object models: rationale

Daniel Jackson
Conceptual integrity is the most important consideration in system design. It is better to have a system omit certain anomalous features and improvements, but to reflect one set of design ideas, than to have one that contains many good but independent and uncoordinated ideas.

—1975

I am more convinced than ever. Conceptual integrity is central to product quality.

—1995
a radical suggestion

be explicit!
basic sketch → scope the problem → fill in details → clarify basic terms, expose confusion → rework model → express ideas, get rep independence → help achieve uniformity & generality → analyze → find subtle bugs
origins of our OM notation

- logic diagrams (Euler, Venn, Peirce)
- ZF set theory
- relational calculus (Tarski)
- relational model (Codd)
- Z notation
- ER & other data models
- model checking
- Alloy Language
- Alloy Diagrams
- object model notations (OMT etc)
- Unified Modeling Language

Historical timeline:
- 1700
- 1900
- 1940
- 1970
- 1980
- 1990
- 2000

Categories:
- mathematical logic
- object-oriented development
- software verification
- relational databases
**about alloy**

Alloy is a language for describing structures and a tool for exploring them. It has been used in a wide range of applications from finding holes in security mechanisms to designing telephone switching networks.

An Alloy model is a collection of constraints that describes (implicitly) a set of structures, for example: all the possible security configurations of a web application, or all the possible topologies of a switching network. Alloy’s tool, the **Alloy Analyzer**, is a solver that takes the constraints of a model and finds structures that satisfy them. It can be used both to explore the model by generating sample structures, and to check properties of the model by generating counterexamples. Structures are displayed graphically, and their appearance can be customized for the domain at hand.

At its core, the Alloy language is a simple but expressive logic based on the notion of relations, and was inspired by the Z specification language and Tarski’s relational calculus. Alloy’s syntax is designed to make it easy to build models incrementally, and was influenced by modeling languages (such as the object models of OMT and UML). Novel features of Alloy include a rich subtype facility for factoring out common features and a uniform and powerful syntax for navigation expressions.

The Alloy Analyzer works by reduction to SAT. Version 4 was a complete rewrite that included **Kodkod**, a new model finding engine that optimizes the reduction, and a new front end.
general points

why not UML?
much more complex notation & semantics
aiming for essence, not diagram of code

why not ER?
can’t express generalization
if it feels like grunt work...

then it is!

do it for real