a un intérêt direct ou indirect au système
- Pour la MOA (Maîtrise d'ouvrage), il est primordial d'identifier l'ensemble des parties prenantes afin de recueillir les différents points de vue
- MOA: l'entité qui est porteuse du besoin, définissant l'objectif du projet, son calendrier et le budget consacré à ce projet. Le résultat attendu du projet est la réalisation d'un produit

#### 2. Définition du périmètre et du contexte du système

#### 3. Recueil ⇒ **Besoins énoncés**

- découvrir les futures exigences du système.

E. Menif Abassi

Spécification formelle

ENICAR

# Spécification

8

## a. Qu'est-ce que la spécification?

### ▣ Activités du processus (suite):

#### 4. Analyse des besoins ⇒ **Besoins réels**

- Reformuler des besoins énoncés sous forme de phrases simples, pour les clarifier et les unifier
- S'assurer que les besoins sont bien définis et bien formulés
- S'assurer d'utiliser un glossaire approprié pour référencer les termes et leur définition pour palier à la différence de culture et de vocabulaire entre une MOA et une MOE qui constitue une source importante d'incompréhension
- un glossaire doit être centralisé, accessible, obligatoire, validé par les parties prenantes et la responsabilité de sa maintenance doit être définie
- MOE (Maîtrise d'œuvre): l'entité chargée de la conception d'un produit, puis de la conduite opérationnelle de travaux

E. Menif Abassi

Spécification formelle

ENICAR

# Spécification

9

## a. Qu'est-ce que la spécification?

### ▣ Activités du processus (suite):

5. **Spécification des besoins** ⇒ **Exigences**
  - Rédiger, détailler, structurer, et documenter les exigences
  - Première version du cahier des charges
  - Langage naturel, semi-formel ou formel
6. **Validation**: analyse et validation des exigences avec l'ensemble des parties prenantes afin de repérer les insuffisances et non-alignement par rapport aux besoins réels
7. **Priorisation des exigences**
8. **Dérivation de nouvelles exigences**
9. **Allocation des exigences aux différents sous-systèmes**

# Spécification

10

## a. Qu'est-ce que la spécification?

### ▣ Activités du processus (suite):

10. **Traçabilité**
  - maintenir un lien bidirectionnel entre le besoin énoncé et la déclinaison de ce besoin en exigences, à travers des différents niveaux de référentiel
  - permet de justifier l'existence des exigences et des artefacts du projet, et de conserver l'historique de la déclinaison des exigences.
11. **Versionnement**
  - les exigences évoluent pour diverses raisons. Il est primordial de gérer efficacement les changements et les ajouts d'exigences au sein d'un référentiel
  - il est important de maîtriser l'incidence de ces changements sur le projet
12. **Vérification**
13. **Validation**

# Plan du chapitre

11

## 1. Spécification

- a. Qu'est-ce que la spécification?
- b. Pourquoi spécifier?
- c. Que faut-il spécifier?
- d. Quand faut-il spécifier?
- e. Méthodes de spécification

## 2. Intégration des méthodes formelles dans la spécification

## 3. Classification des méthodes formelles de spécification

E. Menif Abassi

Spécification formelle

ENICAR

# Spécification

12

## b. Pourquoi spécifier?

- Documentation précise et abstraite:
  - Canal de communication entre plusieurs acteurs du processus logiciel
  - Contrat:
    - Entre le client et le concepteur
    - Entre le concepteur et le développeur
- S'assurer de la complétude du système: Tenir compte de tous les cas y compris les exceptions et la gestion des erreurs
- Contrôler la complexité du système:
  - Décomposition: Décomposer le système et décrire chaque composant individuellement.
  - Composition: Décrire comment intégrer les différents composants

E. Menif Abassi

Spécification formelle

ENICAR

# Plan du chapitre

13

1. Spécification
  - a. Qu'est-ce que la spécification?
  - b. Pourquoi spécifier?
  - c. Que faut-il spécifier?
  - d. Quand faut-il spécifier?
  - e. Méthodes de spécification
2. Intégration des méthodes formelles dans la spécification
3. Classification des méthodes formelles de spécification

E. Menif Abassi

Spécification formelle

ENICAR

# Spécification

14

## c. Que faut-il spécifier?

- ▣ Besoins ( *User requirements* ) ⇒ Exigences ( *System requirements* ):
  - **Besoins de l'utilisateur (énoncés, réels)**: description en très haut niveau d'abstraction (langage naturel et diagrammes) de ce que le système doit fournir comme services et les contraintes sous lesquelles il doit opérer
  - **Exigences du système** appelé aussi **Spécification fonctionnelle** est une description (plus détaillée que les besoins de l'utilisateur) des fonctions, services et contraintes du système ⇒ ce que doit être implanté

E. Menif Abassi

Spécification formelle

ENICAR

# Spécification

15

## c. Que faut-il spécifier?

- Exemple d'un système de gestion d'une clinique psychiatrique:
  - **Besoins de l'utilisateur:** Générer des rapports mensuels qui montrent le coût des médicaments prescrits chaque mois pour chaque service d'une clinique psychiatrique
  - **Exigences du système:**
    - Dernier jour de chaque mois, générer la liste de tous les médicaments prescrits durant le mois, leurs coûts et les services prescripteurs
    - Le système doit générer automatiquement un rapport et l'imprimer après 17h30
    - Un rapport est généré pour chaque service et comprend la liste des médicaments: noms, nombre des prescriptions, doses et le coût total
    - L'accès aux rapports est restreint aux utilisateurs autorisés figurant sur la liste de contrôle d'accès

E. Menif Abassi

Spécification formelle

ENICAR

# Spécification

16

## c. Que faut-il spécifier?

- Les exigences du système sont classifiées en exigences fonctionnelles et non fonctionnelles:
  - **Exigences fonctionnelles:**
    - Services offerts par le système
    - Comment le système doit réagir à certaines entrées
    - Comportement du système pour certaines situations
    - Ce que le système ne devrait pas faire
  - ⇒ Le comportement observable du système: les interactions du système avec son environnement
    - Comportement **requis**
    - Comportement **permis**

E. Menif Abassi

Spécification formelle

ENICAR



# Spécification

17

## c. Que faut-il spécifier?

### ■ Exigences fonctionnelles:

#### ■ Exemples:

- Un utilisateur doit être capable de chercher la liste des rendez-vous dans tous les services de la clinique
- Chaque utilisateur du système doit s'authentifier à l'aide d'un code de 8 caractères

E. Menif Abassi

Spécification formelle

ENICAR

# Spécification

18

## c. Que faut-il spécifier?

### ■ Exigences non fonctionnels:

- Contraintes de fonctionnement du système: fiabilité, temps de réponse, sécurité, ...
- S'applique au système dans son intégralité plutôt qu'aux services du système individuellement
- Plus critiques et plus complexes à implanter, à vérifier et à mesurer
- Une difficulté supplémentaire des exigences non fonctionnelles est issue de leur immatérialité:
  - On ne peut pas les associer à un composant particulier du système
  - Une exigence non fonctionnelle peut générer plusieurs autres exigences (fonctionnelles ou non fonctionnelles)

E. Menif Abassi

Spécification formelle

ENICAR

# Plan du chapitre

19

## 1. Spécification

- Qu'est-ce que la spécification?
- Pourquoi spécifier?
- Que faut-il spécifier?
- Quand faut-il spécifier?
- Méthodes de spécification

## 2. Intégration des méthodes formelles dans la spécification

## 3. Classification des méthodes formelles de spécification

E. Menif Abassi

Spécification formelle

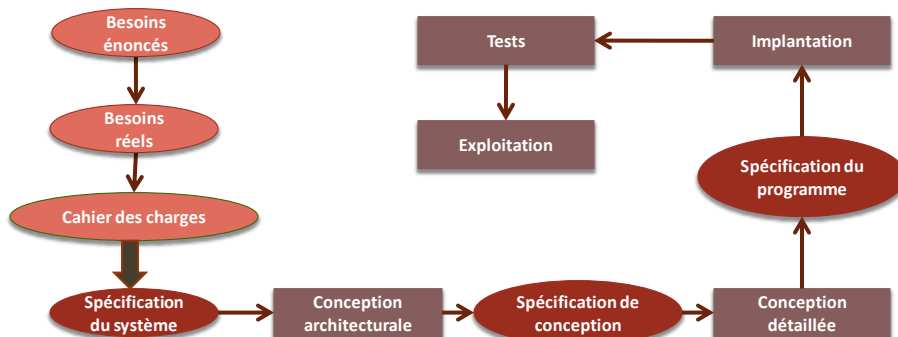
ENICAR

# Spécification

20

## c. Quand faut-il spécifier?

- Depuis le cahier des charges (**référentiel d'exigences**) jusqu'au produit livrable, les objets du systèmes passent par plusieurs niveaux d'abstractions: du plus haut au plus bas ⇒ La spécification est une activité en plusieurs étapes



E. Menif Abassi

Spécification formelle

ENICAR

# Spécification

21

## C. Quand faut-il spécifier?

- **Cahier des charges** (*Software Requirements Document SRD*) :
  - Contient les exigences réels
  - Utilisé par différents utilisateurs: utilisateurs finaux, chefs de projet du client, ingénieurs (système, tests, maintenance)
  - Le niveau des détails dépend:
    - Type du système
    - Processus de développement utilisé

E. Menif Abassi

Spécification formelle

ENICAR

# Spécification

22

## C. Quand faut-il spécifier?

- 1<sup>er</sup> niveau de spécification = **Spécification du système**: Élaboré à partir d'un cahier des charges et donne les exigences du système:
  - Description précise et sans ambiguïté du comportement observable du système
  - Contraintes du système
  - Indépendant de tout détail d'implantation
- 2<sup>ème</sup> niveau de spécification = **Spécification de conception** (Spécification des sous-systèmes)
  - Caractéristiques opérationnelles
  - Structure interne du système
  - Préserve les propriétés du niveau précédent

E. Menif Abassi

Spécification formelle

ENICAR

# Spécification

23

## c. Quand faut-il spécifier?

- ▣ 3<sup>ème</sup> niveau de spécification = **Spécification du programme** (Spécification des composants) met l'emphasis sur:
  - Plus de détails sur les données et les actions
  - Les composants de la conception et leurs interactions ⇒ Les interfaces des composants
  - Choix du langage d'implantation et de la configuration matérielle

# Plan du chapitre

24

1. Spécification
  - a. Qu'est-ce que la spécification?
  - b. Pourquoi spécifier?
  - c. Que faut-il spécifier?
  - d. Quand faut-il spécifier?
  - e. Méthodes de spécification
2. Intégration des méthodes formelles dans la spécification
3. Classification des méthodes formelles de spécification

# Spécification

25

## d. Méthodes de spécification

- Les besoins de l'utilisateur sont souvent exprimés en langage naturel agrémenté par des diagrammes et des tableaux.
  - Les exigences du système peuvent être exprimées à l'aide de différentes méthodes:
    - Langage naturel
    - Langage naturel structuré
    - Langage de conception
    - Notation graphique
    - **Méthodes formelles**
- } **Méthodes informelles**
- } **Méthodes semi-formelles**

E. Menif Abassi

Spécification formelle

ENICAR

# Spécification

26

## d. Méthodes de spécification

- **Langage naturel**: les exigences sont exprimées à l'aide de phrases en langage naturel
  - ✓ Expressif
  - ✓ Intuitif
  - ✓ Universel
  - ✗ Ambigu
  - ✗ Incohérent
  - ✗ Incomplet
  - ✗ Contradictoire

E. Menif Abassi

Spécification formelle

ENICAR

# Spécification

27

## d. Méthodes de spécification

- **Langage naturel**: Les défauts les plus courants (Bertrand Meyer)

Défaut	Description
<b>Bruit</b>	Élément du texte n'apportant d'information sur aucune des caractéristiques du problème
<b>Silence</b>	Caractéristique du problème à laquelle ne correspond pas d'éléments du texte
<b>Sur-spécification</b>	Élément du texte ne correspondant pas à une caractéristique du problème (mais a des caractéristiques d'une solution possible)
<b>Contradiction</b>	Élément du texte définissant de façon incompatible une même caractéristique du problème
<b>Ambiguïté</b>	Élément du texte permettant de comprendre une caractéristique du problème de deux façons ou plus
<b>Référence en avant</b>	Élément du texte utilisant des caractéristiques du problème définies plus loin dans le texte

E. Menif Abassi

Spécification formelle

ENICAR

# Spécification

28

## d. Méthodes de spécification

- **Langage naturel structuré**: les exigences sont exprimées en langage naturel en suivant un modèle préétabli. Chaque champ du modèle est dédié à un aspect d'une exigence
  - Utiliser des constructions de langage de programmation (alternatives et itération)
  - Mettre l'accent sur des concepts clés en utilisant des polices différentes
  - ✓ Uniformiser les documents de spécification
  - ✓ Organiser les spécifications plus efficacement
  - ✗ Ne résout pas les problèmes des langages naturels

E. Menif Abassi

Spécification formelle

ENICAR

# Spécification

29

## d. Méthodes de spécification

### □ Langage de conception (algorithmique):

- Semblable à un langage de programmation
- Utilisant des concepts plus abstraits
- Modèle opérationnel du système
- Rarement utilisé

### □ Notation graphique:

- Modèles graphiques annotés par du texte
- Exigences fonctionnelles
- Exemple: Cas d'utilisation et les diagrammes de séquence de UML

E. Menif Abassi

Spécification formelle

ENICAR

# Spécification

30

## d. Méthodes de spécification

### □ Méthodes formelles:

- Utilise des concepts mathématiques: théorie des ensembles, les machines à états finis, ...
  - ✓ Permet des vérifications automatiques
  - ✓ Les propriétés sont basées sur une argumentation stricte
  - ✓ Réduit les ambiguïtés et les contradictions
  - ✓ ...
- ✗ Difficile à utiliser
- ✗ Nécessite beaucoup d'effort et de moyen

E. Menif Abassi

Spécification formelle

ENICAR

# Plan du chapitre

31

1. Spécification
2. Intégration des méthodes formelles dans la spécification
  - a. Spécification formelle dans le processus logiciel
  - b. Activités de la spécification formelle
  - c. Qualités de la spécification formelle
3. Classification des méthodes formelles de spécification

E. Menif Abassi

Spécification formelle

ENICAR

## Intégration des méthodes formelles dans la spécification

32

- Le choix du formalisme dépend de:
  - ▣ **Type de l'application**: plus avantageux dans les projets complexes, critiques où la sûreté, sécurité, fiabilité, robustesse, ... sont des propriétés clés
  - ▣ **Phase du cycle**
  - ▣ **Degré de formalisme**:
    1. Informel et semi formel
    2. Formel sans outils de preuve
    3. Formel avec outils de preuve
  - ▣ **Taille et structure** : Les méthodes formelles sont utilisées efficacement dans des projets modérés ou uniquement aux composants critiques d'un grand système
  - ▣ **Fonctionnalités**: Plus les fonctionnalités sont critiques et plus le degré de formalisme nécessaire est élevé pour garantir une meilleure rigueur dans la vérification des propriétés

E. Menif Abassi

Spécification formelle

ENICAR



# Plan du chapitre

33

1. Spécification
2. Intégration des méthodes formelles dans la spécification
  - a. Spécification formelle dans le processus logiciel
  - b. Activités de la spécification formelle
  - c. Qualités de la spécification formelle
3. Classification des méthodes formelles de spécification

## Intégration des méthodes formelles dans la spécification

34

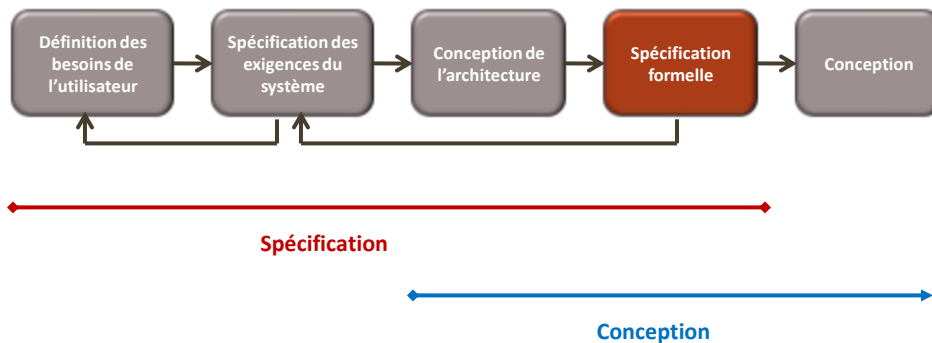
- a. **Spécification formelle dans le processus logiciel**
  - ▣ La spécification est une activité en plusieurs étapes  $\Rightarrow$  processus de spécification.
  - ▣ Spécification et conception sont conduits:
    - Séquentiel
    - Parallèle
  - ▣ La spécification formelle se situe après la spécification des exigences du système et avant la phase de conception

# Intégration des méthodes formelles dans la spécification

35

## a. Spécification formelle dans le processus logiciel

### ■ Processus séquentiel



E. Menif Abassi

Spécification formelle

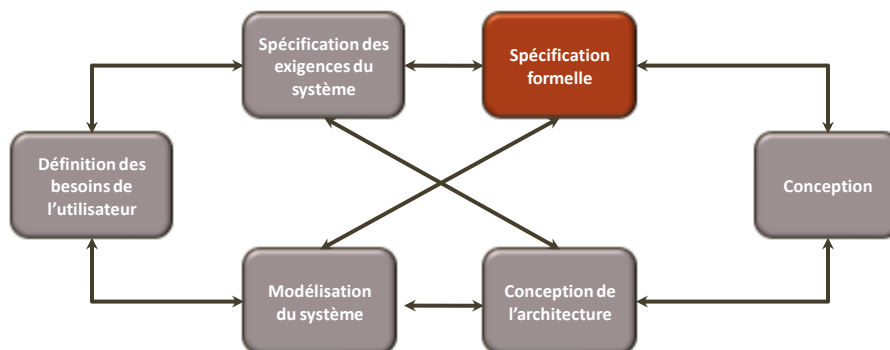
ENICAR

# Intégration des méthodes formelles dans la spécification

36

## a. Spécification formelle dans le processus logiciel

### ■ Processus parallèle



E. Menif Abassi

Spécification formelle

ENICAR

# Intégration des méthodes formelles dans la spécification

37

## a. Spécification formelle dans le processus logiciel

- ▣ Le processus de spécification est itératif et incrémental
  - À mesure qu'on avance dans la spécification, le domaine, le problème et les besoins sont mieux compris
  - Découvrir des erreurs, incohérences, incomplétudes, ambiguïtés, ...
  - Les problèmes découverts doivent être reportés au client et les changements doivent être réalisés avec l'accord de ce dernier

E. Menif Abassi

Spécification formelle

ENICAR

## Plan du chapitre

38

1. Spécification
2. Intégration des méthodes formelles dans la spécification
  - a. Spécification formelle dans le processus logiciel
  - b. Activités de la spécification formelle
  - c. Qualités de la spécification formelle
3. Classification des méthodes formelles de spécification

E. Menif Abassi

Spécification formelle

ENICAR

# Intégration des méthodes formelles dans la spécification

39

## b. Activités de la spécification

- La spécification est assurée par des spécifieurs. Pour la spécification formelle, les spécifieurs doivent avoir les connaissances nécessaires des langages formels
- Un spécifieur peut jouer le rôle:
  - Auteur de la spécification: Construction de la spécification avec le langage formel choisi  $\Rightarrow$  Maîtrise du langage et des propriétés du système
  - Analyste de la spécification:
    - Vérifier la spécification:
      - Exigences manquantes, redondantes ou incorrectes
      - Erreurs dues à une mauvaise utilisation du langage formel
      - Une mauvaise formalisation des besoins
    - Valider chaque opération de raffinement

E. Menif Abassi

Spécification formelle

ENICAR

# Intégration des méthodes formelles dans la spécification

40

## b. Activités de la spécification

- Des membres de l'équipe de spécification doivent assister dans:
  - L'utilisation des outils
  - La compréhension des problèmes du domaine
  - La compréhension de la spécification par le client et les développeurs
  - Démontrer au client que la spécification formelle saisit tous les besoins du cahier des charges
  - Définition des tests fonctionnels pour l'équipe de test (qui utilisent également la spécification formelle)

E. Menif Abassi

Spécification formelle

ENICAR

# Intégration des méthodes formelles dans la spécification

41

## b. Activités de la spécification

- La formalisation et la conception:
  - La formalisation se fait par raffinement des spécifications des besoins en ajoutant des détails:
    - Structures de données
    - Algorithmes
  - Processus de raffinement comprend également une activité de validation
  - La conception peut être partiellement formelle: uniquement les composants critiques sont formalisés

E. Menif Abassi

Spécification formelle

ENICAR

# Intégration des méthodes formelles dans la spécification

42

## b. Activités de la spécification

- La formalisation et l'implantation:
  - Implantation: développement du code, analyse du code et déploiement
  - Développement du code peut être:
    - Automatisé à partir de la conception: l'équipe de développement doit s'assurer que le code couvre les comportements exprimés dans la conception
    - Non automatisé: l'équipe de développement se base sur la conception détaillée pour développer un code qui la respecte
  - L'équipe de développement doit collaborer avec l'équipe de test pour réaliser les tests. En cas de problème, l'équipe de spécification et l'équipe de conception sont invités

E. Menif Abassi

Spécification formelle

ENICAR

# Plan du chapitre

43

1. Spécification
2. Intégration des méthodes formelles dans la spécification
  - a. Spécification formelle dans le processus logiciel
  - b. Activités de la spécification formelle
  - c. Qualités de la spécification formelle
3. Classification des méthodes formelles de spécification

## Intégration des méthodes formelles dans la spécification

44

### c. Qualités de la spécification

- ▣ L'évaluation de la qualité de la spécification formelle:
  - **Processus**: Point de vue du spécifieur
  - **Produit**: Point de vue du concepteur et du développeur

Comment peut-on évaluer la qualité du processus et du produit?

# Plan du chapitre

45

1. Spécification
2. Intégration des méthodes formelles dans la spécification
3. Classification des méthodes formelles de spécification
  - a. Méthodes orientées propriétés
  - b. Méthodes orientées modèles

## Classification des méthodes formelles de spécification

46

- La classification d'une méthode formelle repose sur 4 piliers:
  - ▣ Bases mathématiques:
    - Algèbre, logique, théorie des ensembles, ...
    - Extensions pour exprimer les caractéristiques d'un système (Structuré de composants, Générique, Paramétré, Bonne description des interfaces)
  - ▣ Type du système: Séquentiel, concurrent, distribué (Avec ou sans contraintes temporelles)
  - ▣ Degré de formalisme: Avec ou sans outil de vérification (Deux approches courantes de vérification: Preuve de théorème, Model-checking)
  - ▣ Outils: Vérification syntaxique, analyse et vérification sémantique

# Classification des méthodes formelles de spécification

47

- ▣ En plus des outils pour la vérification syntaxique et sémantique:
  - Outils pour le raffinement
  - Outils pour la vérification des propriétés: Les outils de vérification des propriétés du système dépendent des bases mathématiques utilisées dans le langage de spécification (ex: langage algébrique  $\Rightarrow$  vérification axiomatique)
- ▣ Deux catégories de méthodes de spécification formelle
  - Méthodes orientées propriétés
  - Méthodes orientées modèles

## Plan du chapitre

48

1. Spécification
2. Intégration des méthodes formelles dans la spécification
3. Classification des méthodes formelles de spécification
  - a. Méthodes orientées propriétés
  - b. Méthodes orientées modèles



# Classification des méthodes formelles de spécification

49

## a. Méthodes orientées propriétés

- Spécification = le comportement de ce système est décrit indirectement en listant des propriétés désirées dans ce système
- Construction d'un ensemble d'axiomes que le système doit satisfaire
- Deux types d'approches: axiomatiques et algébriques

E. Menif Abassi

Spécification formelle

ENICAR

# Classification des méthodes formelles de spécification

50

## a. Méthodes orientées propriétés

- Approches axiomatiques:
  - Les objets sont construits à partir de types
  - Les opérations du système spécifiées en termes pré-conditions (contraintes sur les entrées) et post-conditions (contraintes sur les résultats) avec la logique de premier ordre
  - Exemples: Anna, HOL (High-order Logic), PVS, Coq

```
Fixpoint fact (n:nat) :=  
  match n with  
  | 0 => 1  
  | S k => n * (fact k)  
end.
```

E. Menif Abassi

Spécification formelle

ENICAR

# Classification des méthodes formelles de spécification

51

## a. Méthodes orientées propriétés

### ■ Approches algébriques:

- Spécifier les objets ou types abstraits comme algèbres
- Spécifier la signature des fonctions manipulant les types abstraits
- Spécifier des équations mettant en relation ces fonctions, expliquant ainsi les propriétés qu'elles doivent vérifier
- Vérifier une propriété revient à démontrer que c'est un terme de l'algèbre
- Exemples: OBJ3, ACSR, CCS, Larch...

```
Spec: Natural;  
Sorts: Nat;  
Operations:  
  zero :  $\rightarrow$  Nat;  
  succ : Nat  $\rightarrow$  Nat;  
  add : Nat  $\times$  Nat  $\rightarrow$  Nat;  
  mult : Nat  $\times$  Nat  $\rightarrow$  Nat;  
Variables:  
  a, x : Nat;  
Axioms:  
  add(zero, a) = a;  
  add(succ(x), a) = add(x, succ(a));  
  mult(zero, a) = zero;  
  mult(succ(x), a) = add(mult(x, a), a);
```

E. Menif Abassi

Spécification formelle

ENICAR

## Plan du chapitre

52

1. Spécification
2. Intégration des méthodes formelles dans la spécification
3. Classification des méthodes formelles de spécification
  - a. Méthodes orientées propriétés
  - b. Méthodes orientées modèles

E. Menif Abassi

Spécification formelle

ENICAR

# Classification des méthodes formelles de spécification

53

## b. Méthodes orientées modèles

- Spécification = **modèle abstrait** décrivant le comportement du système
- Construction d'un modèle du système en terme d'*ensembles*, *séquences* et *relations*
- Ces objets mathématiques fournissent une description abstraite de l'état d'un système et des opérations sur celui-ci
- Une opération sur un état permet de modifier des attributs de ce dernier
- Exemples: Z, Object-Z, B, VDM, CSP, Réseau de Petri,...

E. Menif Abassi

Spécification formelle

ENICAR

# Classification des méthodes formelles de spécification

54

Type	Séquentiel	Concurrent
Orienté propriétés (Axiomatique, Algébriques)	OBJ, Larch	Lotos, ACSR
Orienté modèle	Z, VDM, B	CSP, Réseau de Petri

E. Menif Abassi

Spécification formelle

ENICAR