Processus itératif de conception de modèles d'informations de gestion

Jean-Philippe Martin-Flatin
jp.martin-flatin@ieee.org

Université de Fribourg, 18 mars 2002

Recherche en collaboration avec Divesh Srivastava (AT&T Labs Research) et Andrea Westerinen (Cisco Systems)

Plan

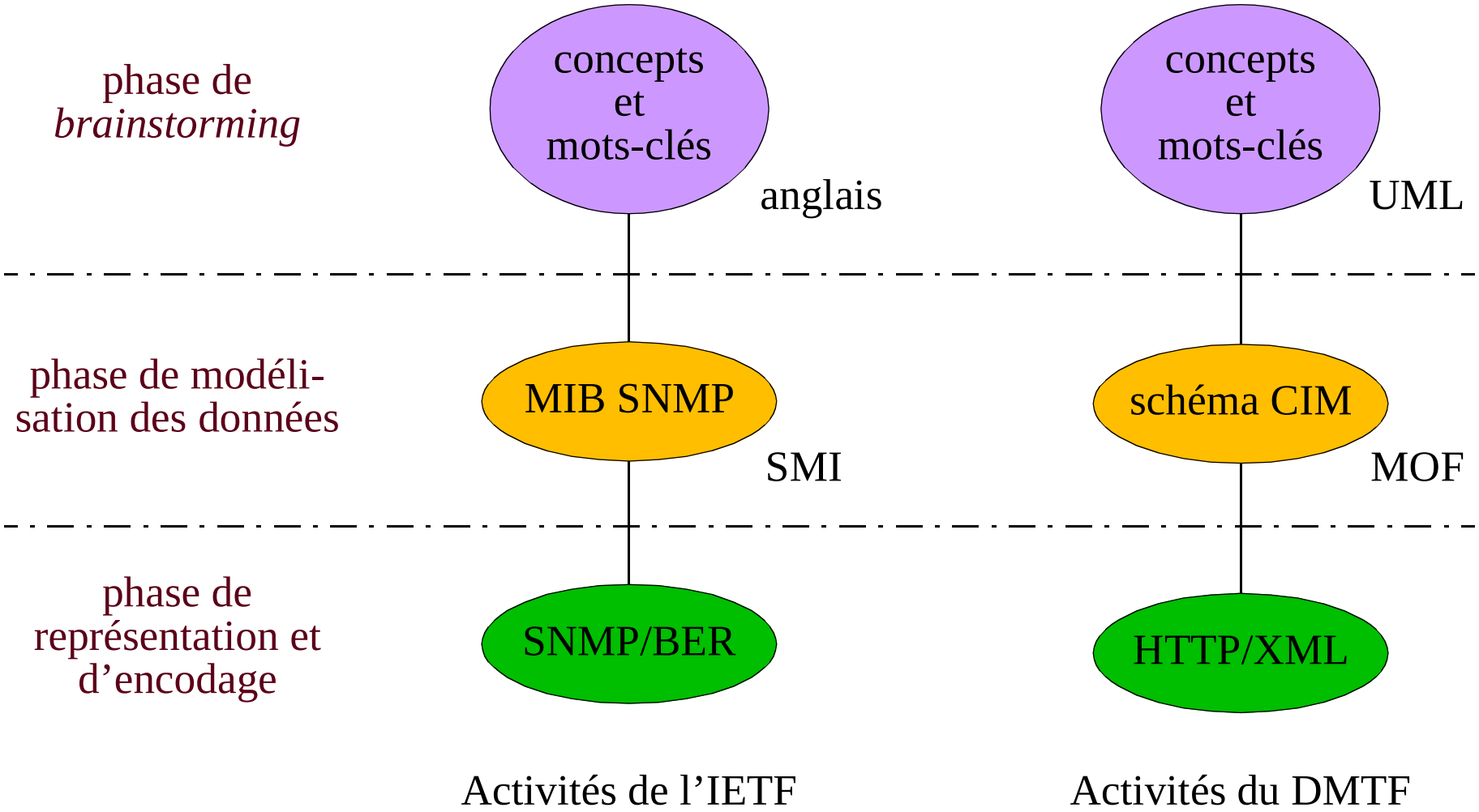
- Modélisation des info. de gestion
- Quatre problèmes
- Analyse
- Modèles multi-couches
- Processus itératif de conception
- Avantages du nouveau processus
- Conclusion

Modélisation des informations de gestion

Activités de normalisation indépendantes de toute technologie

- Métamodèle :
  - DMTF : variante du métamodèle d'UML
    - → classe, objet, association, etc.
  - IETF : métamodèle implicite
    - → dans une MIB, tout est un OID (*Object IDentifier*)
- Langage :
  - MIBs SNMP : SMI
  - PIBs SNMP (politiques) : SPPI…
  - schémas CIM : MOF
- Représentation et encodage des données de gestion :
  - IETF : BER
  - DMTF : XML, *CIM Operations over HTTP*

Activités de normalisation propres à chaque technologie



Quatre problèmes

Qualité insuffisante de certains modèles d'informations de gestion

- Certains modèles contiennent des erreurs :
  - ex. : la RFC 1156 a été immédiatement remplacée par la RFC 1213 à cause de l'*Address Translation Group*
- Des fonctionnalités importantes sont oubliées dans certains modèles d'info. de gestion :
  - ex. : la RFC 1213 ne permet pas de définir de contrôles d'accès (ACLs) pour chaque interface d'un équipement réseau
  - du coup, on doit utiliser `telnet`

Pourquoi cette qualité insuffisante ?

- WGs dominés par les fournisseurs (notamment les équipementiers) :
  - mauvais compromis entre qualité et rapidité
  - *fast design is not beautiful*
- Les efforts de normalisation dans le domaine de la gestion ont beaucoup de mal à attirer :
  - les meilleurs experts en technologies
  - les meilleurs modélisateurs
- Les cahiers des charges sont souvent assez flous :
  - ex. : de quelles info. de gestion a-t-on besoin pour gérer des réseaux privés virtuels à commutation d'étiquettes (*MPLS-based VPNs*) ?

L'antipatterne "réinventer la roue"

- Les organismes de normalisation travaillant en gestion de réseaux, de systèmes et de services sont légion :
  - IETF, DMTF, OMG, TMF, ISO, ITU-T, TOG...
- Peu d'interactions et de pollinisations croisées :
  - syndrome du "ça ne vient pas de chez nous"
  - pas le temps de lire la littérature technique -> recommence à zéro
- Conséquences :
  - la terminologie n'arrête pas de changer :
    - → par ex., évènements, notifications et indications redéfinis par le DMTF
    - → ceci cause une certaine confusion chez les clients
  - les org. de normalisation perdent un temps précieux

Trouver le bon niveau d'abstraction entre deux extrêmes

- Modèles trop abstraits :
  - ex. : l'architecture de métamodèle en 4 couches de l'OMG
  - quelle utilité en pratique ?
  - conçu par des théoriciens pour des théoriciens
  - antipatterne "*over-engineering*"
- Modèles trop détaillés :
  - ex. : de nombreuses MIBs SNMP
  - concepts de base cachés par les détails
  - conçus par des développeurs peu intéressés par l'architecture d'une application
  - antipatterne "*under-engineering*"

Phase d'apprentissage trop difficile

- Les débutants ou nouveaux venus croulent sous les détails lorsqu'ils découvrent un modèle d'info. de gestion :
  - ils doivent parfaitement maîtriser le langage SMI (resp. MOF) pour comprendre ce qui est modélisé dans les MIBs SNMP (resp. schémas CIM)
- Les débutants ou nouveaux venus ont besoin d'une approche plus graduelle :
  - qui mette d'abord en avant les points principaux du modèle
  - qui présente ensuite les détails

Analyse

Que nous enseigne le génie logiciel ? (1/2)

- Avec des modèles d'info. de gestion mono-couches, on essaie d'en faire trop d'un coup et on demande trop de compétences aux mêmes individus :
  - il faut séparer les modèles de conception, de spécification et d'implémentation (cf. séparation entre analyse, conception et implémentation dans le monde OO).
- Les problèmes de gestion d'une technologie donnée sont orthogonaux à l'architecture de gestion (SNMP, WBEM, modèle de gestion OSI...) :
  - il faut isoler ce qui est indépendant de l'architecture de ce qui ne l'est pas :
    - → réutilisation facilitée
    - → modèle conceptuel plus élégant
    - → moins de risques de changements terminologiques

Que nous enseigne le génie logiciel ? (2/2)

- La qualité du logiciel est garantie en attirant les meilleurs experts à chaque étape du processus de développement du logiciel :
  - il faut attirer les meilleurs experts et les meilleurs modélisateurs dans les organismes de normalisation des modèles d'info. de gestion
- Le processus de modélisation en cascade (*waterfall*) n'est approprié que dans les cas les plus simples :
  - quand les problèmes de gestion deviennent plus ardus, il faut migrer vers un processus itératif où l'on modélise de façon incrémentale

Contraintes pratiques (1/2)

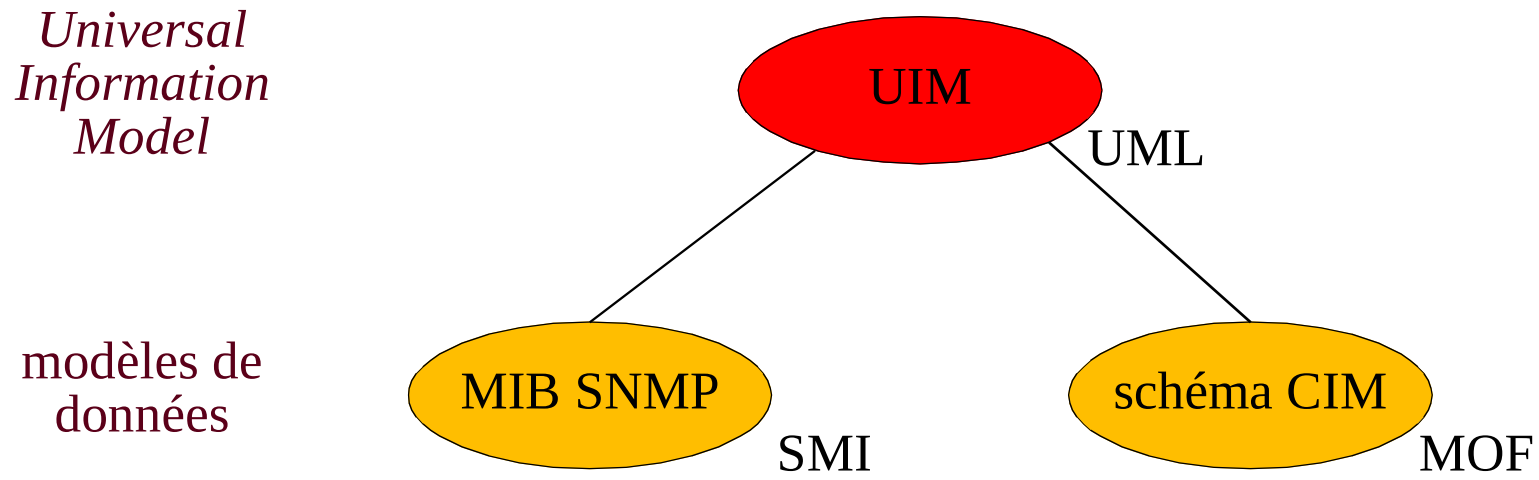
- Dans le monde IP, les plateformes de gestion coûtent bien plus cher aujourd'hui qu'au milieu des années 1990. Le risque encouru est plus grave. Du coup, de nombreux clients exigent aujourd'hui que ces plateformes respectent les normes de gestion, alors que ce n'était pas le cas auparavant.
  - normes = "polices d'assurance"
- Le processus de normalisation des modèles d'info. de gestion ne doit pas substantiellement retarder la mise sur le marché des nouveaux équipements. Le marché IP est trop compétitif pour s'offrir un tel luxe.

Contraintes pratiques (2/2)

- Le redéploiement d'une MIB SNMP ou d'un schéma CIM coûte immensément cher aux fournisseurs comme aux clients. Tout doit être fait pour éviter une telle opération.
  - ceci concerne les erreurs ou oublis dans le modèle
  - mais pas les variations dans le cahier des charges
- La plupart des clients exigent d'avoir des applications de gestion de haute qualité dès qu'ils achètent un nouvel équipement. Les grands NOCs ne peuvent plus se permettre de déployer immédiatement de nouveaux équipements et de les intégrer quelques mois plus tard à leurs plateformes de gestion.

Modèles multi-couches

Exemple : deux couches



Un seul UIM par technologie (1/2)

- UIM = modèle abstrait orienté objet
- Exprimé en UML + livres blancs (*whitepapers)*
- Le but d'un UIM est de faire comprendre à des êtres humains (et non à des machines ou des compilateurs) les principaux concepts d'un modèle d'info. de gestion :
  - les détails sont ignorés
- Les UIMs sont indépendants de l'architecture de gestion :
  - indép. de la base de données
  - indép. du protocole de communication
    - → le modèle de communication et le modèle informationnel sont indép.
- Utilise le métamodèle UML de l'OMG (et non une variante)

Un seul UIM par technologie (2/2)

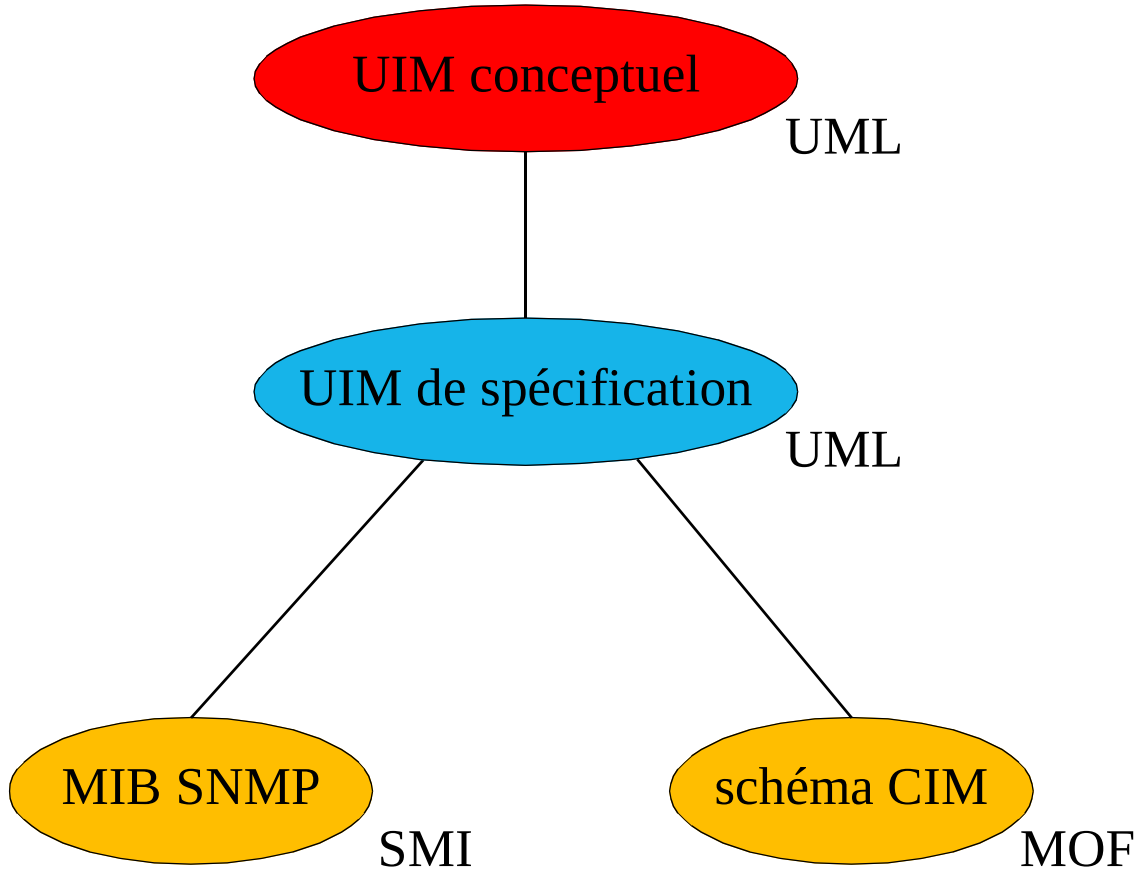
- Durable :
  - terminologie stable
  - nul besoin de re-former les gens
- Réutilisable :
  - partagé par l'IETF, le DMTF…
- Défini par un seul WG incluant :
  - des experts dans la technologie en question
  - des modélisateurs
  - des chercheurs
  - des consultants indépendants
  - des utilisateurs

Plusieurs modèles de données par technologie

- Plusieurs modèles de données peuvent être dérivés d'un même UIM :
  - MIB SNMP
  - schéma CIM
  - schéma de répertoire LDAP
  - etc.
- Pas nécessairement orienté objet
- Le langage pour définir ces modèles n'est pas imposé
- Définis par différents WGs (IETF, DMTF...) incluant :
  - des fournisseurs d'applications de gestion
  - des consultants indépendants
  - des utilisateurs

Trois couches

Universal Information Models



modèles de données

Processus itératif de conception

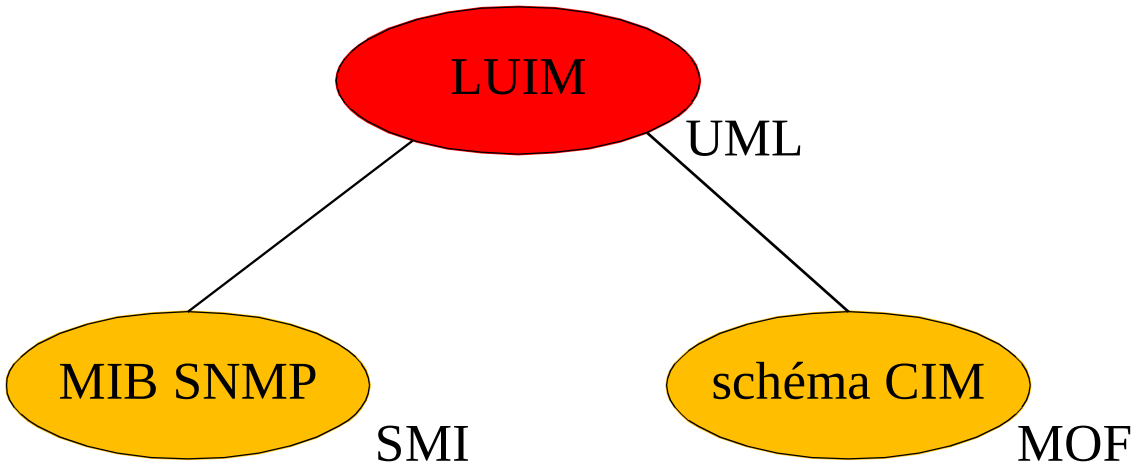
Pourquoi a-t-on besoin de plusieurs itérations ?

- Modèle multi-couche + 1 itération = temps de normalisation nécessairement long
  - retarde le temps de mise sur marché d'une nouvelle technologie
  - équipementiers = refus catégorique !
- Quelle que soit l'expérience des modélisateurs, ils se trompent toujours lorsqu'ils modélisent une technologie complexe pour la première fois
- Le cahier des charges change parfois au cours du temps

Itération n°1 : prototypage

Lightweight Universal Information Model

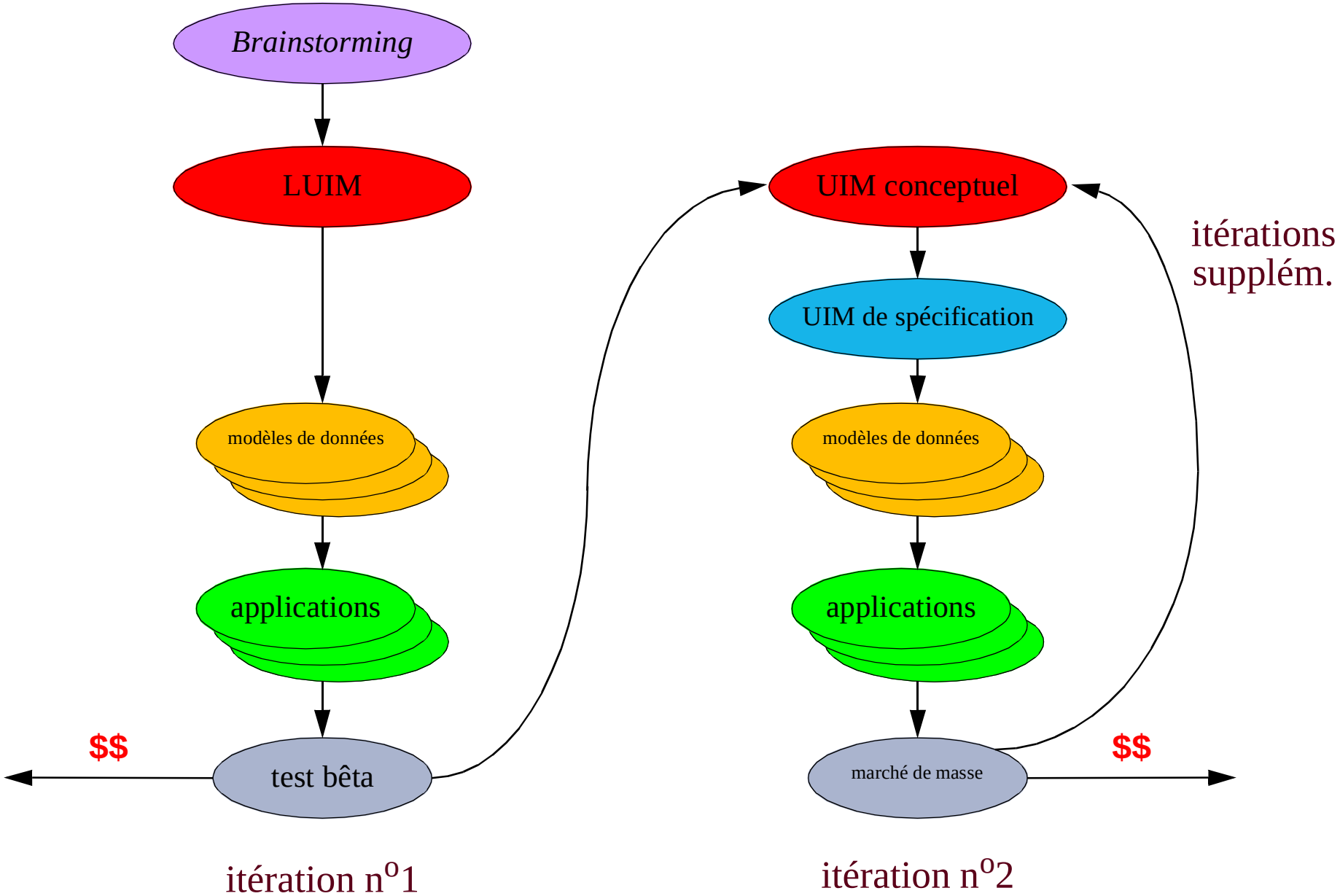
modèles de données



Itération n°2 : affinage

- Formaliser l'UIM :
  - diagrammes de classes UML, diagrammes de séquences UML…
  - livre blanc
- Améliorer l'UIM :
  - prêt pour le marché de masse
- Rendre l'UIM robuste et durable
- Tirer les leçons des erreurs commises durant l'itér. n°1 :
  - *feedback* des bêta-testeurs
- Formaliser par écrit ces leçons :
  - ex. : sous forme d'annotations dans un livre blanc
  - but: les mêmes problèmes ne resurgissent pas quelques années plus tard

Processus itératif et incrémental



Itérations supplémentaires : 2 causes

- Maintenance :
  - les problèmes de gestion ont changé au cours du temps pour cette technologie
- Affinage :
  - de graves problèmes (erreurs ou oublis) ont été découverts dans ce modèle d'info. de gestion

Gestion du temps : une condition nécessaire au succès de ce processus

- Il est nécessaire de gérer le temps de façon stricte :
  - fixer des dates limites pour chaque étape du processus de normalisation
  - le responsable de chaque WG doit veiller à ce que ces dates soient respectées
- Pourquoi les gens respecteraient-ils ces dates limites ?
  - compétition entre org. de normalisation
  - compétition entre experts mondiaux en modélisation d'info. de gestion
  - reconnaissance par les pairs

Avantages du nouveau processus

Qualité insuffisante de certains modèles d'informations de gestion : problème résolu

- On définit les modèles d'info. de gestion couche par couche, au lieu de se ruer sur les modèles de données très détaillés
- Avec la phase de prototypage, on acquiert de l'expérience sur le terrain (*feedback* des bêta-testeurs)
- Les UIMs rendent les efforts de normalisation bien plus attractifs pour des experts mondiaux en technologie ou en modélisation

L'antipatterne "réinventer la roue" : problème résolu

- Pour une technologie donnée, tous les modèles de données sont dérivés d'un seul et unique UIM
- On bâtit sur l'expérience passée au lieu de tout recommencer à zéro :
  - réutilisation
- La terminologie reste stable au fil du temps :
  - ceci évite la confusion terminologique chez les clients

Trouver le bon niveau d'abstraction entre deux extrêmes : problème résolu

- Avec des modèles multi-couches, les modélisateurs peuvent saisir différentes choses à différents niveaux d'abstraction :
  - UIM : les concepts de base
  - modèles de données : les détails
- Quand les problèmes de gestion d'une technologie donnée s'avèrent complexes, on peut définir autant de couches que nécessaire

Phase d'apprentissage trop difficile : problème résolu

- Les modèles conceptuels facilitent la phase d'apprentissage pour les débutants ou nouveaux venus
- Les modèles conceptuels exprimés en UML (*lingua franca*) peuvent être facilement compris par des gens qui connaissent bien la gestion de réseaux, de systèmes ou de services, mais ne sont pas pour autant spécialistes de SMI ou de MOF

Avantages supplémentaires

- Même si la technologie change après la définition du LUIM, il est encore possible de mettre à jour l'UIM durant la 2ème itération (i.e., avant le déploiement à grande échelle)
- Les fournisseurs réalisent de belles économies du fait que les UIMs soient partagés par l'IETF, le DMTF, etc. :
  - plus besoin de payer des gens à travailler en parallèle dans différents WGs
  - les coûts de développement des MIBs SNMP et des schémas CIM sont réduits par cette factorisation du travail
- En imposant une stricte gestion du temps, on fixe une borne sup. au temps de mise sur marché lors de la 1ère itération (celle qui compte pour l'effet d'annonce) :
  - très important pour les départements *marketing*

Que faire s'il y a plusieurs UIMs pour une même technologie ?

- Ceci peut arriver quand :
  - différents membres d'un WG ont des vues différentes sur la façon dont une technologie devrait être gérée
  - différents WGs définissent simultanément des UIMs pour une même technologie
- Problèmes :
  - source de confusion terminologique
  - segmente le marché
- Solution :
  - à la mode IETF : que le marché décide !
  - les utilisateurs peuvent comparer les mérites respectifs de ces UIMs car ils sont tous exprimés dans la même *lingua franca :* UML

Résultats préliminaires

- Définition d'un UIM pour routeurs IP
  - cf. article NOMS 2002
- Combine les approches ascendantes et descendantes (*bottom-up* et *top-down)*
- *Reverse-engineering* de la RFC 1213 (achevé)
- *Reverse-engineering* de deux schémas CIM (en cours) :
  - schéma *System*
  - schéma *Network*
- Mise en évidence de ce qui manque dans les MIBs SNMP et les schémas CIM existants

Conclusion

Résumé (1/2)

- Quatre problèmes relatifs à la modélisation des info. de gestion ont été décrits :
  - qualité insuffisante de certains modèles
  - l'antipatterne "réinventer la roue"
  - trouver le bon niveau d'abstraction entre deux extrêmes
  - phase d'apprentissage trop difficile
- Un nouveau processus de modélisation et de normalisation a été proposé afin d'atténuer ou de résoudre ces problèmes :
  - modèles multi-couches
  - processus itératif de conception

Résumé (2/2)

- La coopération entre organismes de normalisation a été défendue :
  - partage des UIMs conceptuels
  - plus de confusion terminologique
- Les avantages de la multi-spécialisation ont été soulignés :
  - *UIMs* : domaine des concepteurs et experts en technologie
  - *modèles de données* : domaine des développeurs et spécialistes en SMI (SNMP), MOF (WBEM/CIM), etc.

Axes de recherche

- Définition d'UIMs conceptuels :
  - *reverse-engineering* de MIBs SNMP existantes
  - *reverse-engineering* de schémas CIM existants
- Pour une technologie donnée, le fait que plusieurs modèles de données soient dérivés d'un seul UIM facilite-t-il la traduction entre ces modèles de données ?
- Les UIMs nécessitent-ils la définition d'un équivalent du *Core Model* du DMTF ?