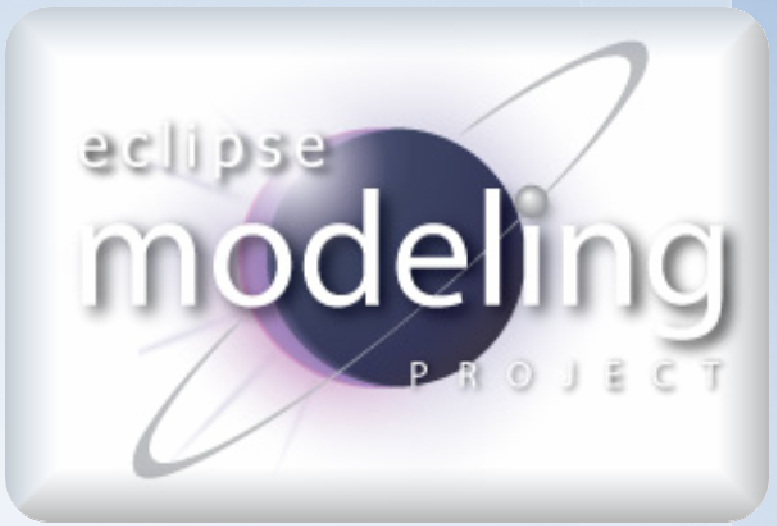


The Unbearable Stupidity of Modeling

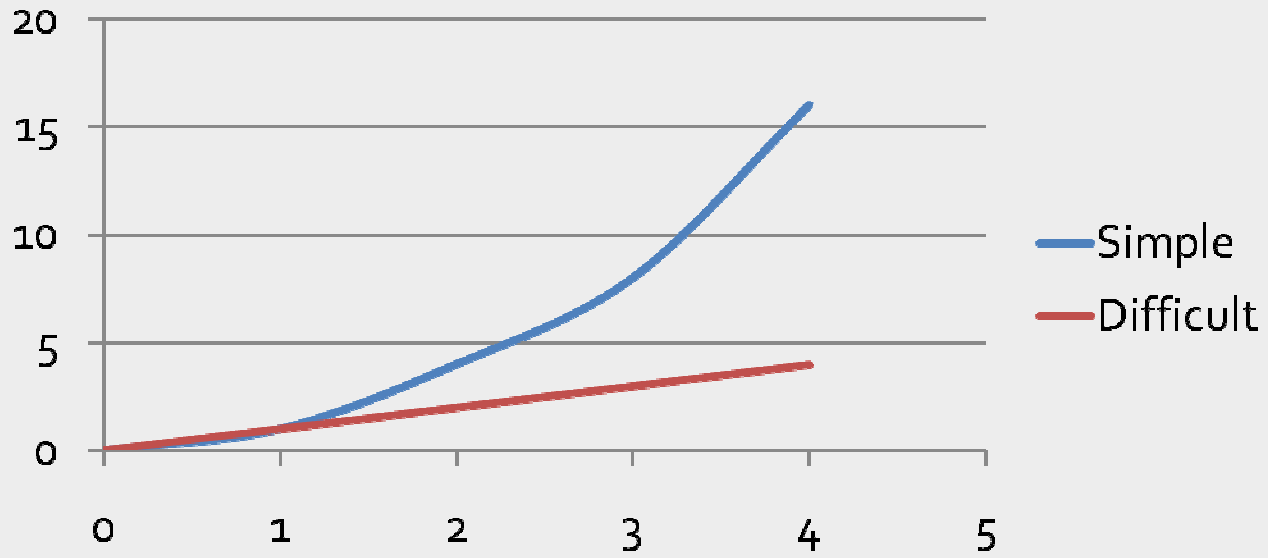
Ed Merks
Macro Modeling

What's all this Fuss I Hear about Modeling?

- Modeling isn't simply irrelevant, it's just plain stupid
  - After all, what's the point in anorexic young women strutting around modeling outrageous clothes?
- This is what we call a misconception
  - Modeling is about a lot more and a lot less than is commonly conceived
- We'll look carefully at the many reasons why modeling is seen as out of fashion

The Learning Curve is Too Steep

- Even the statement itself is a misconception!
  - Don't trust people who use it



Unified Modeling Language Is Complex

- UML is extremely complicated
  - Have you tried to read the specification?
  - You practically need a Ph.D. even to pretend to understand it
  - It's one of the most complex things I've ever seen
  - How can something like that help me do anything?

- Indeed UML is complex, powerful things usually are, but UML is not equal to Modeling

What is Modeling?

- Modeling is about abstraction which wikipedia defines as follows:

  Abstraction is the process or result of generalization by reducing the information content of a concept or an observable phenomenon, typically in order to retain only information which is relevant for a particular purpose.

- If modeling makes things more complex, you're doing something wrong

Models Drive Software Development

- Software is primarily focused on manipulating data
- That data has abstract structure
  - It can be described at a high level
  - It can be represented in different ways
  - It's always a model of something
- Whether it's recognized as such, models drive most software development

Meta What?

- When I hear mention of the word "metamodel," it turns me cold
- When I'm told about "metametamodels," I could just about cry
- These meta levels are like Dante's circles descending into hell
- So don't go there!

What is Meta Really?

- Think of it this way:
  - The description of the data is simply more data
    - It's quite commonly referred to as metadata
    - Meta is a bit confusing because all data is a model of something else and hence all data is meta
- The model of a model is a ~~meta~~model
  - Forget about the meta qualifier because whenever it's used, it generally will confuse rather than clarify
  - There are only models!

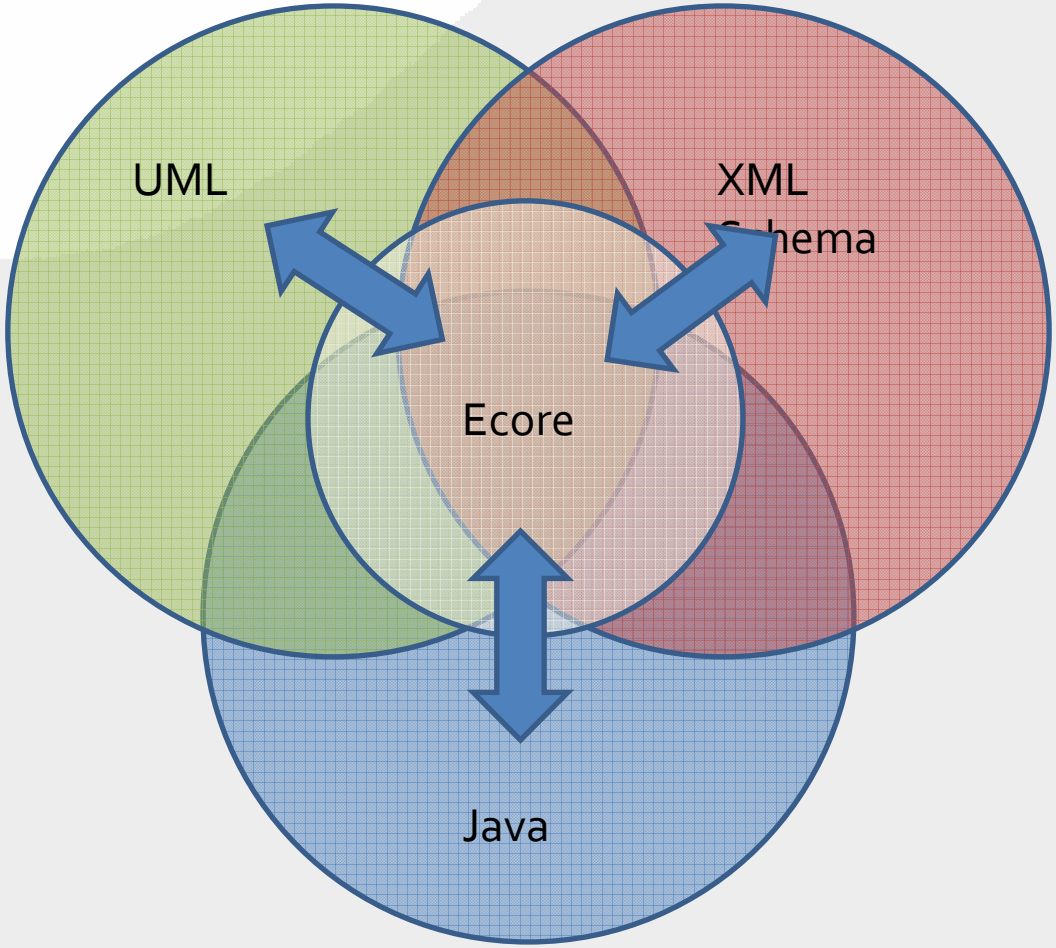
Ecore: The Model of Models

- EMF's Ecore is a simple model for describing models
  - Classification of objects
  - Attributes of those objects
  - Relationships/associations between those objects
  - Operations on those objects
  - Simple constraints on those objects, and their attributes and relationships
- Ecore is self describing, i.e., it is its own model
  - This prevents the descent into hell!

Models of Models all Look Alike

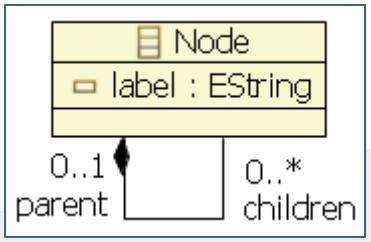
- Models higher up in the meta levels tend to all look the same for a very good reason
  - They begin to conform to our mental model of reality
- Think about the common abstract grammar that underlies human languages
  - The surface syntax differs wildly, but the fact that there are nouns, verbs, adjectives, adverbs and so on remains constant
  - Even if someone doesn't understand what the concept of "grammar" means, they still know how to communicate grammatically

Relationship of Ecore to Other Models

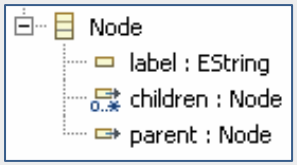


A Model is a Model is a Model

UML



Ecore



XML Schema

```
<xsd:complexType name="Node">
    <xsd:sequence>
      <xsd:element
        name="children"
        type="tree:Node"
        minOccurs="0"
        maxOccurs="unbounded"
        ecore:opposite="parent"/>
    </xsd:sequence>
    <xsd:attribute
      name="label"
      type="xsd:string"/>
</xsd:complexType>
```

Java

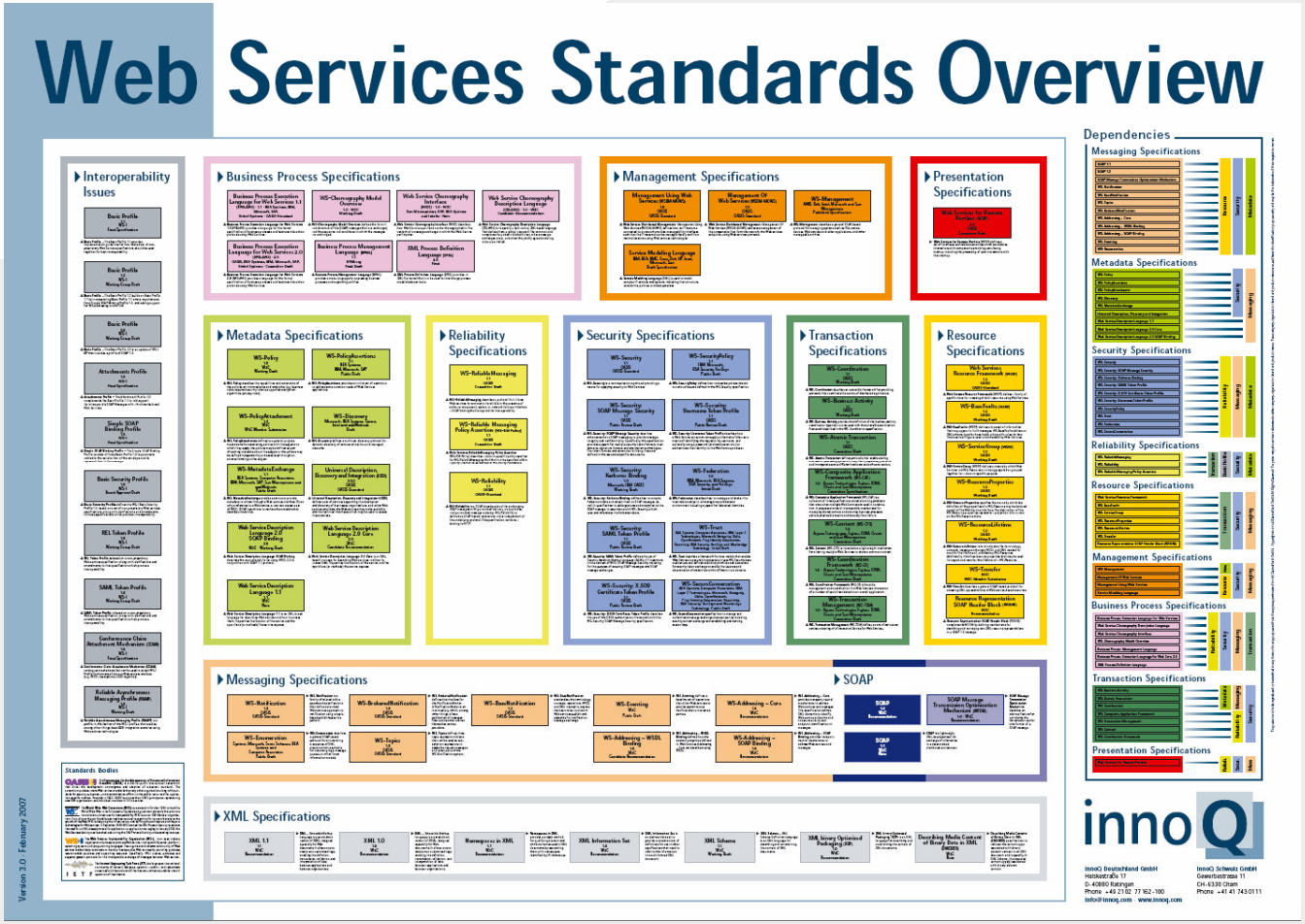
```
public interface Node {
  String getLabel();
  void setLabel(String value);
  List<Node> getChildren();
  Node getParent();
  void setParent(Node value);
} // Node
```

Modeling Will Only Get in the Way

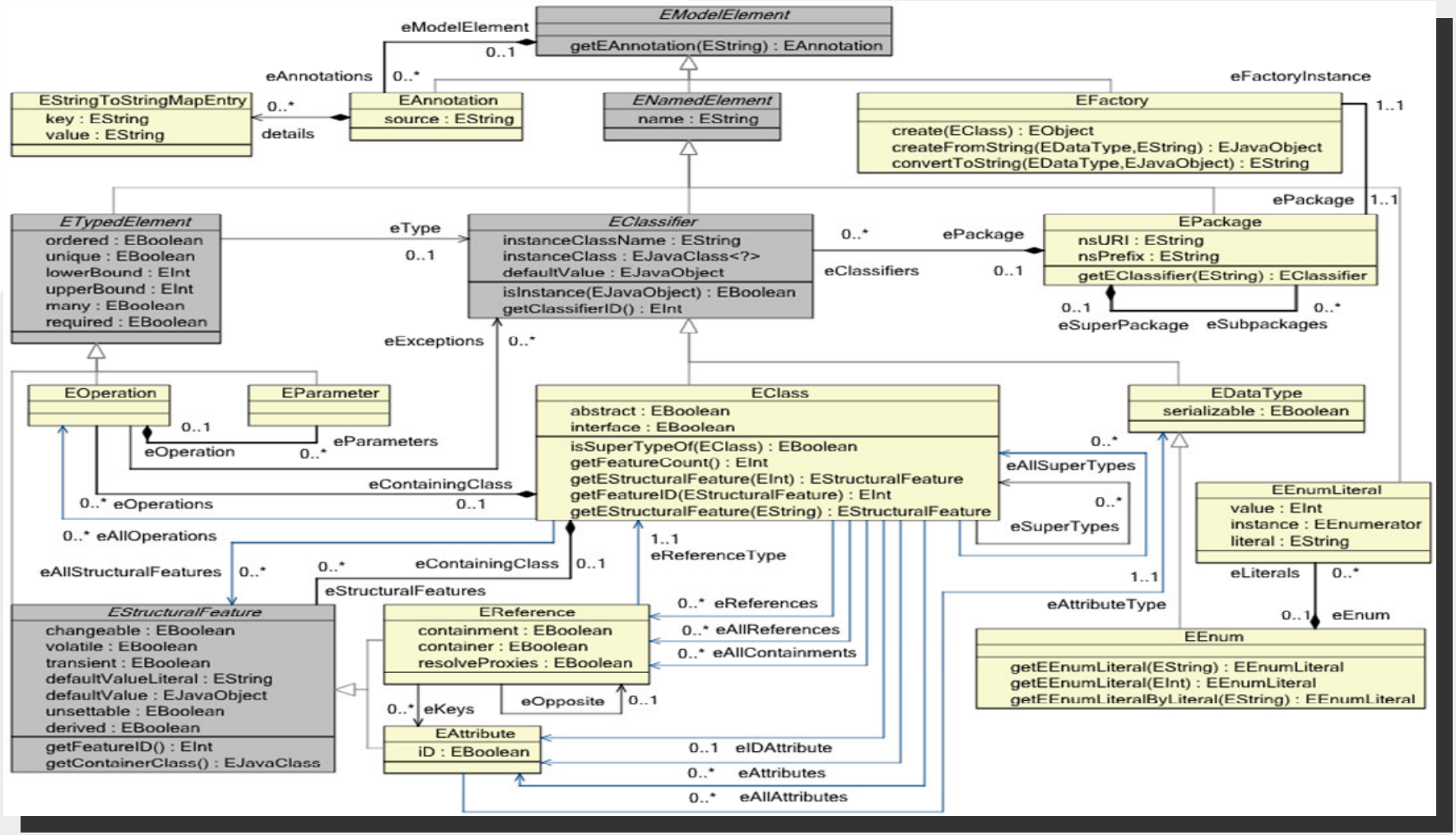
- Modeling's complexity will only distract me from solving the problem at hand
  - This will slow me down!

- Complexity, like beauty, is in the eye of the beholder
  - What we understand is simple
  - What we do not yet understand is complex

Are Web Services Simple?



Is Ecore Simple?



Is Java Simple?

- Have you ever read the specification?
  - http://java.sun.com/docs/books/jls/download/langspec-2.0.pdf
  - It isn't exactly light reading at 500+ pages
- Would you be able to describe what this java.util.Collections method signature means?

```
public static <T extends Comparable<? super T>> void sort(List<T> list)
```

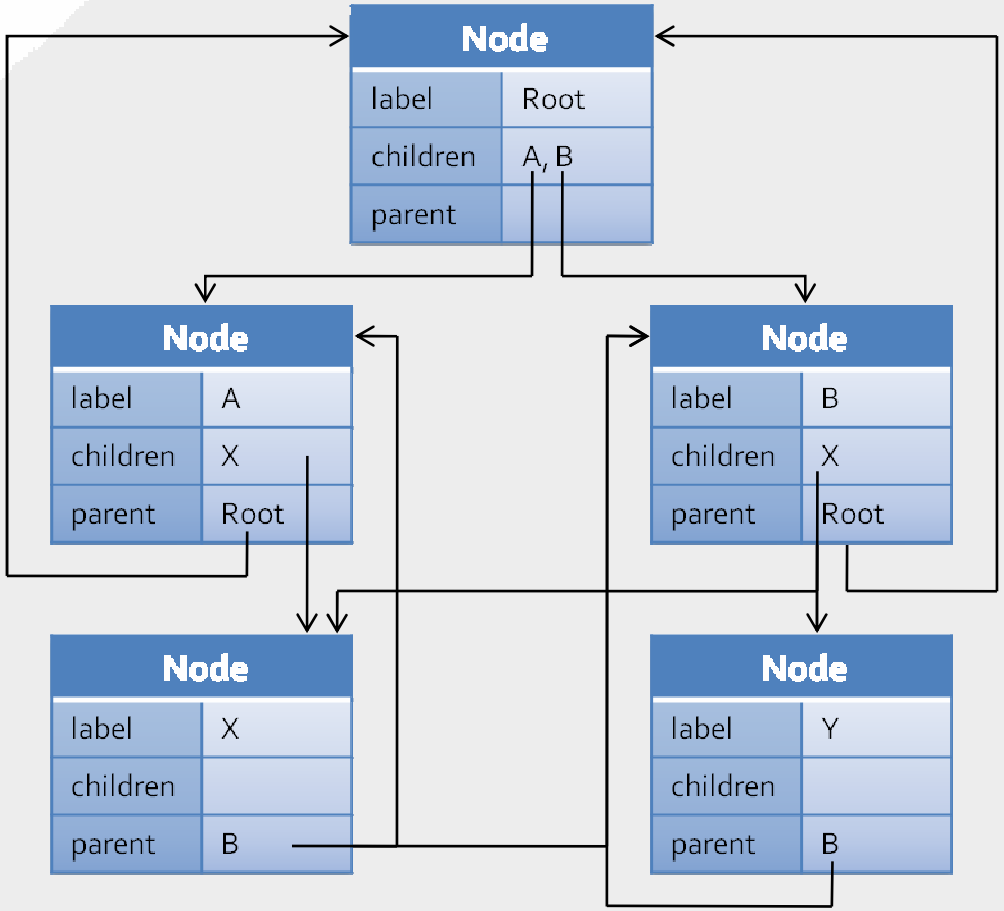
- Would you be surprised that the final assertion fails?

```
Map<?, ?> x = new HashMap<Object, Object>();
Map<?, ?> y = new HashMap<Object, Object>();
assert x.equals(y);
assert x.keySet().equals(y.keySet());
assert x.entrySet().equals(y.entrySet());
assert x.values().equals(y.values());
```

Is it Simpler to Do All This Manually

```
b.getChildren().set(0, x);
```

Notifier	Feature	Type	Old	New
b	children	SET	y	x
y	parent	SET	b	
x	parent	SET	a	b
a	children	REMOVE	x	



Modeling is Totally Redundant

- Java already has a reflective object model, so why would I need another one?
- Learning one general purpose programming language is more than enough of a challenge and should be more than sufficient.
  - They're Turing complete after all, so it's provably true!

- So use a Turing machine, and forget the complex alphabet, stick to 0's and 1's

Modeling Doesn't Replace Programming

- Definitely it's good to invest in learning a good programming language; use the right tool for the right job
- Java's reflective model is at a different level of abstraction than EMF's
  - Given a java.lang.Object, there's very little you can safely and meaningfully do with it
    - You can't even know if getX and setX represent a property; at best you could assume it.
  - Given an EObject, you can know at a high level its abstract properties and its relations to other objects

Modeling is Restrictive

- Modeling is simply too restrictive thereby limiting my creative abilities

- The only way to manage complexity is to try to bring order to the chaos
  - The ability to make simplifying assumptions about 100 different models is more important than having 100 different people being "creative" in their own unique way

Tedium is the True Killer of Creativity

- Given a class X with an attribute y of type string, imagine all the things one might need to write
  - An interface to represent the API for X
    - A method to get the value of y
    - A method to set the value of y
  - A class to implement that API for X
    - A variable to store the state for y
    - Code to initialize the default value
    - A method to get the value of y
    - A method to set the value of y
  - A factory interface for creating instances
    - A method to create an instance of X
  - An class to implement the factory interface
    - A method to create an instance of the class for X
- I fail to see the creativity in this!

Modeling Alone is Insufficient

- It's patently ridiculous to believe that modeling will be sufficient to generate my whole application without need for writing actual code
- Of course it is! Don't believe those who claim it
  - That being said, given a schema, EMF can generate a complete **crude** application for editing instances…
- I believe in learning to take advantage of modeling to do the mundane tasks so I can focus my creativity to add real value

Generated Code Sucks

- Generated code has problems
  - The quality is poor
  - The performance is bad
  - And it is difficult to understand and maintain
- I can do it much better myself by hand

- Oh really?!

Generated EMF Code

- Let's look at a generated EMF property accessor

```
public String getLabel()
{
   return label;
}
```

- How would you improve the quality, performance, readability, and maintainability of this?
- If anyone ever comes up with a better pattern, that solution is assimilated; resistance is futile; no good idea is safe

Performance Factoids

- Which do you think is faster?
  - `eObject.eGet(feature)`
  - `hashMap.get("key")`
  - Reflective look-up is twice as fast at a fraction of the size of look-up in a hash map
- Which do you think is faster?
  - treeNode.getChildren().contains(child)
  - arrayList.contains(x)
  - Reflective testing allows for O(1) compared to O(n)
- The lesson?
  - The information provided by a model can improve quality, performance, and even understandability

Diagrams Suck

- I don't like all those stupid diagrams
  - They just don't scale
  - A textual representation is far more manageable
  - I simply don't need a graphical rendering of my code
- So don’t use them
  - People usually don’t do a good job making nice readable diagrams anyway
  - Textual representations are indeed more manageable, but these aren’t mutually exclusive things
  - The diagram should be a rendering of the high level abstraction, not a graphical rendering of the code

DSLs Will Create a Tower of Babel

- Domain specific languages will create a Tower of Babel rife with formalisms that only the original developer understands

- May I remind you of that web services diagram again where each small box represents a very large specification?
  - The Tower of Babel is already upon us
- Providing specialized frameworks tailored to the abstract needs of domain specialists is the best and only way to make those specialists more productive

XMI Sucks

- XML Metadata Interchange is unspeakably horrible; I want nothing to do with it
- There's the word meta again!
- I can't see my data for all the UUIDs

- It does, so don't use it!
  - The use of the word "meta" is almost inexcusable since XMI is mostly just about interchanging plain old data
  - Other than the xmlns:xmi declaration, there's no good way to tell an XML serialization from an XMI one
  - XMI sucks because XML sucks
  - Like XML, UUIDs are loved way too much ☹

Modeling is a Marketing Ploy

- Modeling is just a ploy to ensure that I'll need to buy expensive tools
- Tool vendors tend to support their tools for only a few years before changing them all around as part of some new marketing campaign, thereby risking my long term investment

- Can you say open source?
  - Building your own tools is grossly expensive as well
  - Doing it all by hand is hardly better
  - Working with a community mitigates the risk

Modeling Enforces Onerous Processes

- I won't be able to do agile iterative development but rather will be stuck with an onerous formal waterfall process

- EMF's generator supports code merging
  - Code can be hand tailored
    - New things can simply be added
    - Generated things can be marked as modified to protect them from subsequent regeneration
  - The model can be changed any time too
    - Changes are merged in
    - Old stuff is swept away
- Follow whatever process turns your crank

My Project is Too Small to Need Modeling

- My project isn't big enough to need all that formal modeling overhead

- There are no small projects, only small minds
- Probably your project is too small to need Java too

Modeling Will Make Me Redundant

- If modeling really did work well, I might as well out-source my high tech job to the developing world

- Wake up quickly, the rest of the world isn't sitting idly by while you write endless getters and setters to make yourself feel productive and useful
- Did you ever wonder whether the management position should be the first to go?

Conclusion

- EMF is the defacto standard reference implementation of MOF
- EMF is a low cost modeling solution for Java
  - SD Times ranks it "top shelf" even relative to pricey commercial software
    - http://www.sdtimes.com/content/article.aspx?ArticleID=32287
- It exploits the models already underlying the application
- It supports iterative development that facilitates both model-based changes and hand-written changes equally well
- It boosts productivity by automating routine and mundane development tasks
- It's the foundation for data integration by providing a uniform way to access all models

Resources

- Online help
  - http://help.eclipse.org/ganymede/index.jsp?nav=/14
- Website
  - http://www.eclipse.org/emf
    - Downloads
    - Wiki
    - FAQ
    - Newsgroup
    - Documentation
- Books
  - Eclipse Modeling Framework
    - First Edition
      - http://safari.awprofessional.com/0131425420
    - Second Edition
      - http://my.safaribooksonline.com/9780321331885