object models:
why?

Daniel Jackson
structures everywhere

structures in a web app
› code: modules, namespaces
› i/o: requests, forms, responses
› state: documents, objects, cookies, sessions, databases

system structures
› network topologies, architectures, naming schemes

problem structures
› organizational roles, reporting hierarchies, access controls
› conceptual structures (what the system is about)
needed: a notation for structures

desiderata
› simple & lightweight
› precise & clear
› abstract (ie, widely applicable)

key challenge
› describe a set of structures, not just one
  all database states, accesses, network configurations
usually infinite!
a comparison: grammars

a grammar describes
› a set of strings
› usually infinite
examples
› all Java programs
› all web pages
› all TCP/IP packets
› all URLs
› all dialing sequences

strings such as:
http://csail.mit.edu/dnj

generated by a grammar:

```
uri ::= [protocol://] address [/ suffix]
address ::= (name .)* domain . tld

tld ::= com | edu | org
suffix ::= ...
```

## object models vs grammars

<table>
<thead>
<tr>
<th></th>
<th>object models</th>
<th>grammars</th>
</tr>
</thead>
<tbody>
<tr>
<td>define sets of graphs</td>
<td>graphs</td>
<td>strings</td>
</tr>
<tr>
<td>order built in?</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>containment?</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>sharing?</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>
sample instance
sample object model
origins of our OM notation

- Logic diagrams (Euler, Venn, Pierce)
- ZF set theory
- Relational calculus (Tarski)
- Relational model (Codd)
- Z notation
- Semantic data models
- Model checking
- Alloy Language
- Alloy Diagrams
- Object model notations (OMT etc)
- Unified Modeling Language

- Mathematical logic
- Object-oriented development
- Software verification
- Relational databases

Timeline:
- 1700
- 1800
- 1900
- 1940
- 1970
- 1980
- 1990
- 2000

- Model checking
- Alloy Language
- Alloy Diagrams
- Object model notations (OMT etc)
- Unified Modeling Language

- Mathematical logic
- Object-oriented development
- Software verification
- Relational databases
notes for the skeptic

why this syntax?
› more expressive than ER diagrams  
› simpler & smaller than UML

why any fixed syntax?
› communicating with others (incl. tools)

why write this stuff down?
› have to figure it out eventually  
› earlier surprises, earlier fixes  
› separation of concerns

why bother at all?
› conceptual integrity