Lenguajes Minimales y Meta-Circulares

Hernán Wilkinson

Historia

- “[Smalltalk] became the exemplar of the new computing, in part, because we were actually trying for a qualitative shift in belief structures--a new Kuhnian paradigm in the same spirit as the invention of the printing press--and thus took highly extreme positions that almost forced these new styles to be invented” – Alan Kay

Software/Construcción de Software

- Software -> Modelo
- Construir software -> Modelar -> Aprender sobre lo que se modela -> Aprendizaje constructivista
- “millions of potential users meant that the user interface would have to become a ***learning environment along the lines of Montessori and Brunet***” – Alan Kay

El lenguaje no es nada…

- “and needs for large scope, reduction in complexity, and end-user literacy would require that ***data and control structures be done away*** within favor of a more biological scheme of protected universal ***cells interacting only through messages*** that could mimic any desired behavior” – Alan Kay
- “El lenguaje no es nada, los objetos son todo”
- Papers “Lambda: The Ultimate …”
- Smalltalk BNF: 2 páginas

El lenguaje no es nada...

- Ejemplos:
  - Algebra de Boole - ifTrue:
  - Iteraciones - whileTrue:
  - Excepciones - on:do: → redefinir handles:

Historia

- “The biggest hit for me while at SAIL in late '69 was to *really understand* LISP. Of course, every student knew about *car, cdr,* and *cons,* but … no one had penetrated the mysteries of *eval* and *apply.* I could hardly believe how beautiful and wonderful the *idea* of LISP was [McCarthy 1960]” – Alan Kay
- “… there were deep flaws in its logical foundations. By this, I mean that the pure language was supposed to be based on functions, but its most important components--such as lambda expressions, quotes, and conds--were not functions at all, and instead were called special forms” – Alan Kay

Historia

- “The actual beauty of LISP came more from the *promise* of its metastructures than its actual model. I spent a fair amount of time thinking about how objects could be characterized as universal computers without having to have any exceptions in the central metaphor. What seemed to be needed was complete control over what was passed in a message send; in particular, *when* and in *what environment* did expressions get evaluated?” – Alan Kay

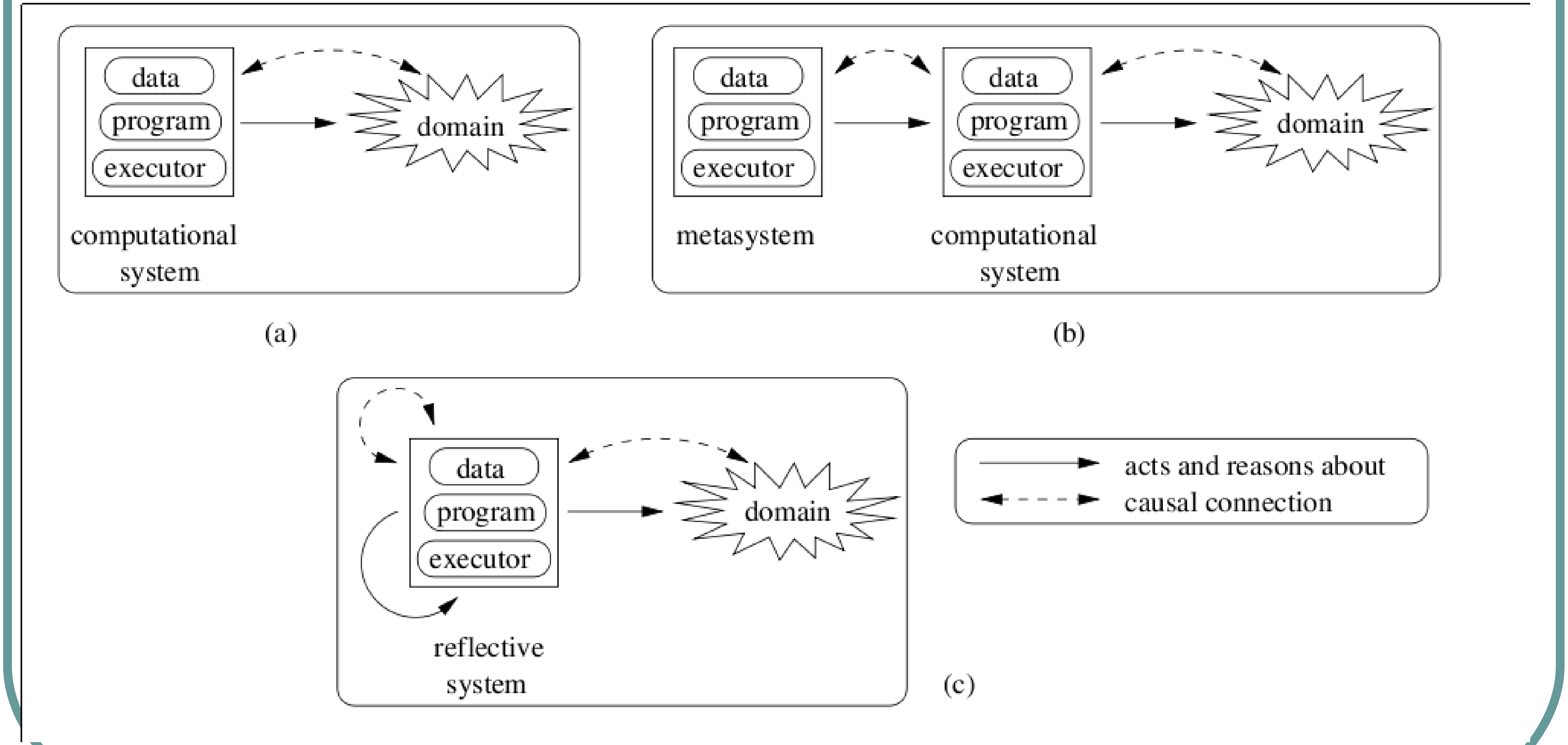
Metaprogramación y Reflexión

- Meta-xxx: Que habla sobre, que define a, xxx
- Ejemplo:
  - Una clase es un Meta-objeto porque define su comportamiento
  - El español es un “meta-lenguaje” porque puede predicar sobre “si mismo”. Ejemplo: “La palabra *casa* tiene 4 letras”

Metaprogramación y Reflexión

- Sistema Computacional:
  - Sistema que actúa y razona sobre un dominio
- Casual connection:
  - Propiedad que asegura que cambios en el dominio se ven reflejados en el modelo y viceversa
- Meta-sistema
  - Sistema cuyo dominio es otro sistema
- Sistema Reflexivo:
  - Meta-sistema “casually connected” consigo mismo

Metaprogramación y Reflexión



Metaprogramación y Reflexión

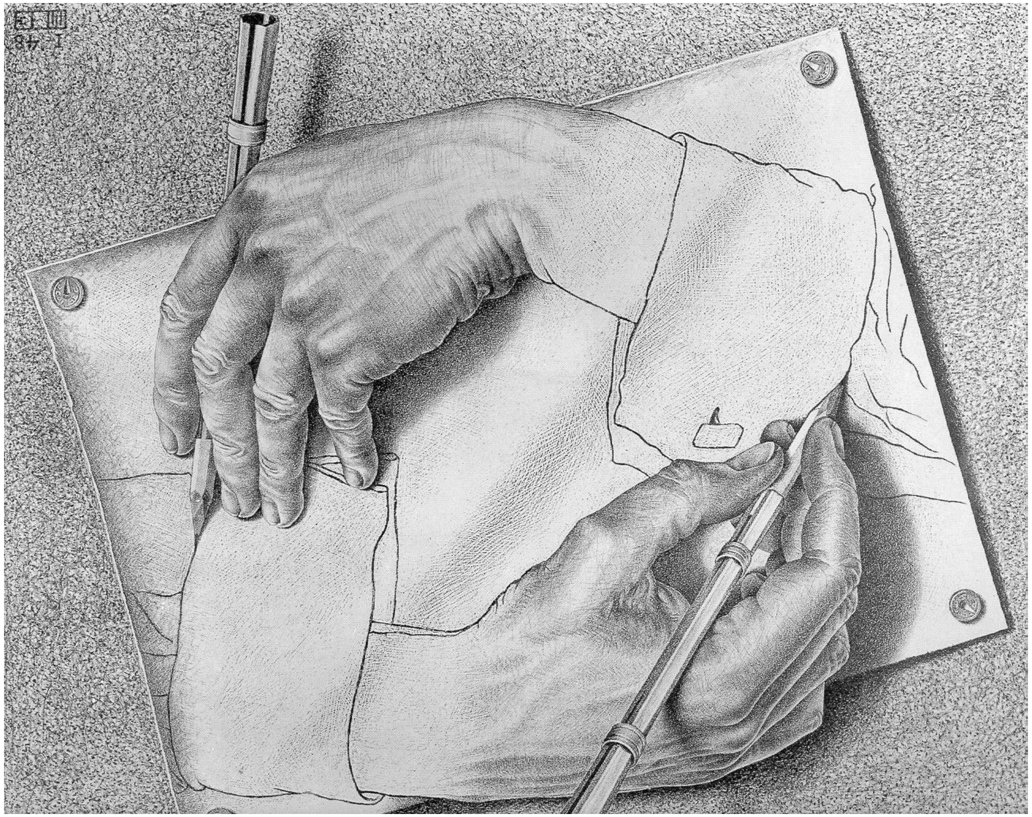
- Reflexión:
  - Habilidad integral de una entidad para representar, operar sobre y tratar consigo mismo en la misma manera que representa, opera sobre y trata con su sujeto primario
- Introspection:
  - La habilidad de un programa de razonar acerca de si mismo y/o la implementación del lenguaje de programación (read)
- Intercession:
  - La habilidad de un programa de “actuar” sobre la reificaciones de si mismo y la implementación del lenguaje de programación (write)

Metaprogramación y Reflexión

- Reflexión Estructural:
  - La habilidad de un programa de acceder a su representación estructural y la implementación del lenguaje de programación
- Reflexión de Comportamiento: (Behavioral Reflection)
  - Habilidad de un programa de acceder a la representación dinámica de si mismo, esto es a la ejecución operacional del programa y de la implementación del lenguaje de programación

Metaprogramación y Reflexión

- Lenguajes meta-circulares



Ejemplos

	Read	Write
Structure	All classes Does implement? Design rules?	addInstVarNamed: compile:
Behavior	Assertion Name Debugger	Create Method Pluggable Proxy Debugger

Historia

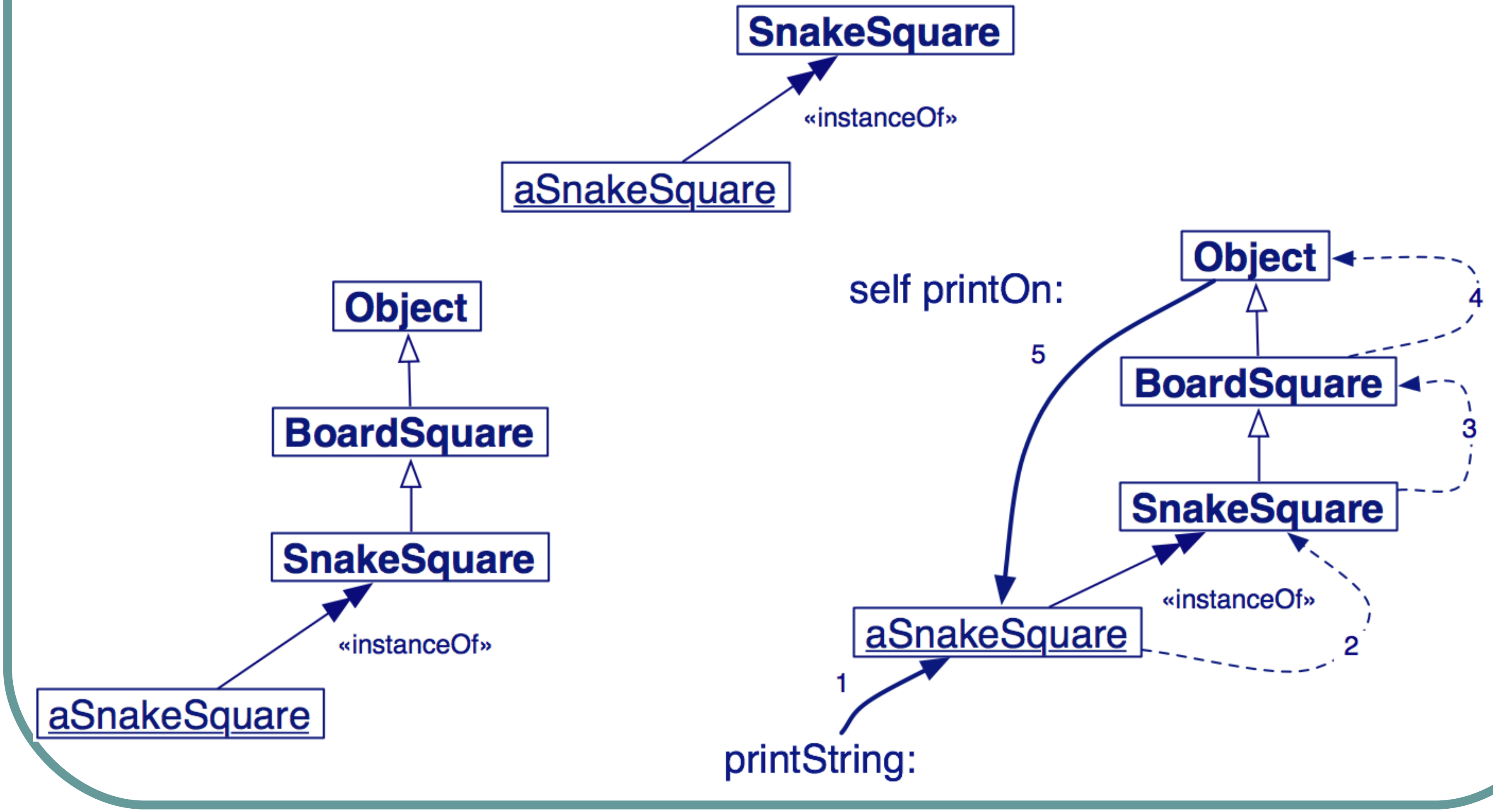
- “Philosophically, Smalltalk's objects have much in common with the monads of Leibniz and the notions of 20th century physics and biology. Its way of making objects is quite Platonic in that some of them act as idealizations *of concepts--Ideas--from* which *manifestations* can be created” – Alan Kay

Metaprogramación y Reflexión

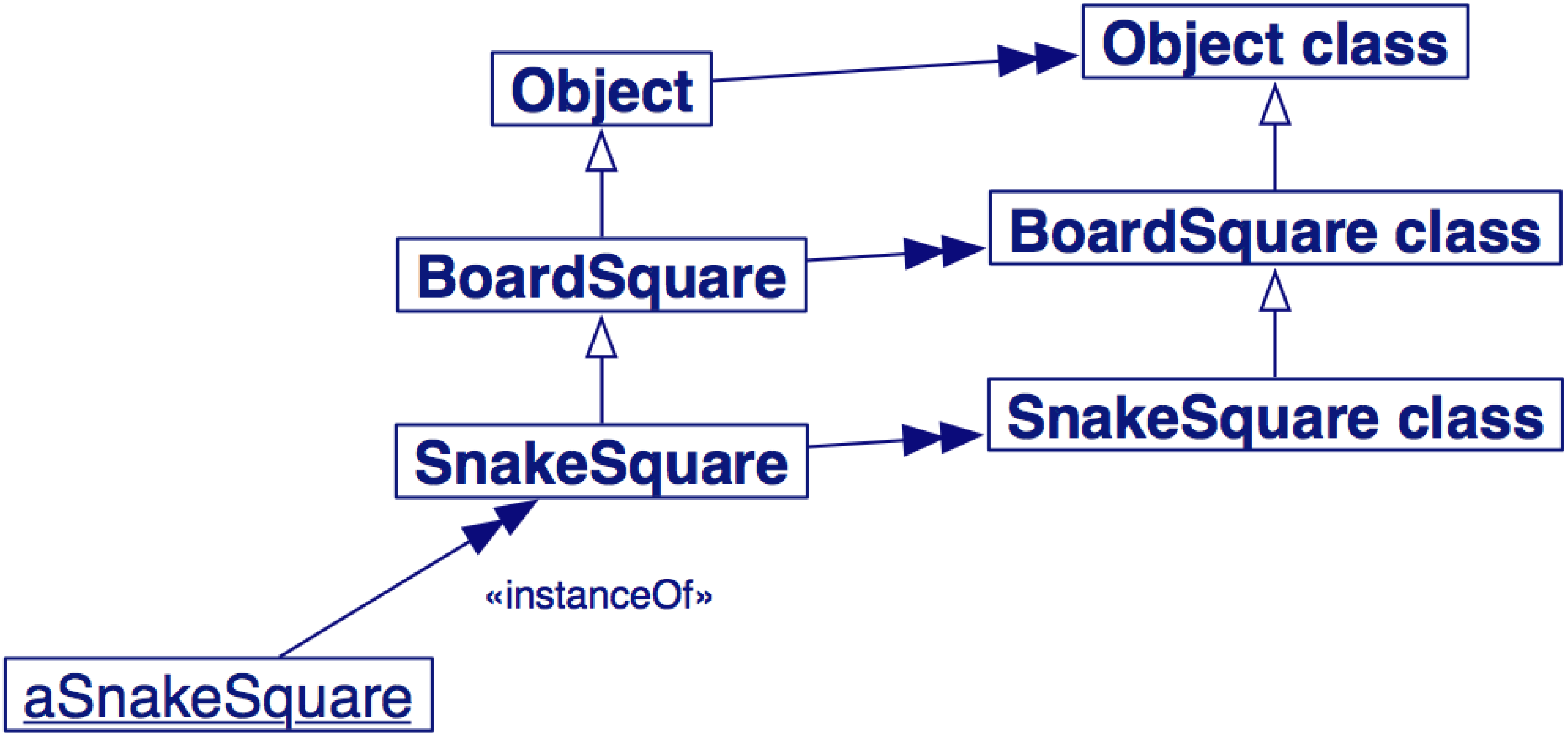
- Algoritmo de Method Lookup:
  - Sea r el receptor y m el mensaje

1) class = r.getClass()
2) method = Resultado de buscar m en method dictionary de class
3) Si lo encuentra ejecutar method. Salir
4) class = class.superclass ()
5) class no es nil ir a 2.
6) Evaluar r.doesNotUnderstand (m)

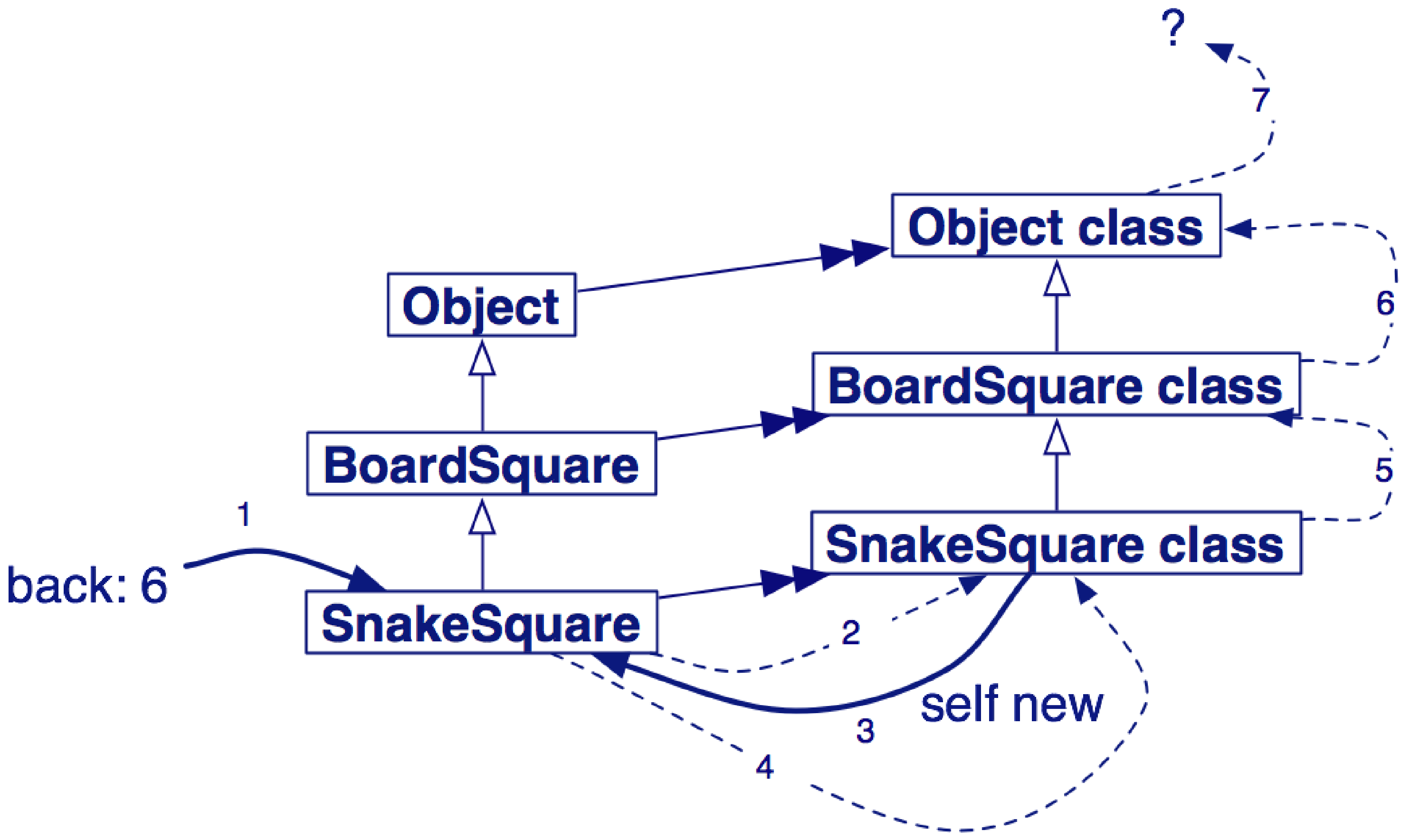
Metaprogramación y Reflexión



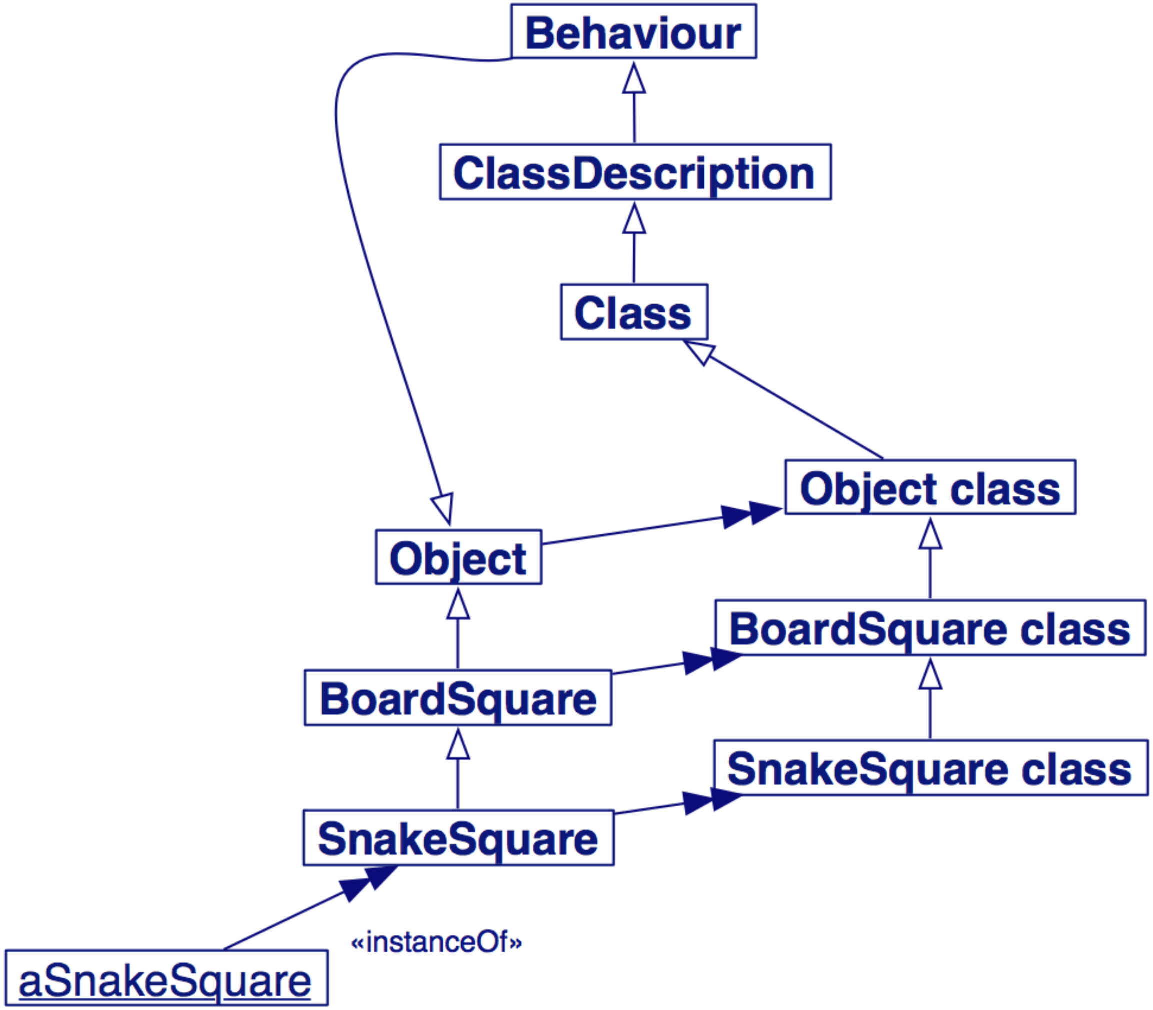
Metaprogramación y Reflexión



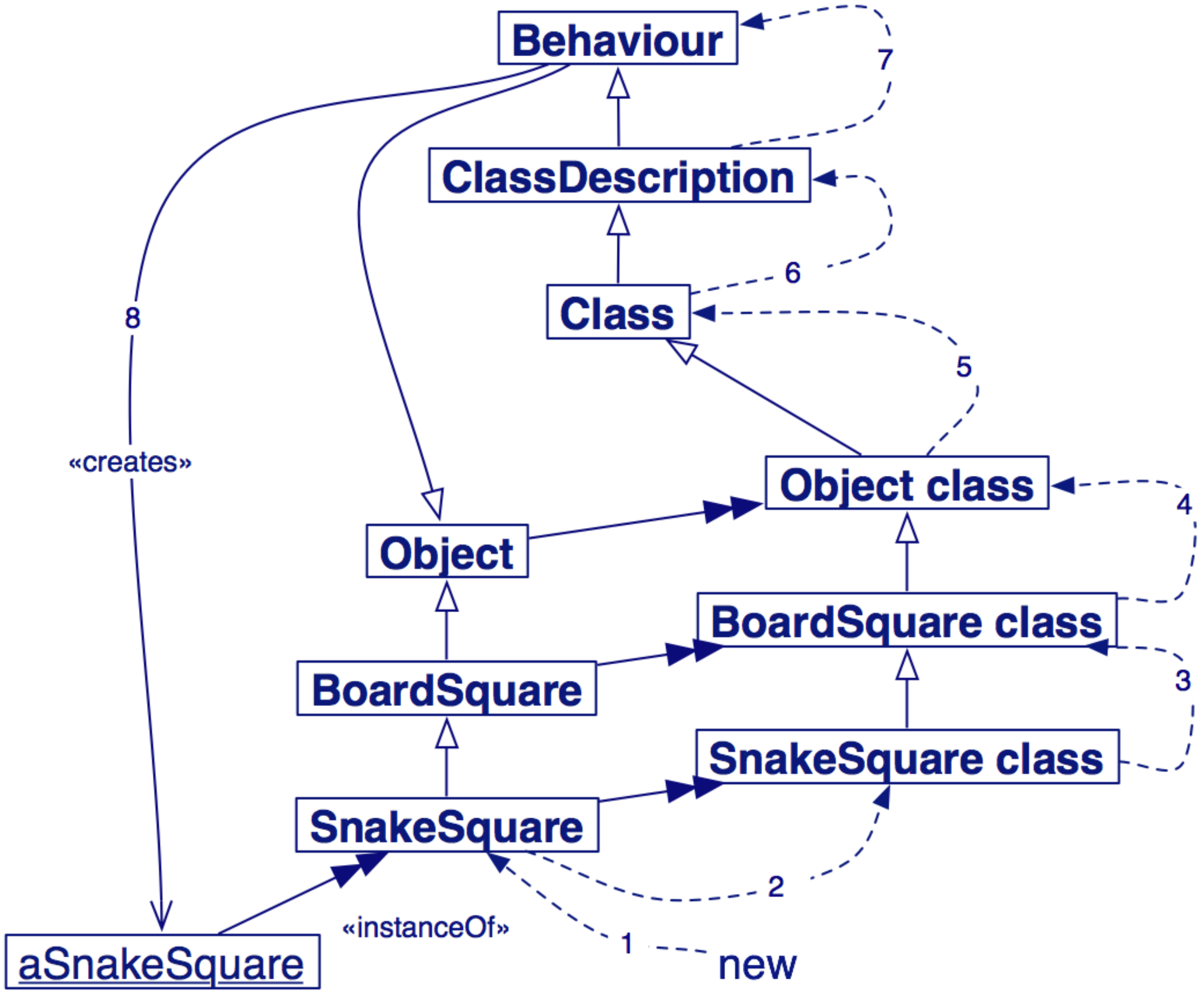
Metaprogramación y Reflexión



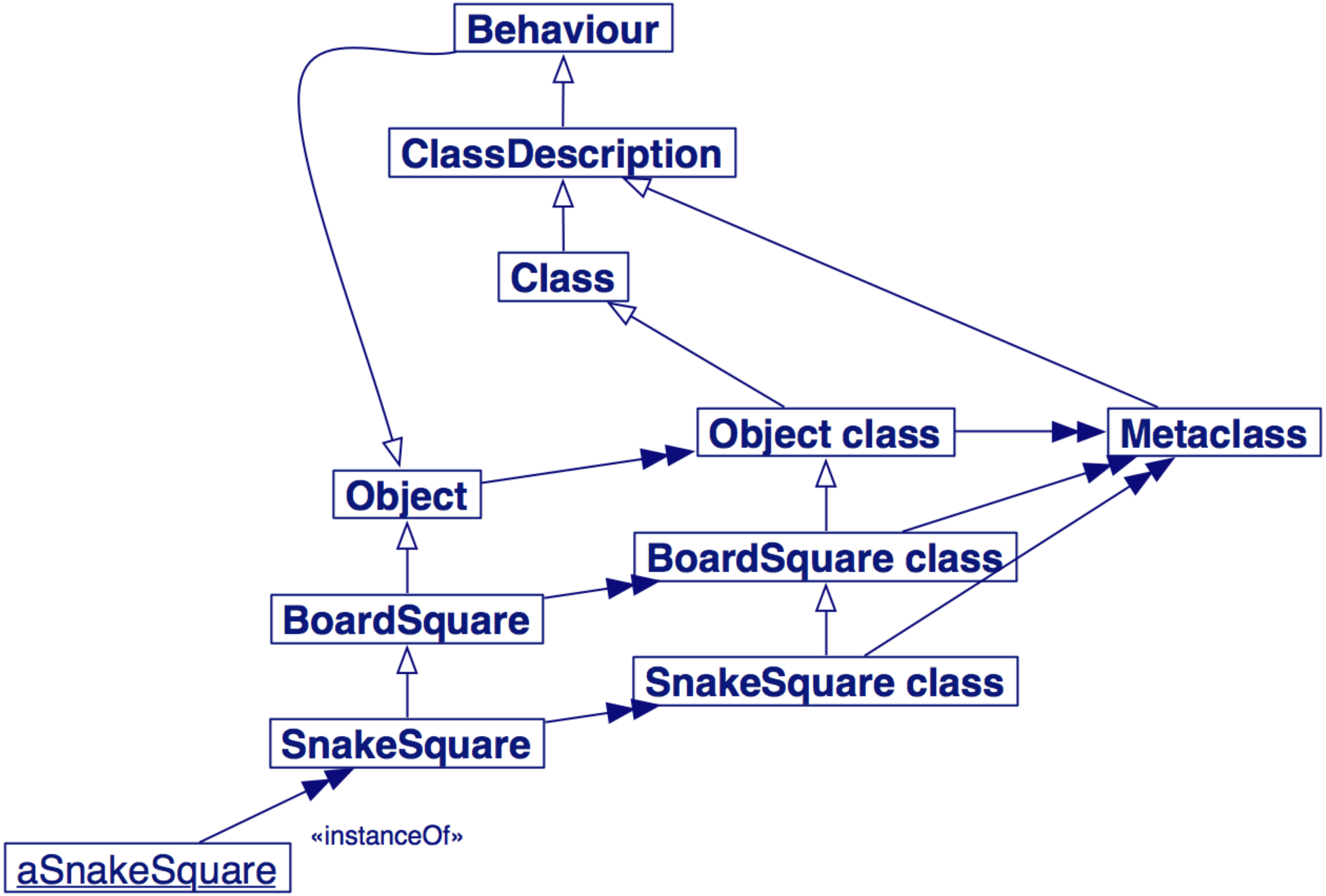
Metaprogramación y Reflexión



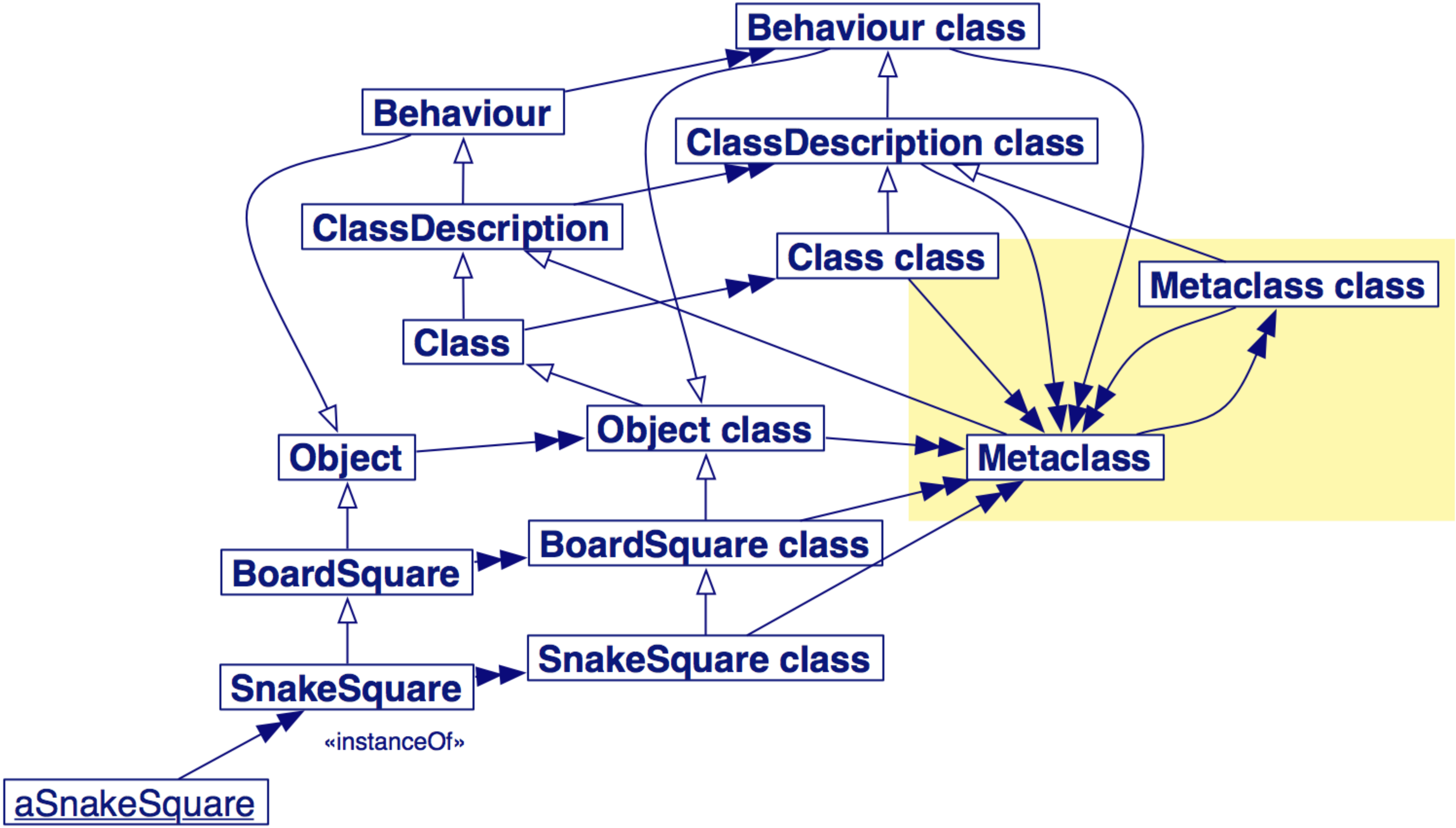
Metaprogramación y Reflexión



Metaprogramación y Reflexión



Metaprogramación y Reflexión



Metamodelo

- “That the Ideas are themselves manifestations (of the Idea-Idea) and that the Idea-Idea is a-kind-of Manifestation-Idea--which is a-kind-of itself, so that the system is completely self-describing- would have been appreciated by Plato as an extremely practical joke” – Alan Kay

Para leer

- “The Early history of Smalltalk” – Alan Kay
- “Reflection and Open Implementations” – Eric Tanter
- “Lambda: The ultimate …” – Guy Steele Jr et al.
- Y muchos más…