Construction Formelle de Logiciels

J. Christian Attiogbé

Master Alma, 2012

Enseignements : IUT de Nantes - UFR Sciences

Dpt Info IUT

- Modélisation de donnée
- modélisation de la dynamique
- Méthode B

Dpt Info UFR Sciences

- Construction formelle de logiciels

LINA - Research Topics

- axe ALD: AeLoS, Ascola, GDD, GRIMM
- axe SAD: COD, COMBI, Contraintes/TASC, OPTI, TALN

AeLoS

P. André, G. Ardourel, C. Attiogbé, A. Lanoix, J-M. Mottu, M. Oussalah, D. Tamzalit + M. Ouederni + doctorants

Introduction

- Système informatique ?
- Développement ?
- Quels problèmes ?
- Quels concepts, théories, méthodes, techniques, outils ?
- Etat de l'art ?
- Besoins ?

Présentation du module

Modélisation et vérification formelles des logiciels
(seule façon de pouvoir prouver la correction de logiciels)

- Méthode B/Atelier B/Rodin (avec la preuve de théorème)
- Lotos NT/CADP (avec l'exploration de modèles)

Motivations

Niveau MASTER ⇒ Conduite de projets industriels informatiques
Domaines variés, tailles variables (petites ou grandes)
Projets informatiques complexes ⇒ Méthodes, Techniques, Outils

- Méthodes d'analyse,
- Méthodes de conception,
- Méthodes de développement.

Motivations (suite)

Exemples de méthodes

- Analyse fonctionnelle (SADT par exemple),
- Analyse structurée (SA, SSADM), SA-RT (Temps-Réel),
- Entités/Associations, Merise, Axiale,
- JSD/JSP,
- Analyse Orientée Objet, OMT, UML,
- Architecture de logicielle (System Level),
- etc

$\Rightarrow$ méthodes semi-formelles

Motivations (suite)

Besoin de méthodes rigoureuses pour certains domaines :

- Sécurité, Certification, Coût, Maintenance
- ITSEC (Information Technology Security Evaluation Criteria) exigent l'usage de méthodes **formelles**
- Echec (d'un vol) de ARIANE !, Erreur du Pentium, etc, etc
- Milieux hostiles à l'homme (nucléaire, chimie, marin, etc)
- Systèmes embarqués (véhicules, équipements, etc)
- Automates (domaine médical, etc)
- etc

Motivations

Les méthodes formelles ⇒

- garantie de correction des logiciels,
- diminuent/éliminent les erreurs, les disfonctionnements,
- facilitent la maintenance.

Méthodes formelles : introduction

Méthodes de développement des systèmes informatiques.
Quelques analogies :
Génie civil
→ Architecture, plans (conception), calculs, construction (réalisation)
Physique
→ Observations, Modélisation, études sur les modèles, réalisation
Informatique
⇒

Analyse des besoins (observations?)

Modélisation,

études des modèles,

réalisation du système

Généralités

Différentes **approches d'utilisation des méthodes** formelles :

- à postériori : On développe (programmation) puis on vérifie que le produit est correct
  → Systèmes de preuve, systèmes de test
- à priori : On développe correctement le produit
  → Méthodes de développement (raffinement, synthèse),
  Systèmes de preuve

Plusieurs méthodes formelles (langages, systèmes de preuve, méthodes)

Généralités

- **Approche** top-down **(descendant)**
  On procède par **décomposition**
  - Analyse globale (étude système, ingénierie de système)
  - Architecture de logiciel
    ↓
  - Codage des composants
    - Programmation ou
    - Développement formel

Généralités

- **Approche** bottom-up **(ascendant)**
  On procède par **composition** de composants élémentaires.
  - Etude des composants disponibles
  - Composition, Réutilisation

Besoin des méthodes formelles

Dans tous les cas (approches) recours au méthodes formelles pour

- Etude des systèmes
- Etude des composants
- Cadre formel pour le raisonnement, analyse, développement

Qu'y a-t-il dans les méthodes formelles ?

- Logique
- Algèbre
- Mathématiques discrètes
- Théorie des ensembles
- Théorie des automates
- Théorie des types
- Théorie du raffinement
- ...

Exemples d'applications industrielles

avec les formalismes/méthodes Z, VDM, CSP

- IBM, INMOS, ...
- CICS: Système transactionnel interactif (1983, Z)
- Conception de circuits, Transputer (Z, CSP)
- et de nombreux autres systèmes

Exemples d'applications industrielles

avec la méthode B (J-R. Abrial)
GEC ALSTHOM, SNCF et MATRA Transport

- Système de controle de vitesse de train (KVS pour SNCF)
- Ligne A du RER - SACEM (signalisation, contrôle de vitesse)
- Metro de Calcutta (CTDC)
- Metro de Montreal(CTDC), Marseilles, Bel horizonte
- Météor (ligne de Métro sans conducteur)
- les portes pallières sur les quais
- Assurance vieillesse, Sécurité sociale
- CICS de IBM (restructuration majeure du logiciel de gestion de transaction, 800000 lignes)
- B et VDM dans le domaine des logiciels de finance, BULL UK

Exemples d'application

Système de contrôle de vitesse (Metro)

- acquisition de données (capteurs, détecteurs, etc),
- 'calcul'/prise de décision,
- envois de commandes aux dispositifs physiques (ralentissement, freinage)
- embarquement du logiciel

Autres approches formelles utilisées aujourdhui

Certaines (outillées) sont industrialisées

- RAISE (Résultat d'un projet ESPRIT)
  Approche algébrique + *processus communicants*
- LOTOS, SDL (Standard européen)
  Approche algébrique + *processus communicants*
- PVS (USA)
- MEC, AltaRica (Université de Bordeaux + industriels)
- Logiques classiques : Logique du premier ordre, Logique de Hoare, etc (Why, Cracatoa, Key -)
- Logiques non classiques, logiques modales
- Coq, Logiques d'ordre supérieur, Théorie des types

Fondement des approches formelles (preuve)

Interprétation de l'isomorphisme de Curry-Howard :

$$\text{Preuve}\ \frac{Axiomes}{Theoremes} \quad \text{équiv. à} \quad \text{Développement}\ \frac{Specifications}{Programmes}$$

♡

Approche de construction formelle

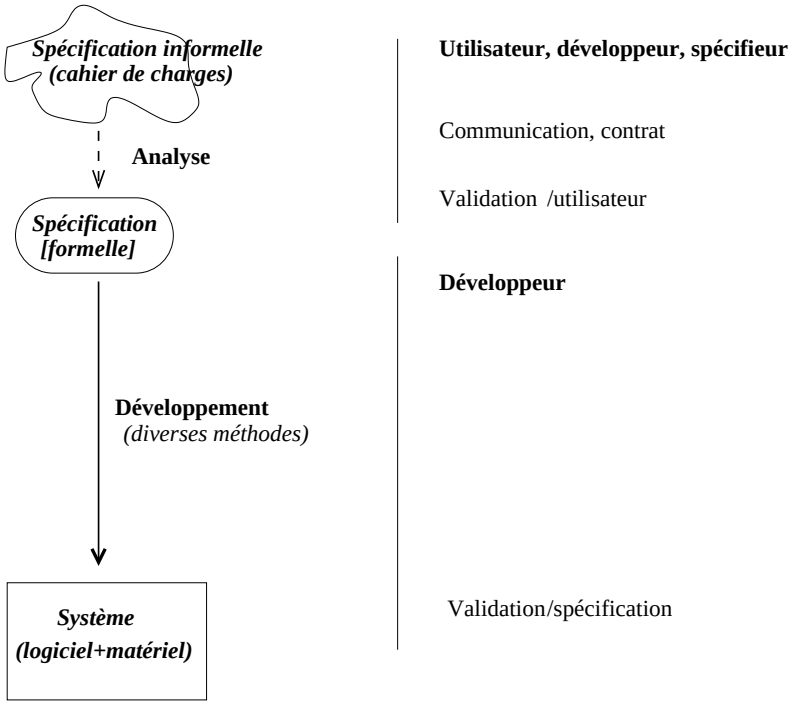

Figure: Problématique du développement de système

Usage des méthodes formelles

- Marteau pour tuer une mouche
  - Selon la nature des besoins
- Environnement professionnel
  - compétences disponibles ?
- Contexte industriel
  - Délais, coûts, productivité
- Certification
  - Obligations des donneurs d'ordre

Quelle approche utiliser ?

→ plusieurs paramètres :

- Concepteur/développeur de gros systèmes
- Concepteur/développeur de petits systèmes "maison"
- Nature des systèmes à développer
- Compétences disponibles,

...

Cas de la méthode B

Le système développé correct par construction

- Initialement, Modèle d'exécution séquentielle - Systèmes séquentiels
  Pas de non-déterminisme de comportements
- Systèmes autonomes et réactifs
- Systèmes centralisés et distribués

Cas de LOTOS

Analyse et vérification du système développé

- Modèle d'exécution séquentielle et parallèle
  Systèmes séquentiels et concurrents
- Possibilité de non-déterminisme de comportements
- Systèmes autonomes et réactifs
- Systèmes centralisés, systèmes distribués

Etapes du Cycle de vie

Cahier de charges (spécification informelle) → système informatique
Plusieurs étapes :

- Analyse (*Analysis*),
- Spécification, Modélisation (*Specification, Modelling*),
- Conception (*Design*),
- Implantation (*Implementation*)
- Maintenance (*Maintenance*)

Catégories de systèmes logiciels

- séquentiels, parallèles (concurrents),
- autonomes (transformationnels), réactifs, temps-réels
- centralisés, répartis,
- embarqués,
- protocoles de communication
- ...

⇒ plusieurs types de systèmes

Difficultés

- Décrire précisément le système voulu spécification
- Construire correctement le logiciel développement
- S'assurer que le logiciel construit est correct par rapport aux besoins
- Suivi du système

Chaque projet est déterminant

- Nature des systèmes *complexes* → multifacette
- Plusieurs méthodes :
  - Méthodes semi-formelles
  - Méthodes formelles (intégrées) → traitement des systèmes complexes

⇒ maîtrise de plusieurs méthodes

Quelques définitions

Modélisation :
Hoare : A scientific theory is formalised as a mathematical model of reality, from which can be deduced or calculated the observable properties and of a well-defined class of processes in the physical world.
Il y a deux principales notions de modèles (en informatique).

1. Modèle = une approximation de la réalité par une structure mathématique.
   Un objet O est modèle d'une réalité R, si O permet de répondre aux questions que l'on se pose sur R.

En Mathématique, Physique, ... système d'équations portant sur des grandeurs (masses, énergie, ...) ou des lois hypothétiques.

Quelques définitions (suite)

2. (Logique, théorie des modèles)
   Un modèle d'une théorie T est une structure dans laquelle les axiomes de T sont valides.
   Une *structure* S est modèle d'une théorie T, ou bien S satisfait T si toute formule de T est satisfaite dans S.
   La réalité est un modèle d'une théorie !
   Théorie (du 1er ordre) = tout ensemble de formules logiques (du 1er ordre) dans un langage donné (précisément défini).
   Modèle comme interpétation d'une spécification - une algèbre comme modèle d'une spécification algébrique (axiomatisation).

Quelques définitions

Ces deux utilisations de *modèle* se retrouvent dans les approches *orientée modèle (ou état)* et *orientée propriétés*.
Dans le langage courant,

- *modèle* = (archétype), ce qui sert ou doit servir d'objet d'imitation pour reproduire quelque chose.
- *modèle* = (paradigme), modèle de déclinaison, de conjugaison, etc
- *modèle* = (référence), ...

Exemples de théorie :

La théorie des ensembles : elle est basée sur un ensemble d'axiomes.
Les objets de cette théorie sont appelés ensembles.
La classe des ensembles est appelée univers.
Les axiomes de la théorie des ensembles (de Zermelo+Fraenkel) sont les suivants:

Quelques définitions

- Axiome de l'ensemble vide : *il existe un ensemble qui ne contient aucun élément : c'est l'ensemble vide.*
- Axiome d'extensionalité : *deux ensembles sont égaux si et seulement si ils contiennent exactement les mêmes éléments*
- Axiome de l'union : *l'union d'ensembles est un ensemble*
- Axiome de l'ensemble des parties : *les parties d'un ensemble forment une partie*
- Axiome du schéma de remplacement (Fraenkel, 1922) : *Lorsqu'on définit une fonction par des formules de la théorie des ensembles, alors éléments pour lesquels cette fonction vérifie une certaine propriété forment encore un ensemble.*

De plus, on ajoute à ces axiomes, l'axiome de l'infini : *il existe un ordinal infini.*
ZFC = ZF + axiome du choix

- Axiome du choix : *Soit une famille d'ensembles disjoints, si on considère un élément de chaque ensemble de la famille, alors on en forme un ensemble.*

Quelques définitions (suite)

Méthode semiformelle =

- Langage graphique [+ formel]
  (syntaxe précise et sémantique non précise) et
- Outils d'analyse divers.

→ Combinaison de langages/méthodes/techniques n'ayant pas tous une sémantique précise.

Exemples : JSD, OMT, OOX, UML

Quelques définitions (suite)

Méthode formelle =

- Langage formel (syntaxe précise et sémantique précise) et
- Système de preuve ou de raisonnement formel.

Exemples : CCS, CSP, HOL, Z, B

Développement formel =

- transformation systématique des spécifications en programmes en utilisant des lois prédéfinies.

Exemples : Synthèse, Raffinement

Quelques définitions (suite)

Vérification : montrer que le système (S)est correct par rapport à des propriétés(P)

$$S \vDash P$$

Validation : montrer que le système est correct par rapport aux spécifications informelles

$$S \sim S_{informelle}$$

Raisonnement formel : Consiste à **appliquer un système formel à une spécification**.

Exemples de raisonnement formel

- Rafinement de specification,
- vérification des proprétés d'un système,
- validation par vérification,
- preuve de théorèmes (*theorem proving*),
- analyse d'un système (représenté par machine à états) par rapport à des propriétés (*model checking*).

$\Longrightarrow$ La logique est le fondement des approches formelles

- pendant longtemps, support méthodologique du développement du logiciel.
- les plus représentatifs de ces cycles de vie :
  - Cycle en V,
  - Cycle en cascade,
  - Cycle de Balzer

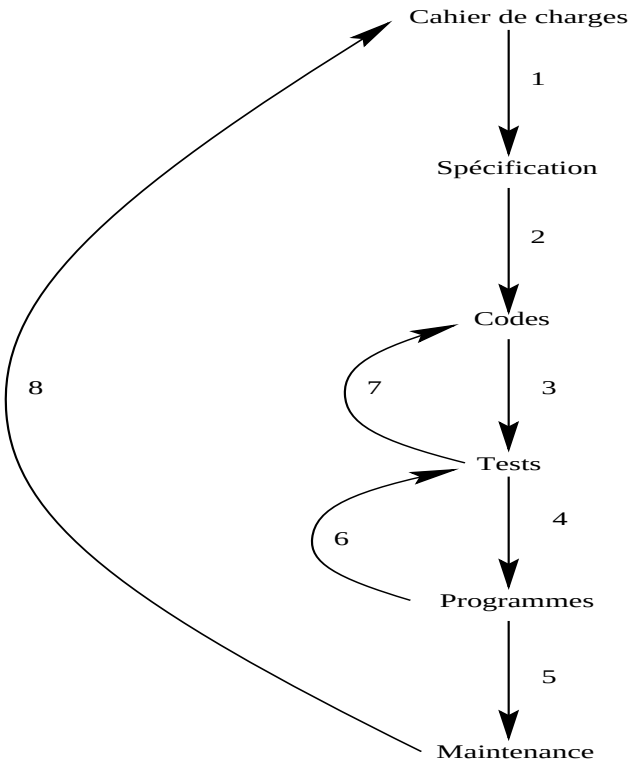

Figure: Cycle de vie en cascade (Boehm 1977)

Intérêts et limites des méthodes semi-formelles

- SADT, SA-RT, SSADM, ...
- JSD-JSP,
- Merise, Axial, ...
- OOA, OMT, UML
- ...

L'analyse du problème est faite.
Contribution positive même si suffisante.
Le problème est dégrossi.
→ impossible de raisonner formellement sur le système en vue.
→ Il peut y avoir des ambiguités, des erreurs.

Spécification formelle

⇒ Expression dans un langage formel du quoi d'un système à développer.

- Résultat de la phase d'analyse
- Plusieurs formes possibles selon la nature du système

On parle de **langages** ou **formalismes** de spécification formelle :
Logique, Z, Langages de spécification algébriques, algèbres de processus, etc

Démarche de spécification

Les aspects données ou les aspects opérations ?

- **Les données d'un système permettent de décrire les états du système**
- **les opérations du système permettent de décrire son fonctionnement ou son comportement par des axiomes**

On parle du paradigme des données et du paradigme des opérations.
Il convient de distinguer les opérations exprimant le comportement d'un système des opérations caractérisant les données du système.
Les premières opèrent sur les données du système alors que les dernières permettent de construire et exploiter les données du système.

Démarche de spécification

Il y a un troisième paradigme qui semble transversal.
C'est le paradigme des processus (les algèbres de processus).
Dans ce paradigme des processus les systèmes sont décrits par des règles ou des équations exprimant leur comportement ou leurs états.
Les principales algèbres de processus à la base des autres formalismes sont :

- CSP (Hoare)
- CCS (Milner)
- ACP (Bergstra)