La Modélisation des Systèmes d'Information avec la méthode formelle B.

Mme Mellah Hakima
Laboratoire Bases de données et systèmes d'information
Divison systèmes d'information Cerist.
e-mail: Hmellah@tassili.cerist.dz

Introduction :

Sans doute du fait que les méthodes sont devenues en Informatique des produits de commerce ,les références formelles ont été abandonnées ,si on ne dispose pas de sémantique bien définie ,si on ne dispose pas de capacité télépathique ,il faut faire appel à l 'expert » [Hab 95].

Dans ces quelques mots si riches que le fondateur de la méthode B J-R Abrial a prononcé lors d'une conférence , nous retrouvons une bonne introduction aux méthodes formelles en général .Les méthodes formelles peuvent combler la lacune relative à l'absence de sémantique précise et réciproquement bénéficier des méthodes semi_ formelles ,c'est en particulier le cas des méthodes orientées modèles comme B dont le principe est de construire un modèle mathématique du système en utilisant la logique des prédicats et la théorie des ensembles. Comme la majorité des méthodes formelles ,B est actuellement et principalement utilisée dans le cas de systèmes sécuritaires notamment connus ou exploités dans les domaines où la sûreté de fonctionnement constitue une préoccupation majeure comme le domaine de l'énergie, du nucléaire ,de l 'espace ,de l'avionique ou des télécommunications .

Dans ce type de domaines ,la moindre défaillance d'un logiciel de sécurité peut entraîner des pertes humaines; les méthodes classiques de développement ont du mal à maîtriser la sûreté de fonctionnement de logiciels .En effet les différentes phases du cycle de vie d'un logiciel utilisent des formalismes distincts .Les méthodes formelles, c'est à dire notamment la méthode B, constituent une nouvelle approche qui apporte sous certains aspects ,des solutions à la maîtrise des développements de logiciels critiques de sécurité .

2. Description du principe fondamental de B :

La méthode B tout au long du cycle de vie d'un logiciel utilise un formalisme unique connu sous le nom du Langage B .En effet B découpe le système en plusieurs machines abstraites AMN (Abstract Machine Notation) ;chaque AMN subira des raffinages successifs jusqu'à obtention d'un pseudo - code qui sera facilement transcodé en un langage de programmation impératif .Ainsi l'implantation peut
Se faire sur les spécifications d'une ou de plusieurs machines abstraites raffinables ; de cette manière ,un projet se construit peu à peu selon une architecture en couches.

2.1. Langage de spécification formelle B (Langage B) :

Le langage B est un langage composé de :
- La théorie des ensembles
- La logique des prédicats
- Et le principe des substitutions généralisées

l'exemple ci dessous a été inspiré du B-Book , il représente la machine abstraite personne identifiée par MA-PERSONNE

MACHINE
MA-PERSONNE
SETS PERSONNES , SEXE={M,F}
VARIABLES
Personnes , Sexe ,Nom
INVARIANT
Personnes $\subseteq$ PERSONNES /\
Sexe $\in$ Personnes $\rightarrow$ SEXE /\
Nom $\in$ Personnes $\rightarrow$ STRING

OPERATIONS
Ajout-personne(p,sexe,nom)
Pre p $\in$ PERSONNES – Personnes /\
sexe $\in$ SEXE /\
nom $\in$ STRING

Then Personnes :=Personnes U {p} ||
Sexe(p) := sexe ||
Nom(p) := nom
End ;

```
Supp_personne(p)
Pre p ∈ Personnes
Then  Personnes := Personnes – {p}  ||
Sexe   :={p} ≤  Sexe_     ||
Nom :={p}   ≤  Nom
end

END
```

PERSONNES : ensemble abstrait de toutes les personnes possibles.
Personnes :variables d'état de l'ensemble des personnes existantes
Sexe :fonction totale de Personnes dans STRING.
Les opérations sont décrites par une sorte de pseudo code ensembliste mais avec une construction non exécutable tel que $\rightarrow$, $\subseteq$, $\in$, $\underline{\leq}$, || ; signalons que les opérateurs :
|| :dénote l'exécution en parallèle de 2 instructions
$S \underline{\leq} F$: restriction de F à domaine(F) –S
Une machine M peut inclure une autre machine M' en déclarant **INCLUDES** M' ;
M a donc trait à appeler les opérations de M' ,les variables de M' ne sont visibles qu'en lecture pour M ; une machine ne peut être incluse que dans une seule autre machine ;On ne peut appeler qu'une opération de la machine incluse dans une opération de la machine appelante ;tous ces critères se rassemblent pour que **l'INCLUDES** n'influera pas sur l'invariant de la machine M .Une machine M peut utiliser une machine M' en déclarant : **USES** M' ;avec le **USES** les variables de M' sont visibles (en lecture)dans M.

2.2.Principe du raffinage :

L'opération du raffinage se fait en une ou plusieurs étapes successives selon la complexité de la machine à raffiner et consiste à reformuler les variables et les opérations de la machine abstraite ; tout au long du raffinage on passera des données abstraites aux données plus concrètes, et la dernière étape de raffinage est l'étape d'implantation ,dans le tableau ci dessus nous résumons en quoi consiste les différentes actions de raffinage aussi bien sur les données du système(aspect statique) que sur les traitements(aspect dynamique).

	Raffinages successifs	Dernier raffinage (implantation)
Données	. transformation des variables en variables plus concrètes .ajout de variables (détails de conception)	.transformation des variables en structures de données programmables .valuation des ensembles et des constantes
Opérations	.réduction du non déterminisme .affaiblissement des pré-conditions .introduction du séquencement .ajout des détails de conception	.suppression du non déterminisme .suppression des pré-conditions .suppression de la simultanéité .ajout des détails de conception .introduction des itérations

3. Apport de la preuve :

Parmi les caractéristiques de B , nous citons la preuve qu 'elle nous apporte dans la spécification d'un système. En fait à quoi revient à faire une démonstration de preuve en B, et quel intérêt nous apporte t-elle ?
Ce qui est démontré dans une preuve d'une MA c'est que les pré conditions citées dans ses opérations doivent préserver l'invariant de la machine qui est en fait la conjonction de l'ensemble des variables nécessaires dans le système à développer (propriétés souhaités du système); lors de la définition des opérations il ne faut en aucun donner un prédicat qui contredit l 'invariant de la MA, ce qui peut se déduire de cela : l'échec d'une preuve alors que le modèle B (fig 2) est cohérent montre la non faisabilité d'une partie du cahier des charges et ce très tôt dans le cycle de vie du système .La démonstration de preuve d'un système formalisé avec B permet d'établir la cohérence totale du cahier des charges.

Dans la figure ci dessous nous schématisons la modélisation d'un système d'information durant tout son cycle de vie par B.

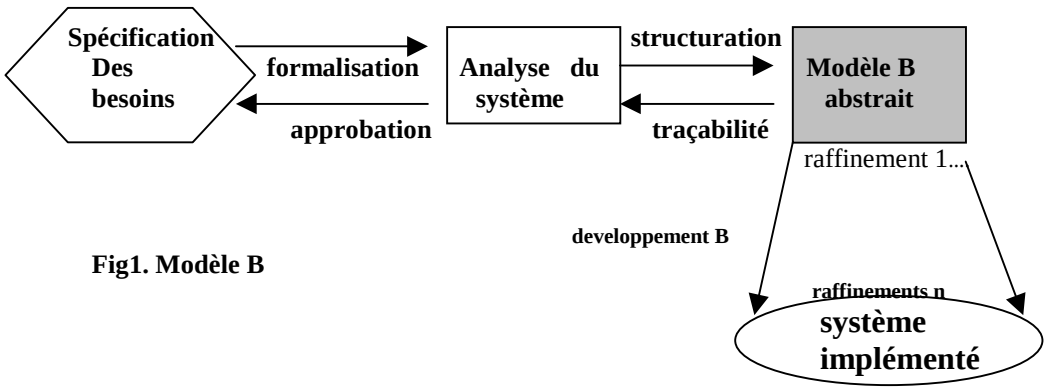

Fig1. Modèle B

3.1.formalisation :

les besoins sont réécrits dans un formalisme mathématique plus ou moins formel .

3.2.approbation :

En cas de conflit avec l'utilisateur sur des données ou des traitements, il est recommandé de revoir le cahier des charges.

3.3.structuration :

L'analyse du système permettra sa structuration en sous systèmes clairs et analysables appelés en B par machine abstraite(élément de base d'une modélisation B).

3.4.traçabilité :

Les propriétés logiques présentes dans l'analyse du système et dans le modèle B doivent être littéralement identiques.

3.5.développement B : Au fur et à mesure que les raffinages successifs s'effectuent sur le modèle abstrait (développement B) nous constatons la constitution progressive d'un code de programme exécutable du système à développer.

La démonstration de preuve dans la phase de raffinage assure dans un sens le respect du modèle abstrait jusqu'à l'implantation finale du système

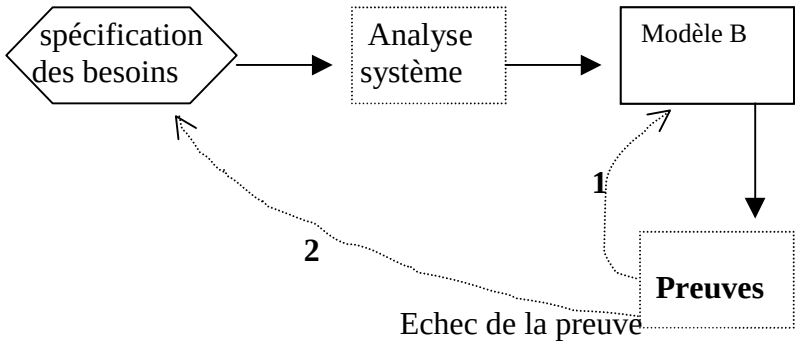

fig.2.

Dans le cas d'échec de la preuve (fig.2) ,ce qui revient à ce que un prédicats cité dans une des opérations de la MA ne préserve pas son invariant ,il faut donc revoir la spécification des opérations. Une autre possibilité, dans le cas où le modèle B est tout à fait correct donc ce qu 'il faut revoir c'est le cahier des charges qui aurait dû être mal rédigé .

4.Peut on simuler certains concepts en B avec des concepts de l'orienté objets ?

On trouve fréquemment dans les méthodes de conception des systèmes d'information une notion d'héritage ;en réalité ,il n existe aucune équivalence directe de l'héritage dans les méthodes formelles « usuelles » ; certaines extensions comme VDM++ et Object Z * ont trait à utiliser en parallèle avec les concepts de la modélisation formelle la notion d'héritage de l'orienté objets ;

5. L'Atelier B:

L'atelier B est un outil de génie logiciel qui permet une utilisation opérationnelle de la méthode B, il est commercialisé par Stéria Méditerranée qui l 'a developpé avec la collaboration de J_R Abrial fondateur de la méthode et GEC Alsthom transport, parmi les performances de l'Altelier B nous citons :
- Les analyseurs statiques (Type checker, B0 checker ,le vérificateur de projets)
- le prouver (démonstration automatique des obligations de preuve)

- les traducteurs automatiques (existent actuellement en C et en Ada)
- la représentation graphique de projets
- les bibliothèques réutilisables
- le générateur de documentation (génère un dossier complet sur le projet)
- le navigateur hypertexte (permet la navigation dans les composants du projet)
- la gestion de projets
- l' animateur (simule le fonctionnement du système dans une phase préléminaire)

Conclusion :

La particularité de B par rapport aux autres méthodes formelles c'est qu 'elle couvre tout le cycle de vie du logiciel à développer dans un cadre formel uniforme. L'apport de la preuve dans B présente l'avantage que le logiciel produit respecte la spécification puisque il en découle totalement .Elle a toutefois l'inconvénient de ne pas fournir de guide de réalisation aussi précis et mûr que certaines méthodes* du marché et c'est dans ce but que les études de cas peuvent être d'un grand intérêt dans l'utilisation de la méthode.

- :ces méthodes (semi formelles) bien qu'elles ont l'avantage cité, comme inconvénient de B ne nous garantissent en aucun cas la validité d'une phase du cycle de vie d'un logiciel par rapport une autre .

Références Bibliographiques

- The B-Book "assignig programs to meaniong" J.Raymond .Abrial.1996

- Les apports de la preuve en B :C. Roques – F. Bustany –D.Sabatier
  Stéria Méditerranée

- Des pécifications informelles aux spécifications formelles : Compilation ou interprétation P.Facon – R.Laleau CNAM

- Comprendre les méthodes formelles panorama et outils logiques
  Jean françois Monin

- [Hab 95] Z twenty years on – What is its futur ?
  actes de la conférence Nantes ,Sept 95 H.Habrias (éditeur)

- La méthode B :Concepts,Démarche ,exemples d'application
  Karl Emeriau.