

Iterative, Multi-Tier Management Information Modeling

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

Talk at ETH Zürich
February 18, 2002

*Work in collaboration with Divesh Srivastava, AT&T Labs Research
and Andrea Westerinen, Cisco Systems*

Outline

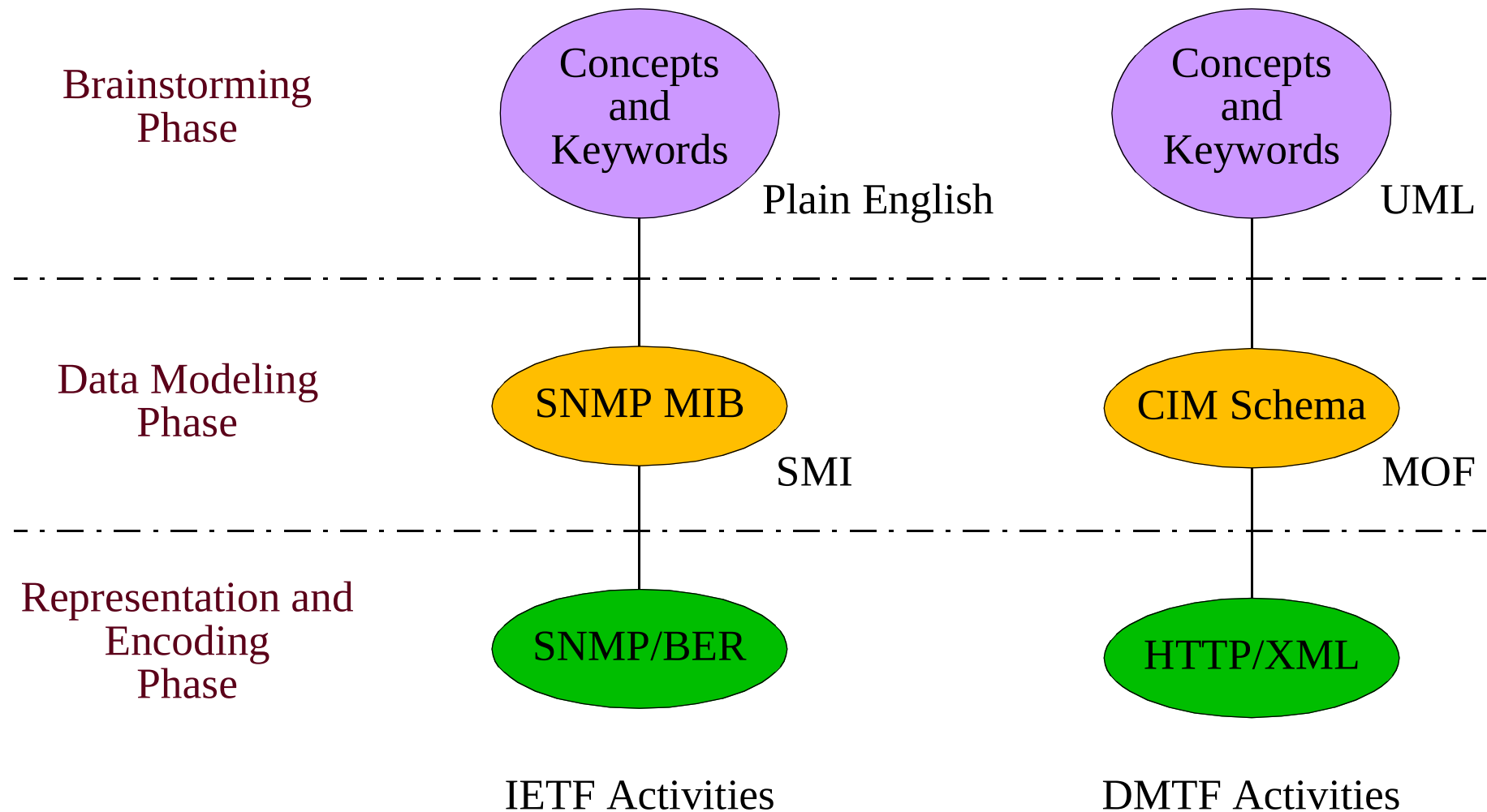
- mgmt. info. modeling in the IP world
- Four problems
- Analysis
- Multi-tier models
- Iterative process
- Advantages of our new modeling process
- Conclusion

Management Information Modeling in the IP World

Technology-Independent Standardization Activities

- Metamodel:
 - DMTF: variant of UML metamodel
 - class, object, association, etc.
 - IETF: implicit metamodel
 - everything in a MIB is an OID
- Language:
 - SNMP MIBs: SMI
 - SNMP PIBs (policies): e.g., SPPI
 - CIM Schemas: MOF
- Representation and encoding of mgmt. data:
 - IETF: BER
 - DMTF: XML, CIM Operations over HTTP

Per-Technology Standardization Activities



Four Problems

Some Models Are Not Good Enough

- Some models contain errors:
 - e.g., RFC 1156 immediately replaced with RFC 1213 because of the Address Translation Group
- Some models miss important features:
 - e.g., no per-interface ACLs in RFC 1213
 - must use telnet

Why?

- WGs are mostly driven by vendors:
 - poor trade-off between quality and timeliness
 - fast design is not beautiful
- Management standardization efforts often fail to attract:
 - the best technology experts
 - the best information modelers
- Fuzzy requirements:
 - e.g., what dials and knobs do we need to manage MPLS-based VPNs?

The *Reinvent the Wheel* Antipattern

- Many standards bodies in the enterprise mgmt. arena:
 - IETF, DMTF, OMG, TMF, ISO, ITU-T, TOG, etc.
- Little cross-pollination between them:
 - *not invented here* syndrome
 - no time to read the literature -> start from scratch
- Consequences:
 - Terminology keeps changing:
 - e.g., DMTF: event, notification, indication
 - customers are confused
 - Standards bodies waste precious time

Finding the Right Level of Abstraction Between Two Extremes

- Overly abstract models:
 - e.g., OMG's four-tier metamodel architecture
 - usefulness in practice?
 - devised by theoreticians
 - over-engineering antipattern
- Overly detailed models:
 - e.g., many IETF's SNMP MIBs
 - bottom line blurred by details
 - devised by application developers
 - under-engineering antipattern

The Learning Curve Is Too Steep

- Newcomers are swamped by details:
 - must read SMI fluently to understand SNMP MIBs
 - must read MOF fluently to understand CIM schemas
- Newcomers need a better way to understand first the bottom line, and then the details

Analysis

Four Solutions from Software Engineering (1/2)

- With one-tier MIMs, we try to do too many things at a time, and require too many skills from the same people:
 - Split between conceptual, specification, and implementation models (analysis, design, and implementation phases).
- Going from one mgmt. architecture to another does not make the mgmt. issues any different for a given technology:
 - Isolate the architecture-independent core from the rest:
 - facilitate reuse
 - more elegant design
 - decrease the risks of terminological changes and confusion

Four Solutions from Software Engineering (2/2)

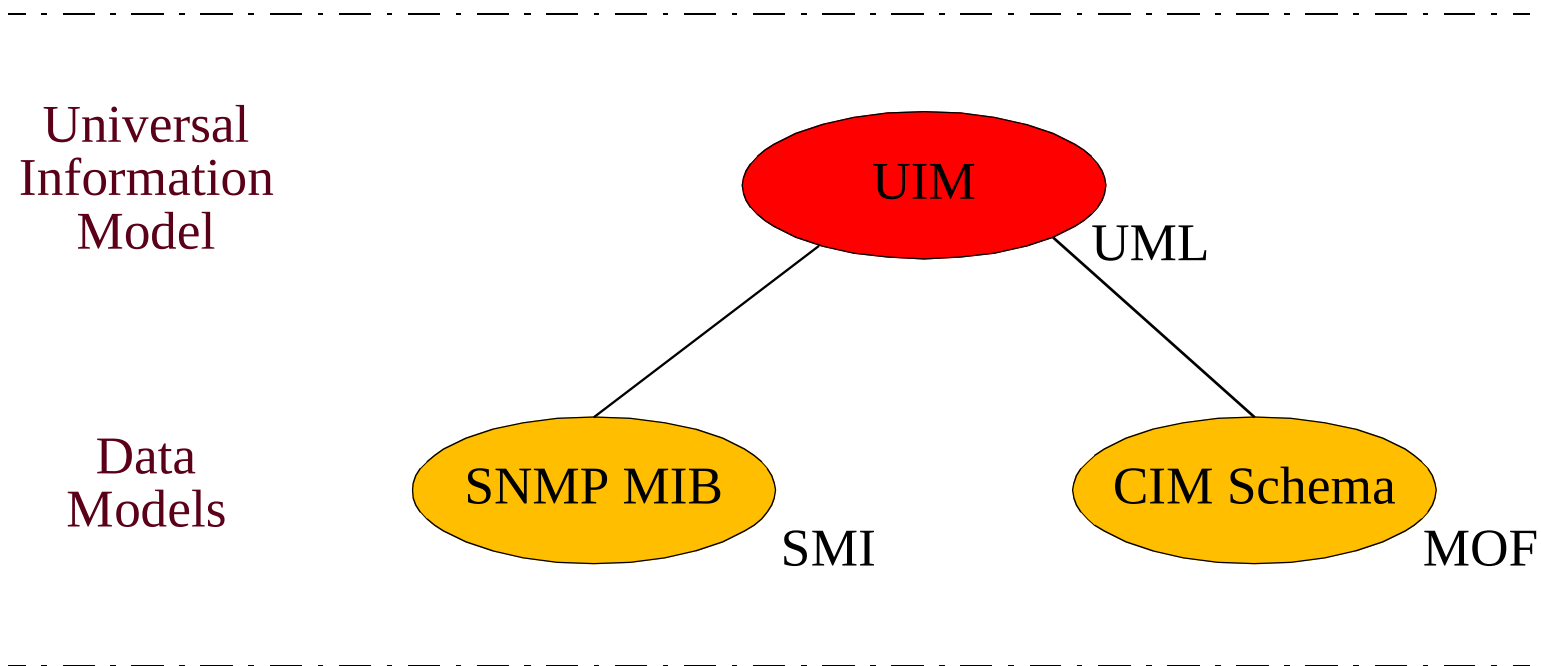
- Software quality is best assured by attracting the best people to fulfill each task throughout the software development process:
 - We need to attract the best technology experts and the best information modelers in standards bodies
- The waterfall process works only in simple cases:
 - As management issues become more complex, we need to migrate to an iterative and incremental modeling process.

Constraints from Real Life

- In the IP world, mgmt. systems are much more expensive today than in the mid-1990s. In consequence, many customers now demand standards (“insurance policy”).
- Any new modeling process must allow vendors to release new technologies fast. Their market is *very* competitive.
- Redeploying a MIM is extremely expensive to customers and vendors. To avoid it, every effort should be made to devise good models in the first place.
 - addresses poor models, not changing requirements
- Many customers demand high-quality management applications as soon as they buy a new piece of equipment. Large NOCs cannot afford to deploy now and manage later.

Multi-Tier Models

Example: Two Tiers



One UIM per Technology (1/2)

- UIM = object-oriented abstract model
- Expressed in UML + whitepapers
- Goal = convey the big picture to humans, not machines or compilers:
 - ignore details
- Independent of mgmt. architecture:
 - indep. of data repository
 - indep. of communication protocol
 - communication and information models are indep.
- Uses OMG's UML metamodel

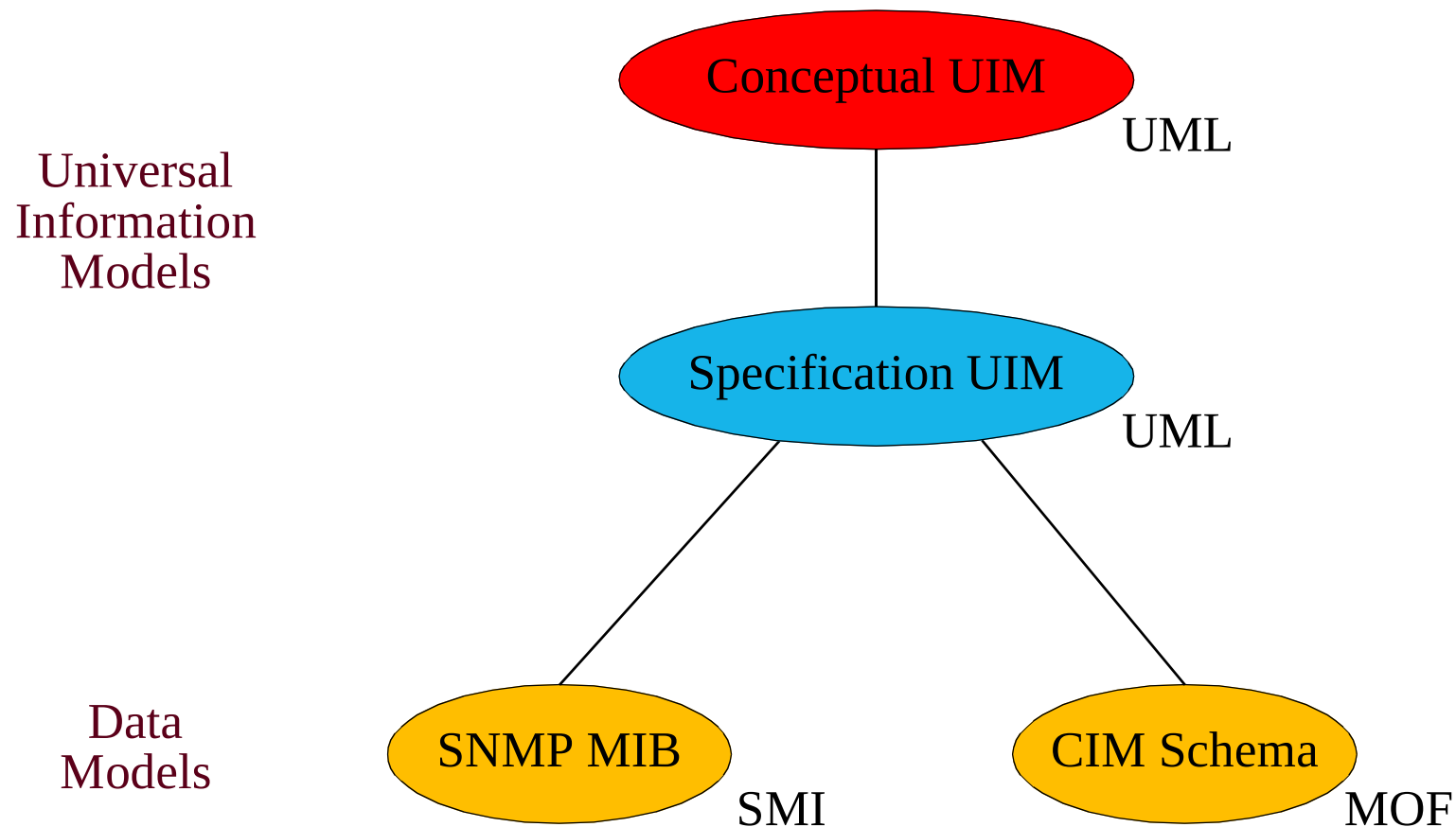
One UIM per Technology (2/2)

- Durable:
 - stable terminology
 - no need to retrain people
- Reusable:
 - shared by IETF, DMTF, etc.
- Devised by joint IETF/ DMTF WGs including:
 - some of the best technology experts in the world
 - some of the best mgmt. info. modelers in the world
 - researchers
 - independent consultants
 - end-users

Multiple Data Models per Technology

- Several data models are derived from a single UIM:
 - SNMP MIB
 - CIM schema
 - LDAP directory schema
 - etc.
- Not necessarily object oriented
- Language for devising data model: not prescribed
- Data models defined by separate WGs including:
 - vendors developing mgmt. applications
 - independent consultants
 - end-users

More than Two Tiers



Iterative Process

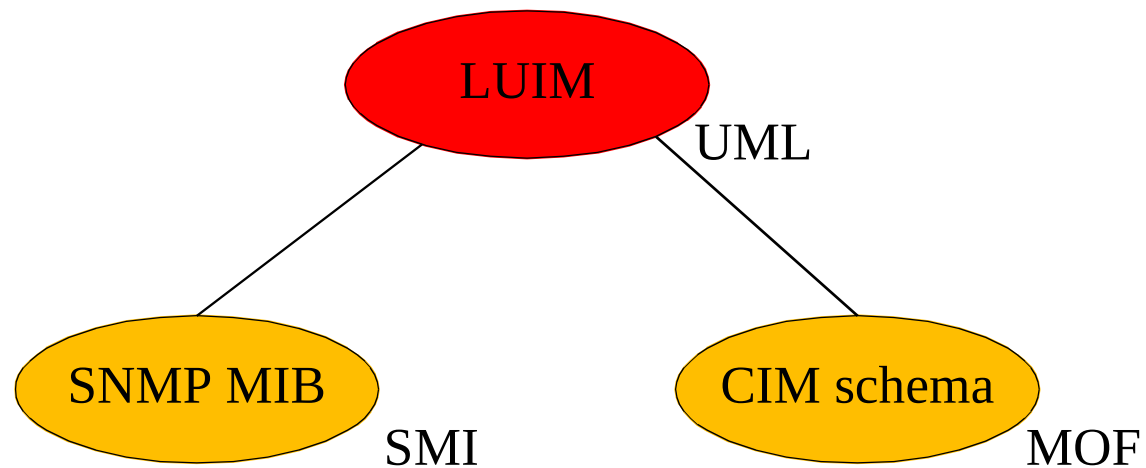
Why Do We Need Multiple Iterations?

- N-tier models + 1 iteration = long standardization time
 - delays time-to-market for new technologies
 - vendors = no-no
- Whatever the experience of model designers, they will always get it wrong the first time they model a complex technology!!!
- Requirements sometimes change over time

Iteration 1: Prototyping

Lightweight
Universal
Information
Model

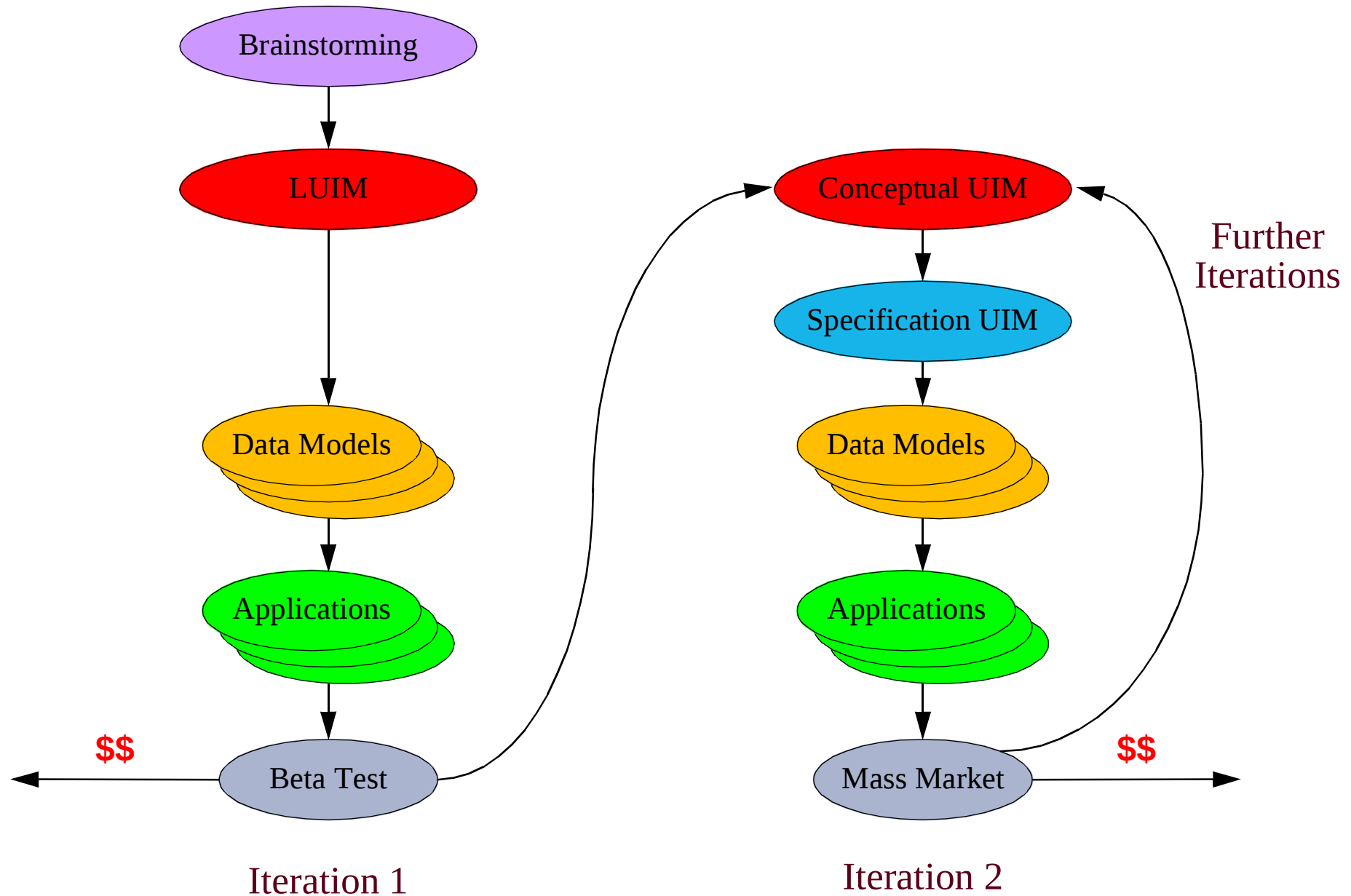
Data
Models



Iteration 2: Refinement

- Formalize the UIM:
 - UML class diagrams, sequence diagrams, etc.
 - whitepaper
- Improve the UIM:
 - ready for mass-market
- Make the UIM robust and durable
- Learn from the mistakes made in iteration 1:
 - feedback from beta-testers
- Formalize in writing the lessons learned:
 - e.g., annotations to the whitepaper
 - goal: the same problems will not resurface in the future

Iterative and Incremental Process



Further Iterations

- Maintenance:
 - mgmt. issues changed over time
- Refinement:
 - flaws were discovered in the info. model

Managing Time: A Condition for Success

- Must manage time strictly:
 - set deadlines for each step of the standardization process
 - chairperson of each WG must enforce deadlines
- Why would people bother to meet these deadlines?
 - competition between standards bodies:
 - DMTF wants to beat IETF in enterprise mgmt. (IP world)
 - TMF (telephony world) wants to have impact in IP world
 - OMG and TOG want to show the world that their architectures are useful, and enterprise mgmt. is a good application domain for them
 - competition between top-notch model designers
 - recognition by peers

Advantages of Our New Modeling Process

Some Models Are Not Good Enough: Solved

- We devise multi-tier MIMs, step by step, instead of jumping directly to data models
- With the prototyping phase, we learn from experience gathered in the field
- With UIMs, standardization efforts are a lot more attractive to the best worldwide technology experts and info. modelers

The *Reinvent the Wheel* Antipattern: Solved

- For a given technology, all data models are derived from a single UIM
- Build on past experience:
 - reuse
- Stable terminology:
 - avoid terminological confusion

Finding the Right Level of Abstraction Between Two Extremes: Solved

- With multi-tier models, we allow info. modelers to capture different things at different levels:
 - UIM: big picture
 - data models: details
- When the mgmt. issues for a given technology are complex, we can have as many tiers as necessary

The Learning Curve Is Too Steep: Solved

- Conceptual models make it easier for newcomers to get started with the mgmt. of a given technology
- Conceptual models expressed in UML (*lingua franca*) can be readily understood by people who do not know the idiosyncrasies of SNMP or WBEM

More Advantages

- If the technology changes during prototyping, once the LUIM is devised, we still have a chance to update the UIM in iteration 2 (i. e., before large-scale deployment)
- Having UIMs shared by the IETF and DMTF helps vendors cut their mgmt. software development costs when they support both SNMP MIBs and CIM schemas
- By imposing strict time mgmt., we put an upper bound on the time-to-market for iteration 1. This is important for marketing people.

How to Deal with Multiple Competing UIMs?

- Occurs when:
 - different people in a WG have conflicting views on the way a technology should be managed
 - different WGs come up with different UIMs, which are both consistent and smart
- Problems:
 - causes terminological confusion
 - segments the market
- Solution:
 - IETF's way: let the market decide
 - customers can compare UIMs: all expressed in the same *lingua franca* (UML)

Preliminary Results

- MIM for an IP router
- NOMS 2002 paper
- Combine bottom-up and top-down approaches
- Reverse-engineer RFC 1213 (done)
- Reverse-engineer CIM System and Network schemas (ongoing)
- Highlight what is missing in existing SNMP MIBs and CIM schemas

Conclusion

Summary (1/2)

- We described 4 problems pertaining to mgmt. info. modeling in the IP world:
 - some models are not good enough
 - *reinvent the wheel* antipattern
 - finding the right level of abstraction
 - learning curve is too steep
- We proposed a new modeling and standardization process to alleviate or solve these problems:
 - multi-tier models
 - iterative process

Summary (2/2)

- We advocated the cooperation between standards bodies (especially IETF and DMTF):
 - share conceptual UIMs
 - avoid terminological confusion
- We advocated multi-specialization:
 - *UIMs*: designers and technology experts
 - *data models*: developers and specialists of SMI (SNMP), MOF (WBEM/CIM), etc.

Directions for Future Work

- Define conceptual UIMs:
 - work with AT&T and Cisco
 - reverse-engineer existing SNMP MIBs
 - reverse-engineer existing CIM schemas
- For a given technology, does the fact that data models are derived from a single UIM facilitate the translation between these data models?
- Do UIMs require an equivalent to DMTF's Core Model?