The CLOS Metaobject Protocol
OOP:
What is an object?

“An object has state, behavior, and identity.”
(Grady Booch, 1991)
OOP: State

```plaintext
obj
```

```
  |
  |
  |
```
OOP: State

\[
\text{obj} \\
\downarrow \\
\begin{array}{c}
\text{(slot-value obj \texttt{`x})} \\
\text{(slot-value obj \texttt{`y})} \\
\text{(slot-value obj \texttt{`z})}
\end{array}
\]
OOP: State

```
obj

(slot-value obj 'x) ➔ (aref obj 0)
(slot-value obj 'y) ➔ (aref obj 1)
(slot-value obj 'z) ➔ (aref obj 2)
```
OOP: Identity

(eq obj1 obj2) => nil
OOP: Identity

(eq obj1 obj2) => nil
(setf obj2 obj1)
OOP: Identity

(eq obj1 obj2) => nil

(setf obj2 obj1)
OOP: Identity

(eq obj1 obj2) => nil
(setf obj2 obj1)
(eq obj1 obj2) => t
OOP:
How to map slots?

(defclass point ()
  (x y))

(defclass point-3d (point)
  (z))
OOP:
How to map slots?

1. compute class precedence list
2. compute slots
3. determine slot locations
OOP:
How to map slots?

1. (compute-class-precedence-list ...)
2. (compute-slots ...)
3. (slot-definition-location ...)
The Idea!

- Make compute-class-precedence-list, compute-slots, and so on, generic functions!
- Allow changes to the CLOS object model!
- Question: How to distinguish between standard and non-standard behavior?
Hierarchy for metaobject classes

t
standard-object

class slot-definition generic-function method
Hierarchy for metaobject classes

standard-class

standard-object

class slot-definition generic-function method
Hierarchy for metaobject classes

- t
- standard-object
- class slot-definition generic-function method
- standard-class standard-generic-function
Hierarchy for metaobject classes

t
standard-object

class  slot-definition  generic-function  method

standard-class   standard-generic-function  ...

Class

metaobject classes

• (defclass persistent-class (standard-class)
  ((database-connection ...)))

• (defclass person ()
  ((name ...)
   (address ...))
  (:metaclass persistent-class))
The Instance Structure Protocol

- (defmethod person-name ((object person))
  (slot-value object ‘name))

- (defun slot-value (object slot)
  (slot-value-using-class
   (class-of object) object slot))

- (defmethod slot-value-using-class
  ((class standard-class) object slot)
  (aref ...))
The Instance Structure Protocol

- (defmethod slot-value-using-class
  ((class persistent-class) object slot)
  (fetch-slot-from-database ...))
Other Protocols

• Initialization protocols
• Class finalization protocol
• Instance structure protocol
• Funcallable instances
• Generic function invocation protocol
• Dependent maintenance protocol
Example: The Python object model

1. Define a mix-in for hashtable-based slots.
2. Ensure that this mix-in is used.
3. Modify the slot access protocol.
Links
(Common Lisp)

- Andreas Paepcke, “User-level Language Crafting”
- [http://common-lisp.net/project/closer/](http://common-lisp.net/project/closer/)
Links
(Scheme)

• http://community.schemewiki.org/?object-systems
Links
(Smalltalk)

• http://www.laputan.org/#Reflection
Links
(C++)

- Ira Forman, Scott Danforth, “Putting Metaclasses to Work”
Links (Java)

- Ira Forman, Nate Forman, “Java Reflection in Action”