

Programação Orientada a Objetos II

REFLEXÃO computacional



2024/1 - Ciência da Computação – IFC Bnu - Prof. Hylson

Texto 01



MAES, Pattie. 1987. Concepts and experiments in computational reflection. OOPSLA.

Professora no MIT.

Texto 01



Professora no MIT.

MAES, Pattie. 1987. Concepts and experiments in computational reflection. OOPSLA.

Participa em laboratório que tenta reinventar radicalmente a interação humano-máquina.

<https://www.media.mit.edu/people/pattie/overview/>

MAES, 1987

1. Introdução
2. Definição de reflexão
3. Uso da reflexão
4. O que é arquitetura reflexiva
5. Arquiteturas reflexivas existentes
6. Histórico de LOOs e reflexão
7. Uma archit. reflexiva em uma LOO
8. Um novo estilo de programação



ATENÇÃO

Uma tarefa

Anotar um
termo/definição
que achou mais
impressionante

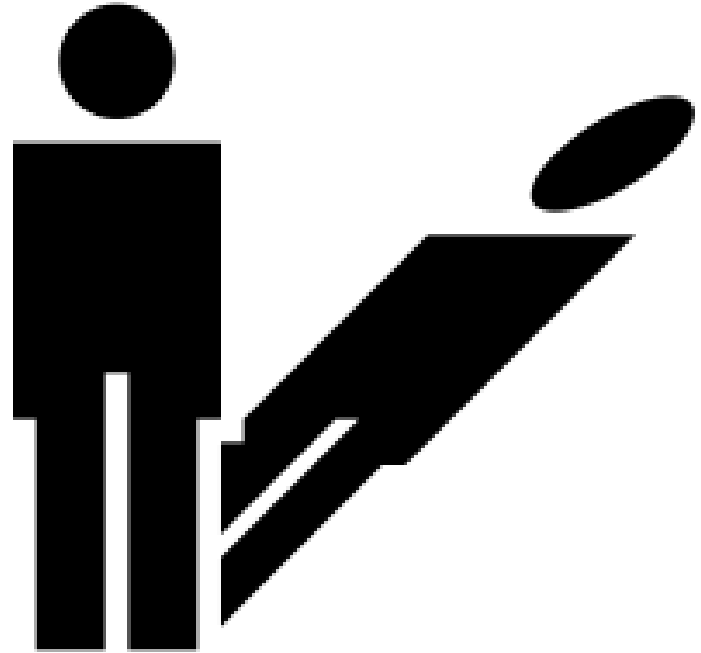


No final vou pedir pra
alguns comentarem :-)

Seção 1: introdução

Reflexão computacional

Atividade realizada
pelo sistema
computacional
quando faz
computação sobre a
própria computação



Limitações

Linguagens OO “atuais” suportam reflexão...

- * limitações (conjunto fixo de instruções)
- * implementadas **ad hoc**



projetos
obscuros

Seção 2: definição de reflexão

O que é reflexão?

- Definição aplicável a qualquer modelo computacional
 - Procedural
 - Dedutivo
 - Imperativo
 - Troca de mensagens

Reflexão computacional

- Comportamento exibido por um sistema reflexivo
 - Sistema computacional que trata dele mesmo de uma forma “**causalmente**” conectada

Sistema computacional

- Sistema computacional (ou sistema): sistema baseado em computador cujo propósito é responder questões e suportar ações de um domínio.
 - Dados
 - Relações
 - Programa para manipular os dados

Causalmente conectado

- Um sistema está causalmente conectado a um domínio se as **estruturas internas** e o **domínio** que ele representa estão ligados de forma que se um deles muda, isso leva ao efeito correspondente no outro.

Exemplo: braço robótico

- Coloca o **robô de pé,**
braços baixos
- Liga o robô



Exemplo: braço robótico

- Coloca o **robô de pé, braços baixos**
- Liga o robô
- Inicialização:
- **estado = 0;** // robô de pé, braços baixos



Exemplo: braço robótico

- estado = 0; // robô de pé, braços baixos



Exemplo: braço robótico

- estado = 0; // robô de pé, braços baixos
- Comando de voz:
“abra os braços em forma de cruz”



Exemplo: braço robótico

- ~~estado = 0; // robô de pé, braços baixos~~
- Comando de voz:
“abra os braços em forma de cruz”



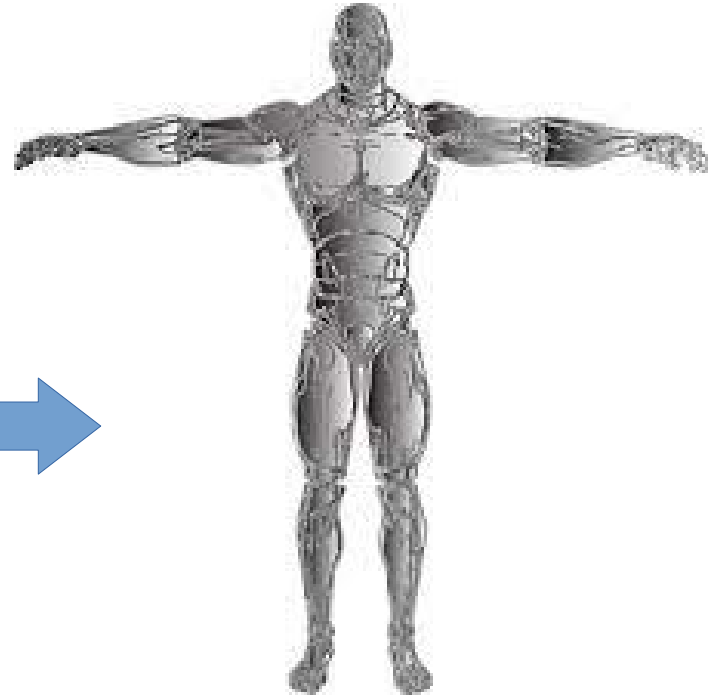
Exemplo: braço robótico

- Comando de voz:
*“abra os braços em
forma de cruz”*
- **estado = 1**; // braços
em forma de cruz



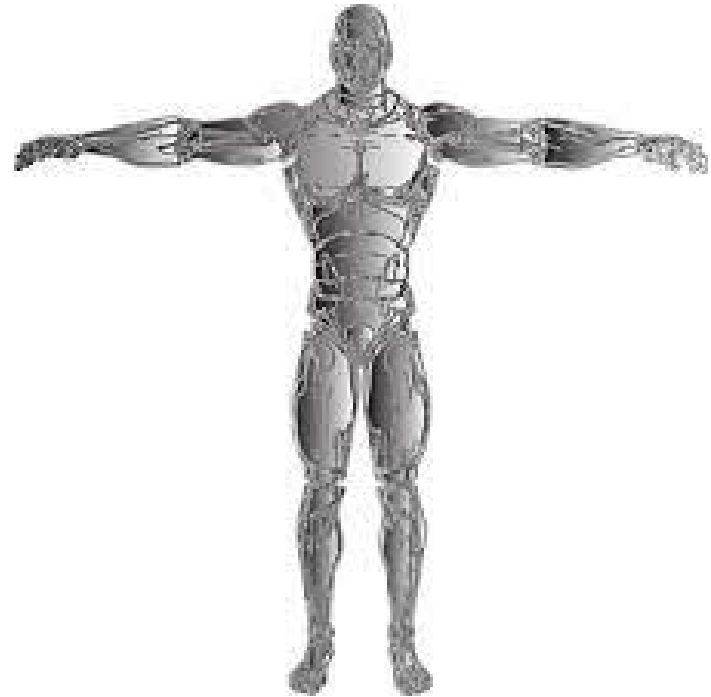
Exemplo: braço robótico

- Comando de voz:
*“abra os braços em
forma de cruz”*
- **estado = 1;** // braços
em forma de cruz



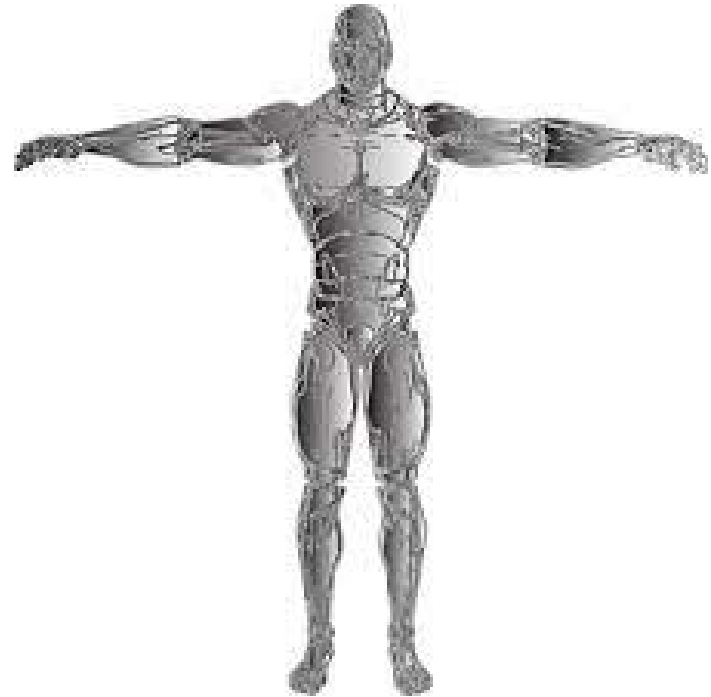
Exemplo: braço robótico

- estado = 1; // braços em forma de cruz



Exemplo: braço robótico

- estado = 1; // braços em forma de cruz
- Alguém abaixa o braço direito e move o esquerdo p/ frente



Exemplo: braço robótico

- estado = 1; // braços em forma de cruz
- Alguém abaixa o braço direito e move o esquerdo p/ frente



Exemplo: braço robótico

- ~~estado = 1; // braços em forma de cruz~~
- Alguém abaixa o braço direito e move o esquerdo p/ frente




Exemplo: braço robótico

- Alguém abaixa o braço direito e move o esquerdo p/ frente
- **estado = 2;** // braço ← esquerdo para frente, direito baixo



Exemplo: braço robótico

- Estruturas internas (variável)
 - Domínio representado (robô mecânico)
- Representação precisa
- 

Sistema reflexivo

Incorpora estruturas que representam o próprio sistema.

Sistema reflexivo

Incorpora estruturas que representam o próprio sistema.

Auto-representação

Sistema reflexivo

Incorpora estruturas que representam o próprio sistema.

Auto-representação



O sistema responde questões e atua sobre ele mesmo.

Sistema reflexivo

O sistema tem uma representação precisa de si mesmo.

Sistema reflexivo

O sistema tem uma representação precisa de si mesmo.

O sistema pode fazer modificações em si mesmo a partir de seu próprio processamento.

Seção 3: uso de reflexão

Reflexão

Fascinante

Misterioso

Reflexão

Fascinante

Misterioso

Valor prático substancial

Reflexão

Valor prático substancial: exemplos

Manter estatísticas
de desempenho

Manter informação
para depuração

Auto-otimização

Auto-modificação
(aprendizagem)

Reflexão

Não contribui diretamente para resolver problemas do domínio externo

Reflexão

Não contribui diretamente para resolver problemas do domínio externo

Contribui para a organização interna do sistema e sua interface para o mundo externo

Reflexão

Se o programador quer seguir o fluxo do programa, frequentemente inclui declarações extras

```
print(x)
```

Reflexão

Se o programador quer seguir o fluxo do programa, frequentemente inclui declarações extras

 `print(x)`

Quando termina a depuração...

Reflexão

Se o programador quer seguir o fluxo do programa, frequentemente inclui declarações extras



Quando termina a depuração...

...remove código e podem surgir ERROS

Seção 4: o que é arquitetura reflexiva

Arquitetura reflexiva

Interpretador da linguagem fornece ao programa em execução acesso a dados que representam o programa

Arquitetura reflexiva

Interpretador da linguagem fornece ao programa em execução acesso a dados que representam o programa

Programa pode conter código que define como esses dados podem ser manipulados (código reflexivo)

Arquitetura reflexiva

Interpretador tem garantia **causal** de conexão entre esses dados (representativos do programa) e os aspectos do sistema representados pelos dados

Arquitetura reflexiva

Interpretador tem garantia **causal** de conexão entre esses dados (representativos do programa) e os aspectos do sistema representados pelos dados

Modificações realizadas pelo programa em sua própria representação são refletidas no comportamento do programa e no seu funcionamento



Arquitetura reflexiva

Um “novo” paradigma de computação

Arquitetura reflexiva

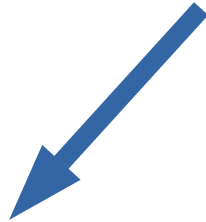
Um “novo” paradigma de computação

O sistema tem uma parte de
objetos e uma parte reflexiva

Arquitetura reflexiva

Um “novo” paradigma de computação

O sistema tem uma parte de
objetos e uma parte reflexiva

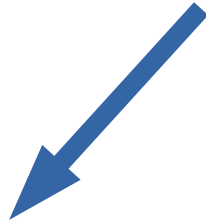


Resolve problemas
do domínio

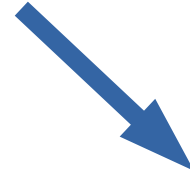
Arquitetura reflexiva

Um “novo” paradigma de computação

O sistema tem uma parte de
objetos e uma parte reflexiva



Resolve problemas
do domínio

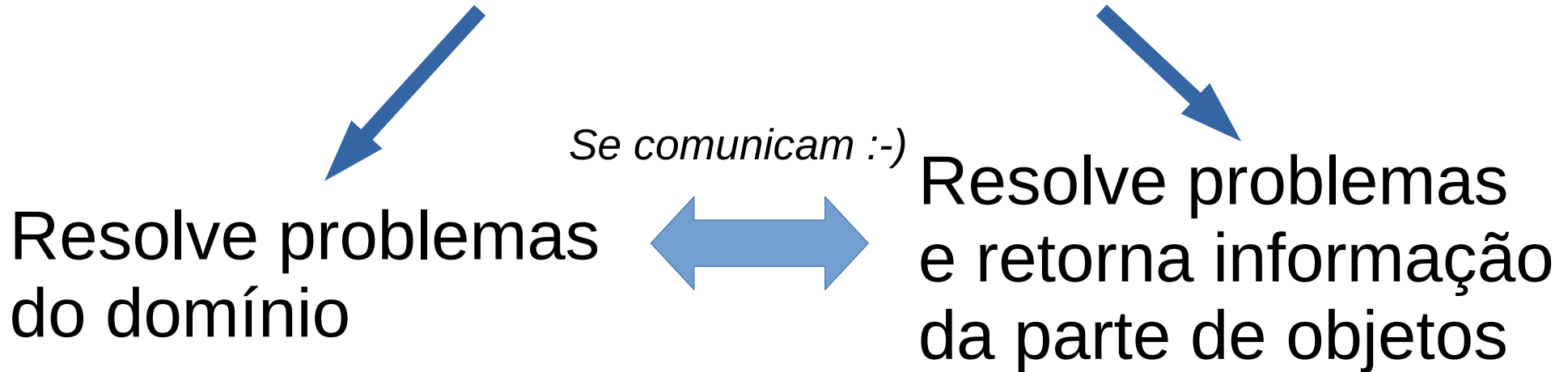


Resolve problemas
e retorna informação
da parte de objetos

Arquitetura reflexiva

Um “novo” paradigma de computação

O sistema tem uma parte de objetos e uma parte reflexiva



Exemplo

Exibir informação de rastreamento de programa em um sistema baseado em regras

Exemplo

Exibir informação de rastreamento de programa em um sistema baseado em regras

Com reflexão: adicionar código reflexivo...

```
IF uma regra tem a maior prioridade na situação  
THEN imprima a regra e os dados que levaram a essa condição
```

Exemplo

Exibir informação de rastreamento de programa em um sistema baseado em regras

Sem reflexão: adicionar um código...

```
print(regra+dados)
```

...após cada regra

Benefício da reflexão

Melhor modularidade

Benefício da reflexão

Melhor modularidade

O que leva a sistemas mais...

Benefício da reflexão

Melhor modularidade

O que leva a sistemas mais...

...gerenciáveis, legíveis, fáceis
de compreender e modificar.

Seção 5: arquitetura reflexivas existentes

Arquiteturas reflexivas

Linguagens...

Procedurais

Lógicas

Baseadas
em regras

Arquiteturas reflexivas

Procedurais

Exemplos:
3-LISP, BROWN

Função reflexiva: local (temporária),
manipula dados que representam o código, o
ambiente e a continuidade do processamento

Arquiteturas reflexivas

Lógicas

Exemplos:
FOL, Meta-prolog

Meta-teoria: deduções sobre outra teoria, em vez de sobre problemas do domínio.

Exemplo: `provable(Theory, Goal)`

Arquiteturas reflexivas

Baseada
em regras

Exemplos:
Teiresias, Soar

Meta-regras: especificam computação sobre o processamento em execução. Exemplo de variável disponível para uma meta-regra: existe-um-impasse-no-processo-de-inferência

Arquiteturas reflexivas

Característica comum: possuem...

Arquiteturas reflexivas

Característica comum: possuem...

... interpretador meta-circular

Arquiteturas reflexivas

Interpretador meta-circular:
uma representação da interpretação na linguagem

Arquiteturas reflexivas

Interpretador meta-circular:
uma representação da interpretação na linguagem

A representação é usada para
executar (interpretar) a linguagem.

Arquiteturas reflexivas

Interpretador meta-circular:
é uma maneira fácil de garantir a **conexão causal**

***Auto-representação** é exatamente o
processo de interpretação meta-circular*

Arquiteturas reflexivas

Interpretador meta-circular:
é uma maneira fácil de garantir a conexão causal

*Auto-representação é exatamente o
processo de interpretação meta-circular*

Denominação: reflexão **procedural**

Arquiteturas reflexivas

A mesma representação implementa o sistema e contém dados sobre o sistema.

Reflexão
procedural

Arquiteturas reflexivas

A mesma representação implementa o sistema e contém dados sobre o sistema.

Reflexão
procedural

A linguagem e os dados possuem um
formato comum

Arquiteturas reflexivas

A mesma representação implementa o sistema e contém dados sobre o sistema.

Reflexão
procedural

A linguagem e os dados possuem um
formato comum

Os programas podem ser vistos como
estruturas de dados da linguagem.

Arquiteturas reflexivas

Um problema: servir a dois propósitos...

Reflexão
procedural

Arquiteturas reflexivas

Um problema: servir a dois propósitos...

Reflexão
procedural

Ser **reflexivo**: prover uma boa base de informações sobre o sistema

Ser **eficiente**: implementar o sistema.

Arquiteturas reflexivas

Um problema: servir a dois propósitos...

Requisitos
contraditórios

Ser **reflexivo**: prover uma boa base de informações sobre o sistema

Ser **eficiente**: implementar o sistema.

Arquiteturas reflexivas

Proposta: desenvolver uma arquitetura na qual a auto-representação do sistema NÃO seja a implementação do sistema

Arquiteturas reflexivas

Proposta: desenvolver uma arquitetura na qual a auto-representação do sistema NÃO seja a implementação do sistema

Denominação: reflexão **declarativa**

Arquiteturas reflexivas

Proposta: desenvolver uma arquitetura na qual a auto-representação do sistema NÃO seja a implementação do sistema

Denominação: reflexão **declarativa**

Podem ser feitas declarações sobre o sistema. Exemplo: o sistema deve executar em um tempo máximo especificado.

Arquiteturas reflexivas

A auto-representação é composta por um conjunto de restrições que o sistema deve cumprir.

Reflexão
declarativa

Arquiteturas reflexivas

A auto-representação é composta por um conjunto de restrições que o sistema deve cumprir.

Reflexão
declarativa

Garantir a conexão causal é mais difícil: o interpretador precisa traduzir representações declarativas em procedurais, que implementam o sistema

Revisão: arquiteturas reflexivas

Reflexão procedural
Reflexão declarativa

Arquiteturas em linguagens procedurais,
lógicas e baseadas em regras

Seção 6: histórico de LOOs e reflexão

Pausa

São muitos slides

“Mãos de nuvem”

Reflexão na orientação a objetos

SIMULA e SMALLTALK-72: não incorporam facilidades para realizar reflexão

Reflexão na orientação a objetos

SIMULA e SMALLTALK-72: não incorporam facilidades para realizar reflexão

Mas as características da OO favorecem o desenvolvimento de reflexão

Reflexão na orientação a objetos

Recursos de OO: abstração

Um objeto é livre para desempenhar seu papel da maneira que quiser

Reflexão na orientação a objetos

Recursos de OO: abstração

Um objeto é livre para desempenhar seu papel da maneira que quiser



O objeto não realiza computação apenas sobre o domínio, mas também sobre como vai realizar a computação

Reflexão na orientação a objetos

Momento de desenvolvimento da OO:

Reflexão na orientação a objetos

Momento de desenvolvimento da OO:

A distinção entre classes e instâncias é necessária?

Quais formas de herança são necessárias?

Como deve ser o formato de mensagens entre objetos?

Reflexão na orientação a objetos

Reflexão torna possível especializar o interpretador da linguagem, com a própria linguagem

Reflexão na orientação a objetos

Reflexão torna possível especializar o interpretador da linguagem, com a própria linguagem

Objetos podem fornecer uma representação modificável sobre como eles devem ser impressos.

Objetos podem adotar um procedimento diferente para criar instâncias.

Reflexão na orientação a objetos

Encapsulamento de dados do
domínio com dados reflexivos

Reflexão na orientação a objetos

Encapsulamento de dados do domínio com dados reflexivos

Quando o objeto foi criado?

Por quem o objeto foi criado?

Quais restrições o objeto precisa cumprir?

Reflexão na orientação a objetos

Gerenciamento de objetos com reflexão:

Reflexão na orientação a objetos

Gerenciamento de objetos com reflexão:

Encapsular o valor de um dado com seu valor padrão e a forma de computar esse valor.

Acionar procedimentos quando um valor é instanciado ou alterado.

Reflexão na orientação a objetos

Até agora (1987), reflexão nas linguagens
é feita *ad hoc*

*Smalltalk-72 e Flavors: objeto não
contém somente informação sobre
entidade representada pelo objeto, mas
também sobre a representação do objeto*

Reflexão na orientação a objetos

Smalltalk-72 e Flavors: objeto não contém somente informação sobre entidade representada pelo objeto, mas também sobre a representação do objeto

Exemplo: classe Pessoa contém: I) método para calcular idade da pessoa (domínio), e II) método que informa como deve ser a impressão da pessoa (reflexivo)

Reflexão na orientação a objetos

Flavors: possui conjunto de métodos reflexivos

Qual é a
descrição sobre
uma mensagem?

Como criar uma
instância?

```
:DESCRIBE (message): ()  
GET-HANDLER-FOR: (OBJECT OPERATION)  
MAKE-INSTANCE: (FLAVOR-NAME &REST INIT-OPTIONS)  
:OPERATION-HANDLED-P (message): (OPERATION)  
SYS:PRINT-SELF (message :PRINT-SELF):  
                (OBJECT STREAM PRINT-DEPTH SLASHIFY-P)  
:SEND-IF-HANDLES (message): (MESSAGE &REST ARGS)  
:WHICH-OPERATIONS (message): ()
```


Reflexão na orientação a objetos

Problemas na reflexão dessas linguagens...

Conjunto fixo
de facilidades
para reflexão

Reflexão na orientação a objetos

Problemas na reflexão dessas linguagens...

Conjunto fixo
de facilidades
para reflexão



Novas facilidades?
Modifique o interpretador!

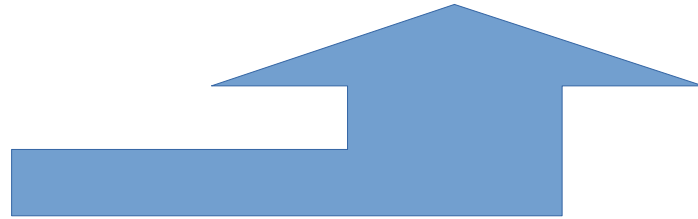
Reflexão na orientação a objetos

Problemas na reflexão dessas linguagens...

Conjunto fixo
de facilidades
para reflexão

Mistura níveis de
objeto e reflexão

Objeto Livro: slot *Autor* é
do domínio ou do objeto?



Reflexão na orientação a objetos

Solução: criação de **Meta-classes**



*Distinguem informação de
objeto e informação reflexiva*

Reflexão na orientação a objetos

Solução: criação de Meta-classes



*Distinguem informação de
objeto e informação reflexiva*

Meta-classes **SÓ** especificam informações
internas do sistema, sobre a classe



Reflexão na orientação a objetos



*Smalltalk-80 ainda mistura
informação de domínio com
informação de implementação*



*Usuários de Smalltalk-80 relatam
confusão no uso de meta-classes*

Reflexão na orientação a objetos

Novas linguagens: Plasma,
Actors, RLL, ObjVLisp

*Tudo são objetos: class,
instância, meta-classe, variável,
método, mensagem, ambiente*



Reflexão na orientação a objetos

Primeiro recurso de reflexão: o
que a instância é (de que tipo)

*Recurso disponibilizado de
instâncias para: classes,
meta-classes, mensagens, etc*

Reflexão na orientação a objetos

Limitações:



Não reconhecem reflexão
como conceito primário

Reflexão na orientação a objetos

Limitações:



Não reconhecem reflexão
como conceito primário

Possuem conjunto finito
de recursos de reflexão

Reflexão na orientação a objetos

Limitações:



Não reconhecem reflexão
como conceito primário

Possuem conjunto finito
de recursos de reflexão

São implementados *ad hoc*

Seção 7: Arquitetura reflexiva em uma linguagem OO

Arquitetura reflexiva em uma LOO

Arquitetura da linguagem 3-KRS

Arquitetura reflexiva em uma LOO

1

Divisão ordenada entre nível
de domínio e nível reflexivo

Arquitetura reflexiva em uma LOO

1

Divisão ordenada entre nível
de domínio e nível reflexivo

Cada objeto tem um meta-objeto
(que aponta para o objeto)

Arquitetura reflexiva em uma LOO

1

Divisão ordenada entre nível de domínio e nível reflexivo

Cada objeto tem um meta-objeto
(que aponta para o objeto)

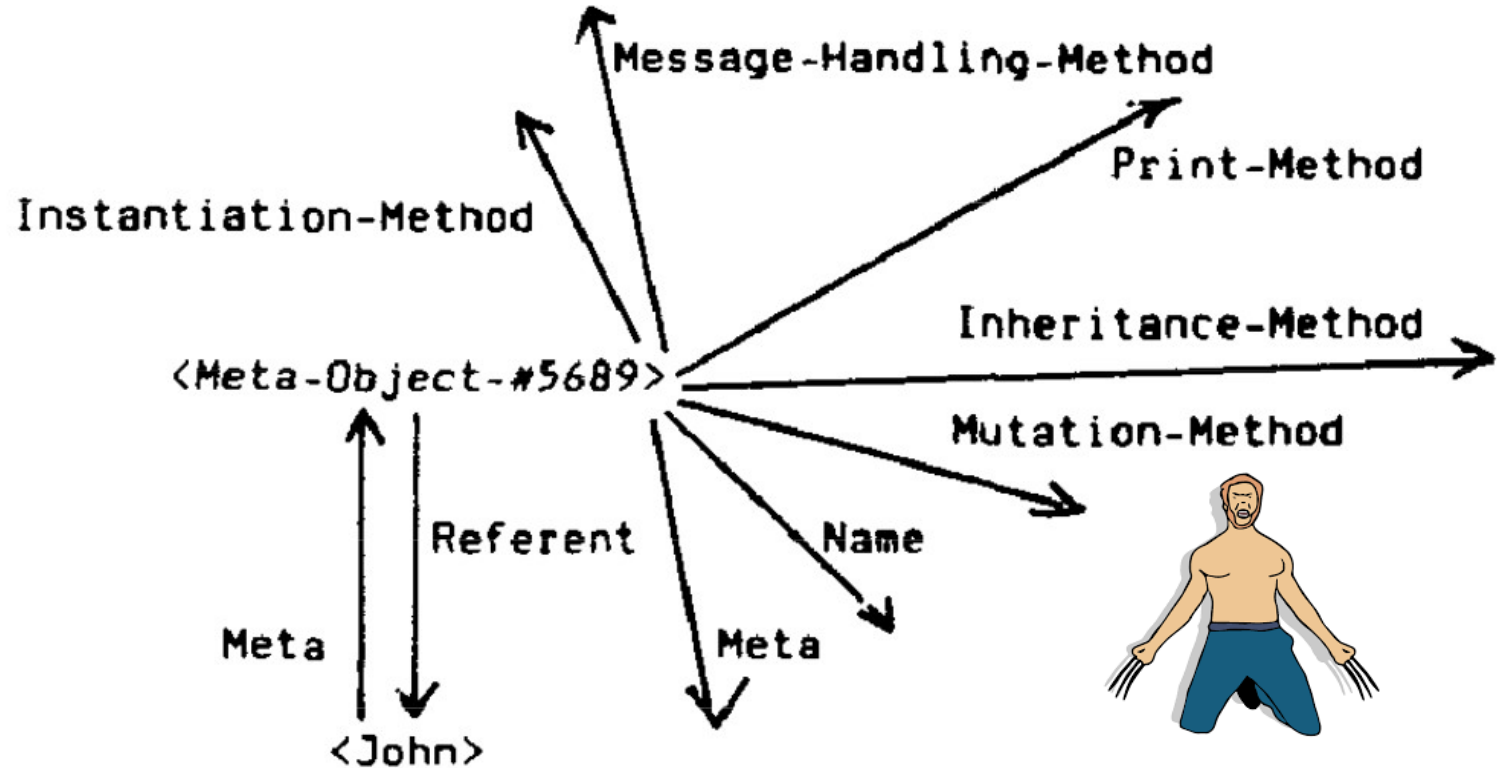
Contém dados sobre o domínio

Contém dados de reflexão

Arquitetura reflexiva em uma LOO

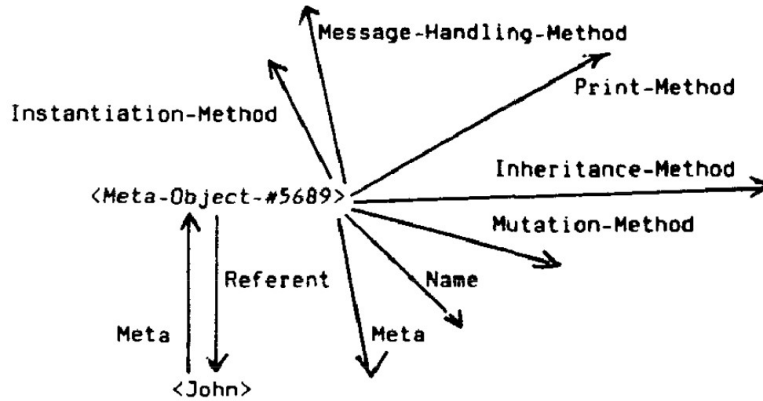
1

Exemplo



Arquitetura reflexiva em uma LOO

1

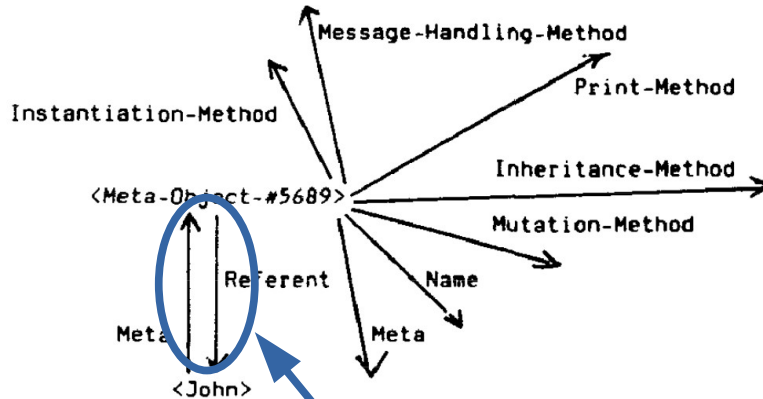


Exemplo

Arquitetura reflexiva em uma LOO

1

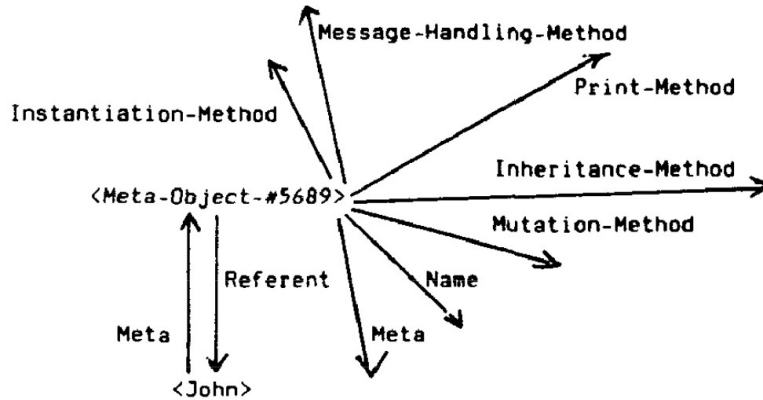
Exemplo



Relação 1-1.
Poderia combinar?

Arquitetura reflexiva em uma LOO

1



Exemplo

NÃO!

Manter **separados**
dados de domínio
e reflexão!

Relação 1-1.
Poderia combinar?

Arquitetura reflexiva em uma LOO

2

Auto-representação **uniforme**:
tudo é objeto.

*Instância, classe, slot (atributo),
métodos, meta-objetos, mensagens, ...*

Arquitetura reflexiva em uma LOO

2

Auto-representação **uniforme**:
tudo é objeto.



*Instância, classe, slot (atributo),
métodos, meta-objetos, mensagens, ...*



Todo aspecto pode ser refletido!

Arquitetura reflexiva em uma LOO

2

Meta-objetos são criados de forma
“lazy”: apenas quando necessário

Auto-representação
uniforme

Arquitetura reflexiva em uma LOO

3

Auto-representação **completa**

Meta-objetos contém toda informação sobre o objeto disponível na linguagem

Arquitetura reflexiva em uma LOO

3

Auto-representação **completa**

Meta-objetos contém toda informação sobre o objeto disponível na linguagem

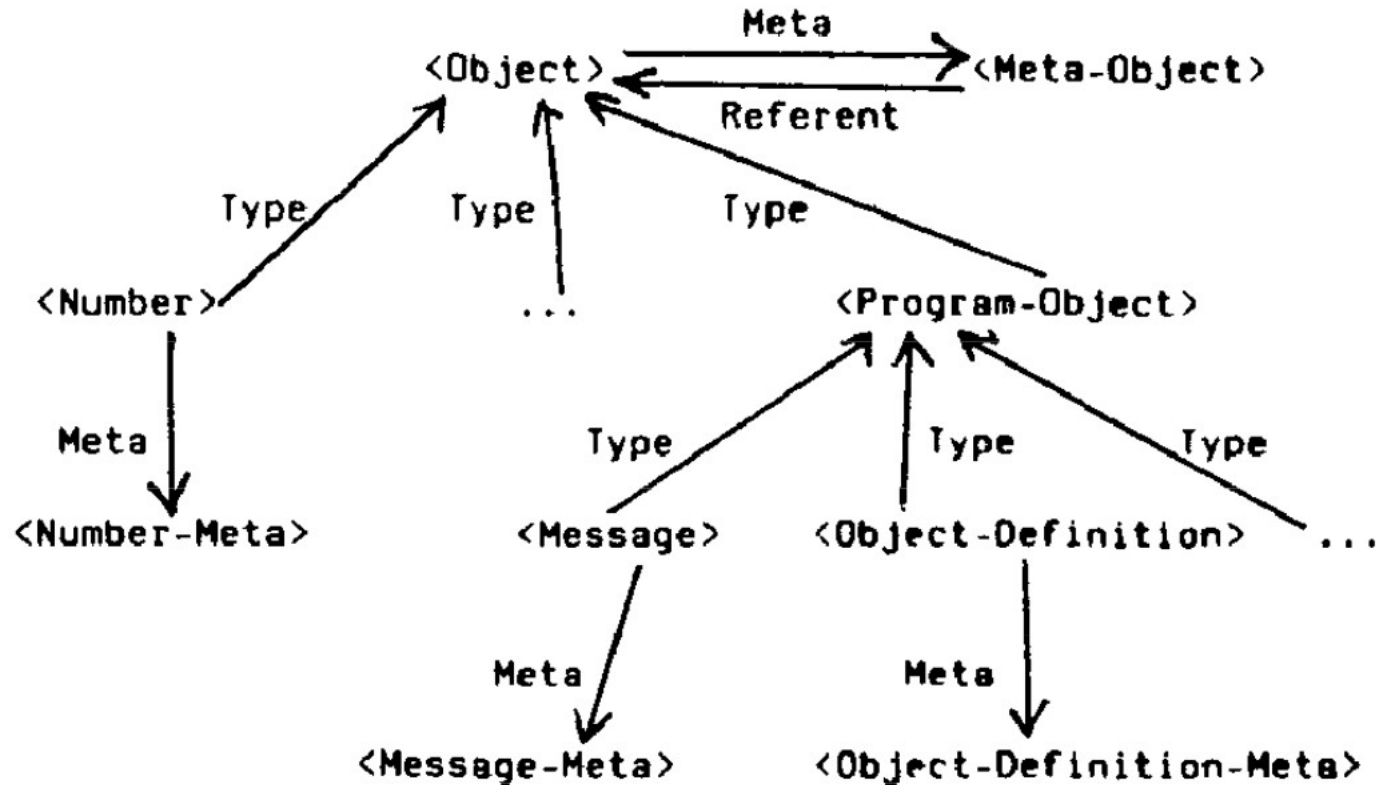


O conteúdo do meta-objeto é base do interpretador!

Arquitetura reflexiva em uma LOO

3

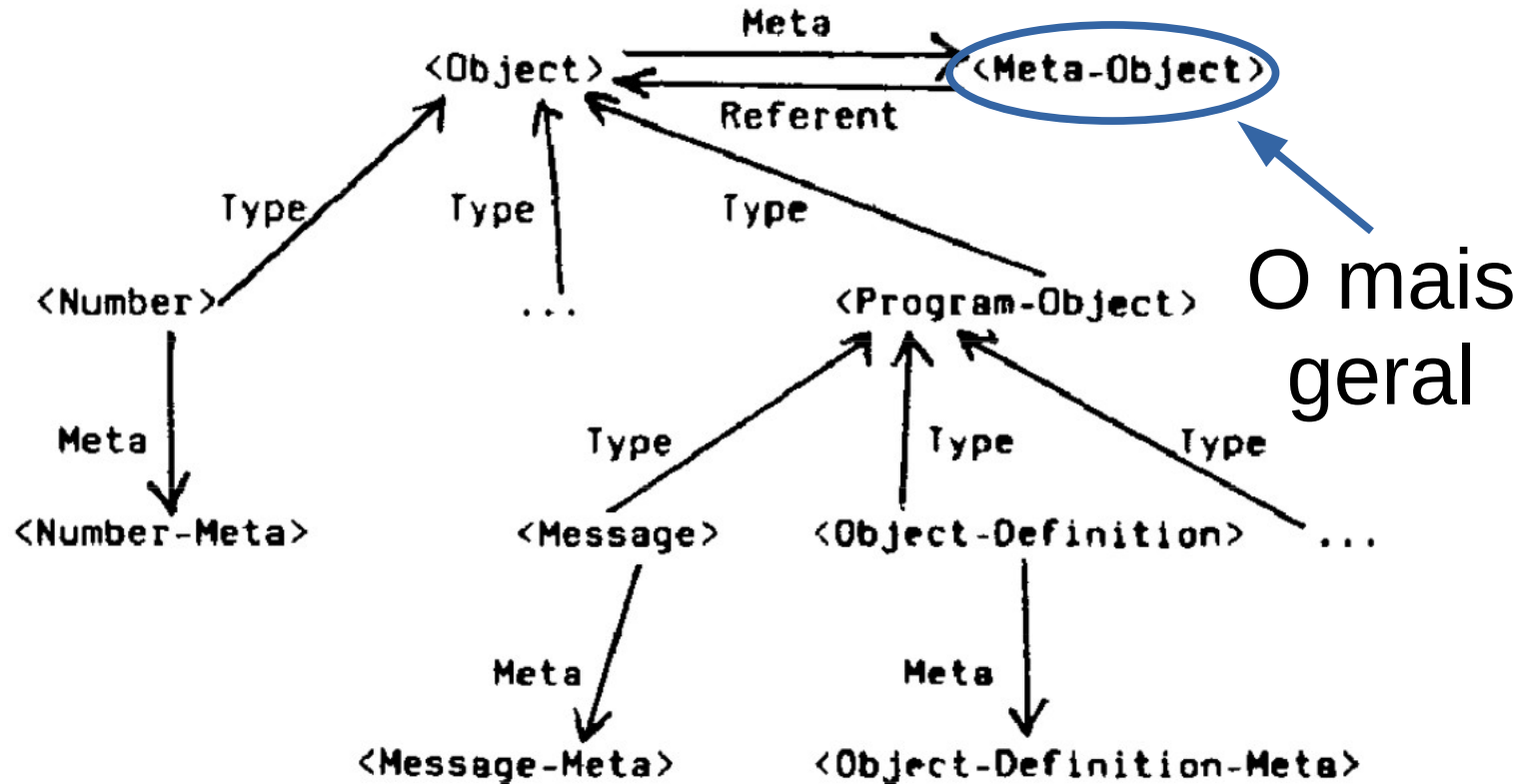
Auto-representação
completa



Arquitetura reflexiva em uma LOO

3

Auto-representação
completa



Arquitetura reflexiva em uma LOO

4

Consistência: a auto-representação é usada para implementar o sistema

Cada ação realizada em um objeto é feita requisitando o meta-objeto



Arquitetura reflexiva em uma LOO

4

Consistência: a auto-representação é usada para implementar o sistema

Cada ação realizada em um objeto é feita requisitando o meta-objeto

Exemplo: criar uma instância, enviar uma mensagem, avaliar um objeto, ...

Arquitetura reflexiva em uma LOO

4

Consistência

Para evitar um loop infinito, um segundo interpretador implementa o comportamento padrão.

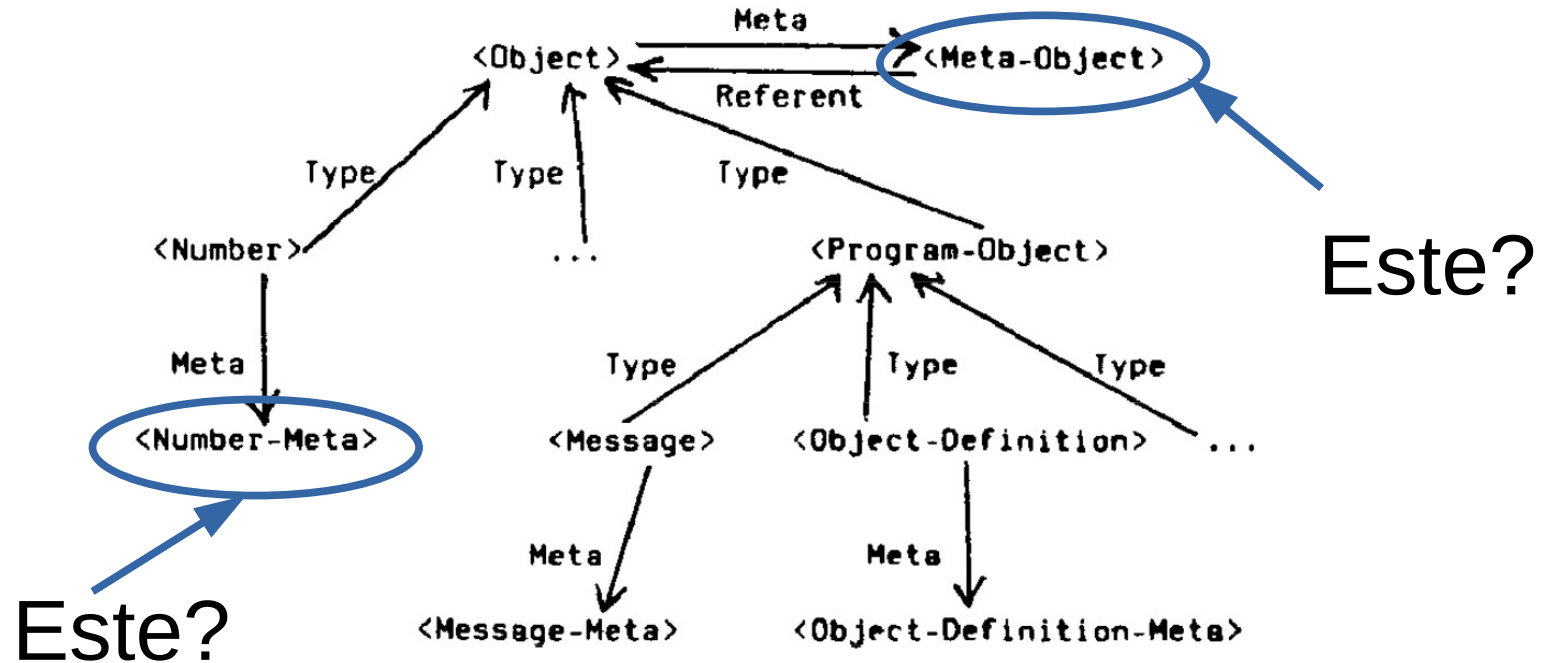


*Verifica qual
“meta” código
deve ser
executado*

Arquitetura reflexiva em uma LOO

4

Consistência



Qual “meta” código deve ser executado para exibir um número?

Arquitetura reflexiva em uma LOO

5

A auto-representação pode ser **modificada**, refletindo na computação em progresso



<https://www.youtube.com/watch?v=clpBpGQ0XTI>

Seção 8: novo estilo de programação (1987)

Novo estilo de programação

Unidade básica de
informação: objeto

Novo estilo de programação

Unidade básica de
informação: objeto

*Todo objeto tem um
meta-objeto*

Novo estilo de programação

Unidade básica de
informação: objeto

*Meta-objeto possui
informações de
implementação e
interpretação do objeto*

*Todo objeto tem um
meta-objeto*

Novo estilo de programação

Unidade básica de
informação: objeto

*Meta-objeto possui
informações de
implementação e
interpretação do objeto*

*Todo objeto tem um
meta-objeto*

*Objeto pode pausar a
computação, refletir sobre
si mesmo, e modificar seu
comportamento futuro.*

Novo estilo de programação

A implementação dos objetos e dos meta-objetos pode ser feita separadamente

Novo estilo de programação

A implementação dos objetos e dos meta-objetos pode ser feita separadamente

*Existe um padrão de comunicação
entre objeto e meta-objeto*

Novo estilo de programação

A implementação dos objetos e dos meta-objetos pode ser feita separadamente

*Existe um padrão de comunicação
entre objeto e meta-objeto*



*Módulos (partes de programas)
podem se comunicar entre si*

Novo estilo de programação

A implementação dos objetos e dos meta-objetos pode ser feita separadamente

*Existe um padrão de comunicação
entre objeto e meta-objeto*



*Módulos (partes de programas)
podem se comunicar entre si*

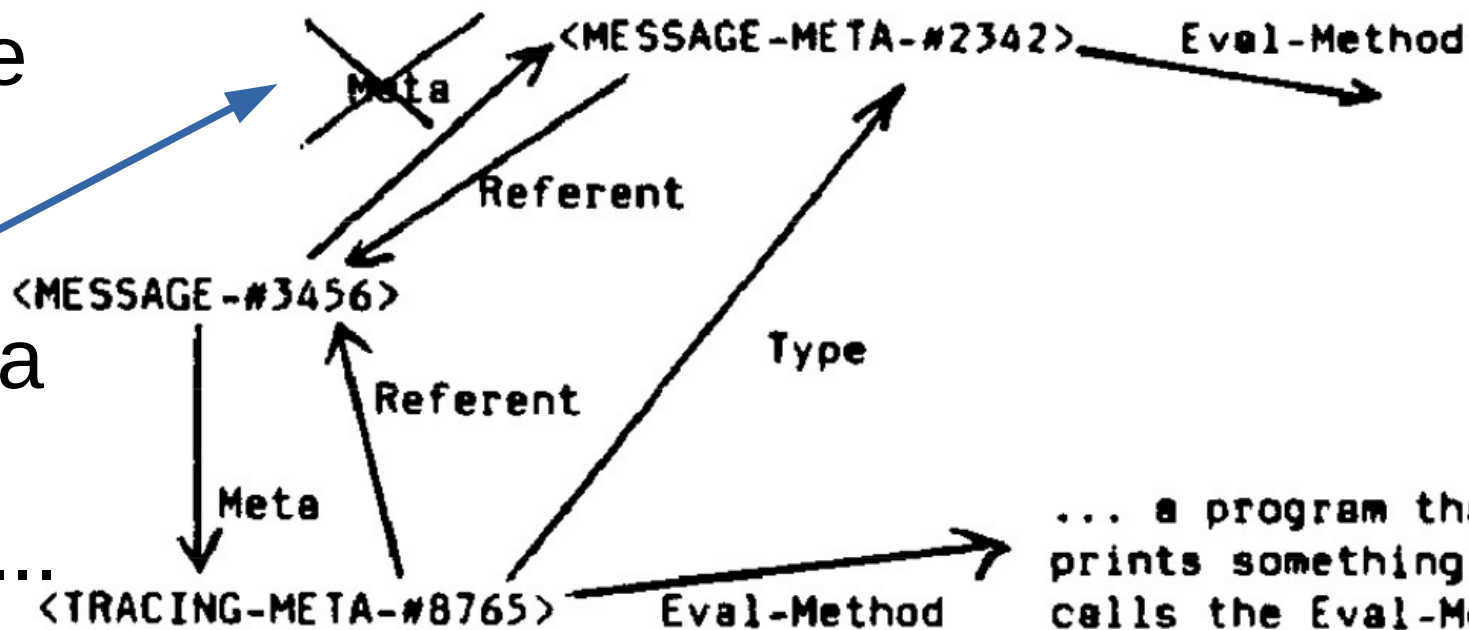
*Bibliotecas de reflexão
podem ser construídas*



Novo estilo de programação

Exemplo:
rastreo de
código

Antigo meta
objeto foi
substituído...

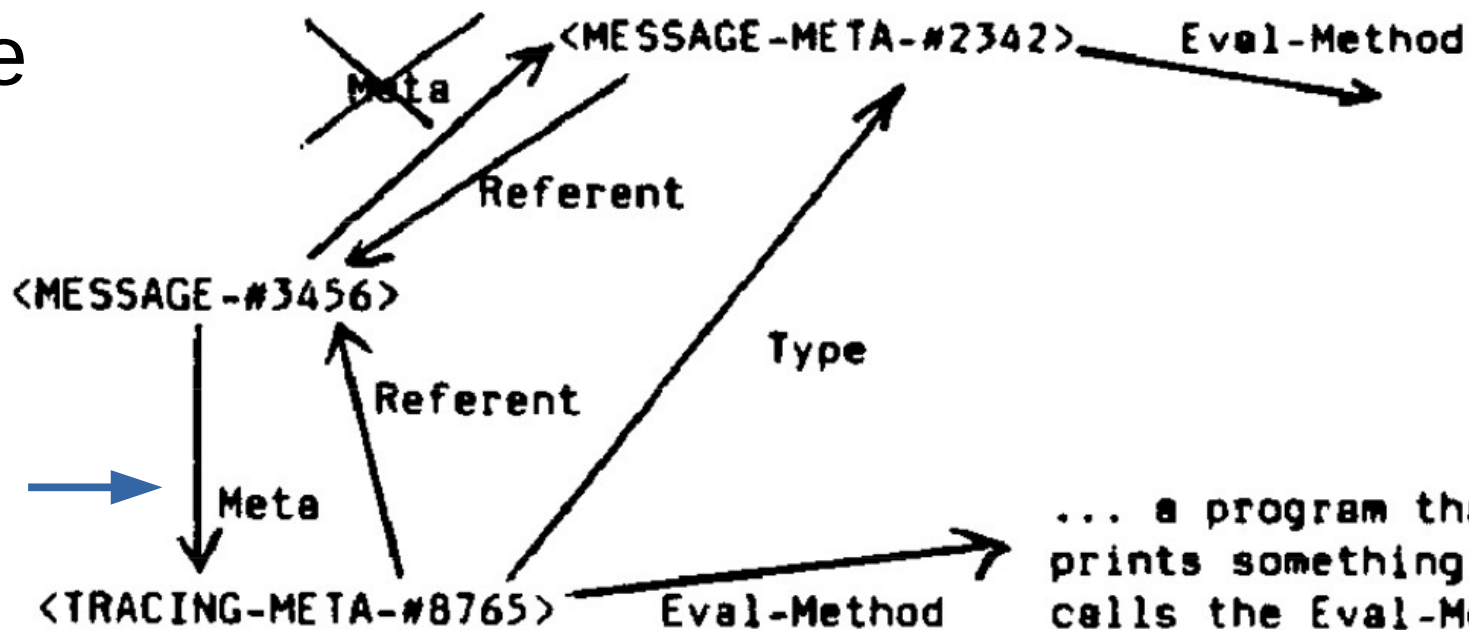


... a program that
prints something, then
calls the Eval-Method
of the type, followed
by another print-out

Novo estilo de programação

Exemplo:
rastreo de
código

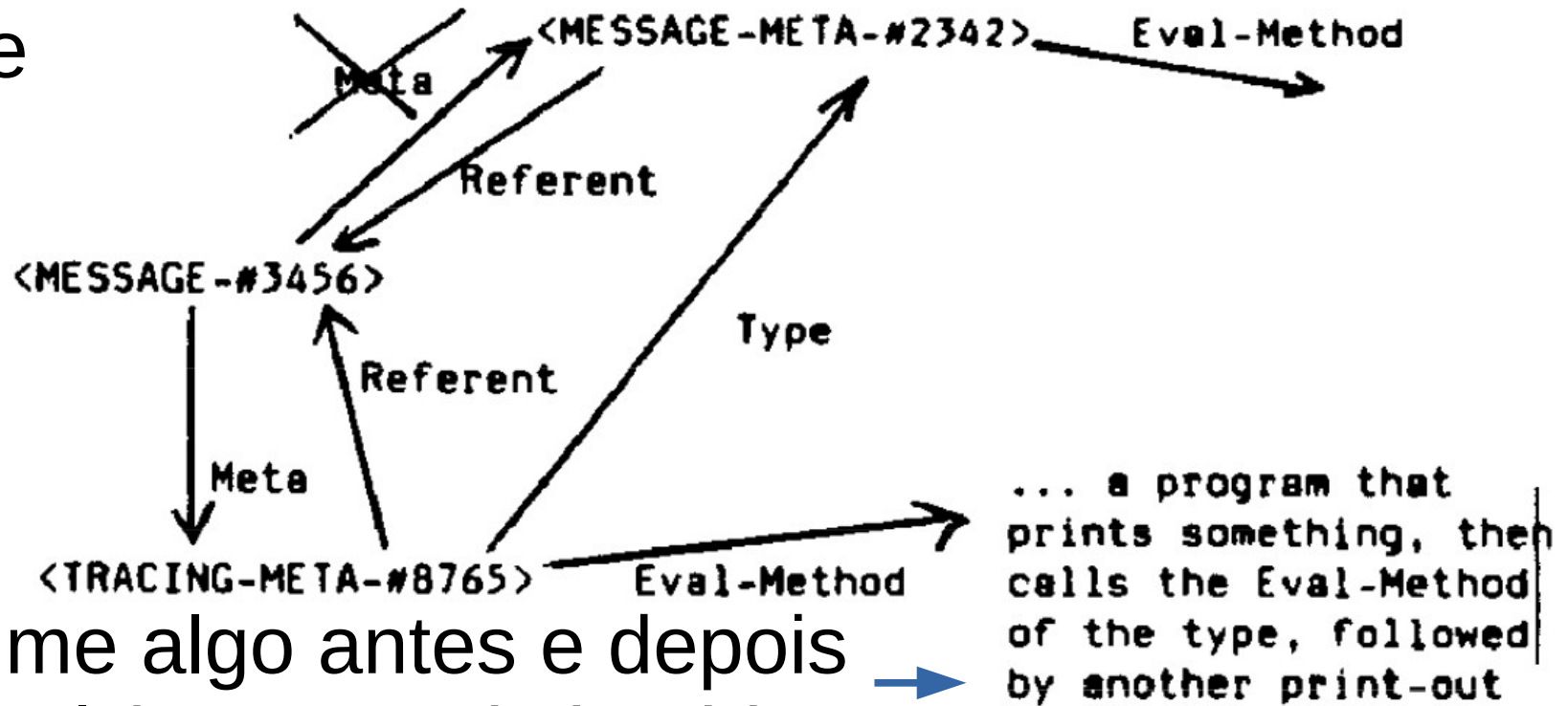
...novo
meta é um
tracing



... a program that
prints something, then
calls the Eval-Method
of the type, followed
by another print-out

Novo estilo de programação

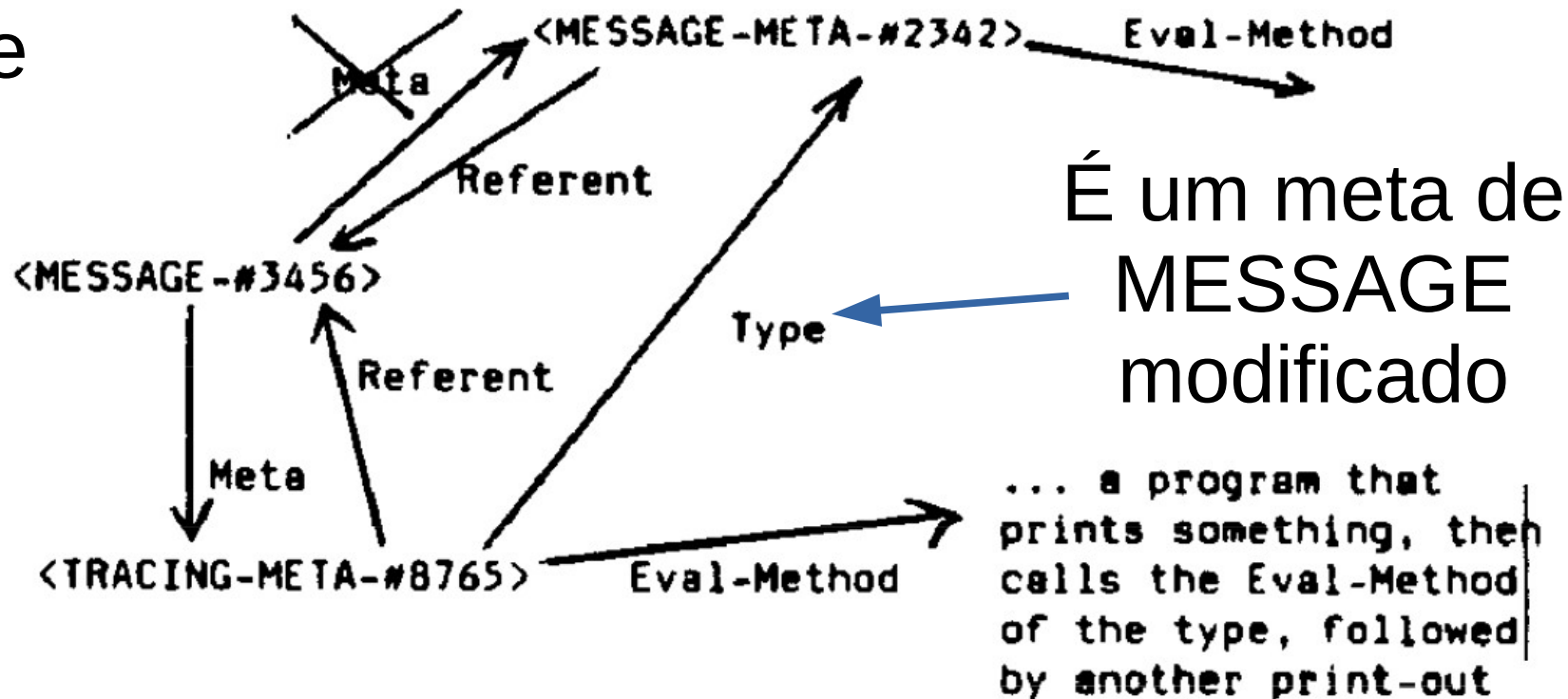
Exemplo:
rastreio de
código



Imprime algo antes e depois
do Eval (execução) do objeto

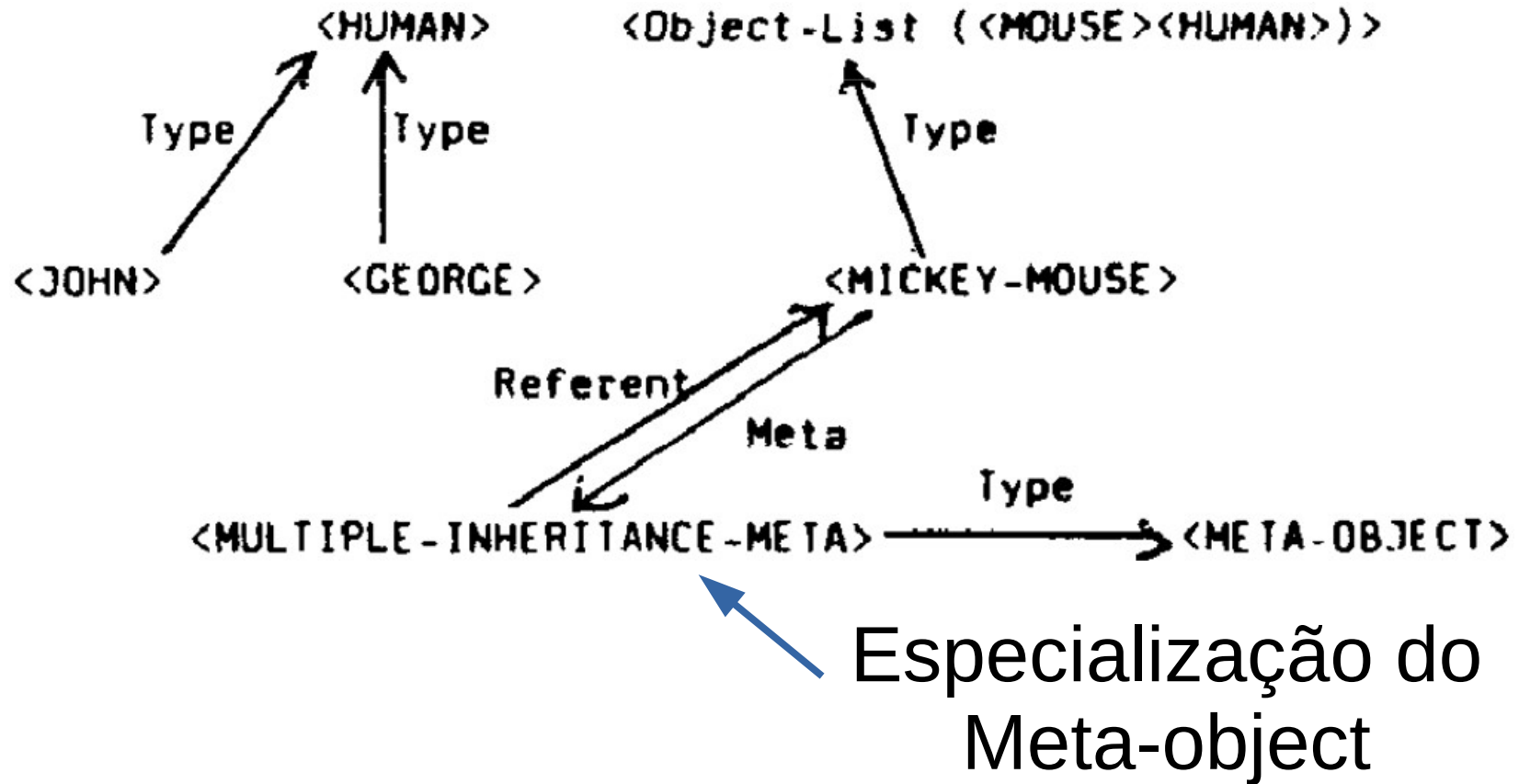
Novo estilo de programação

Exemplo:
rastreio de
código



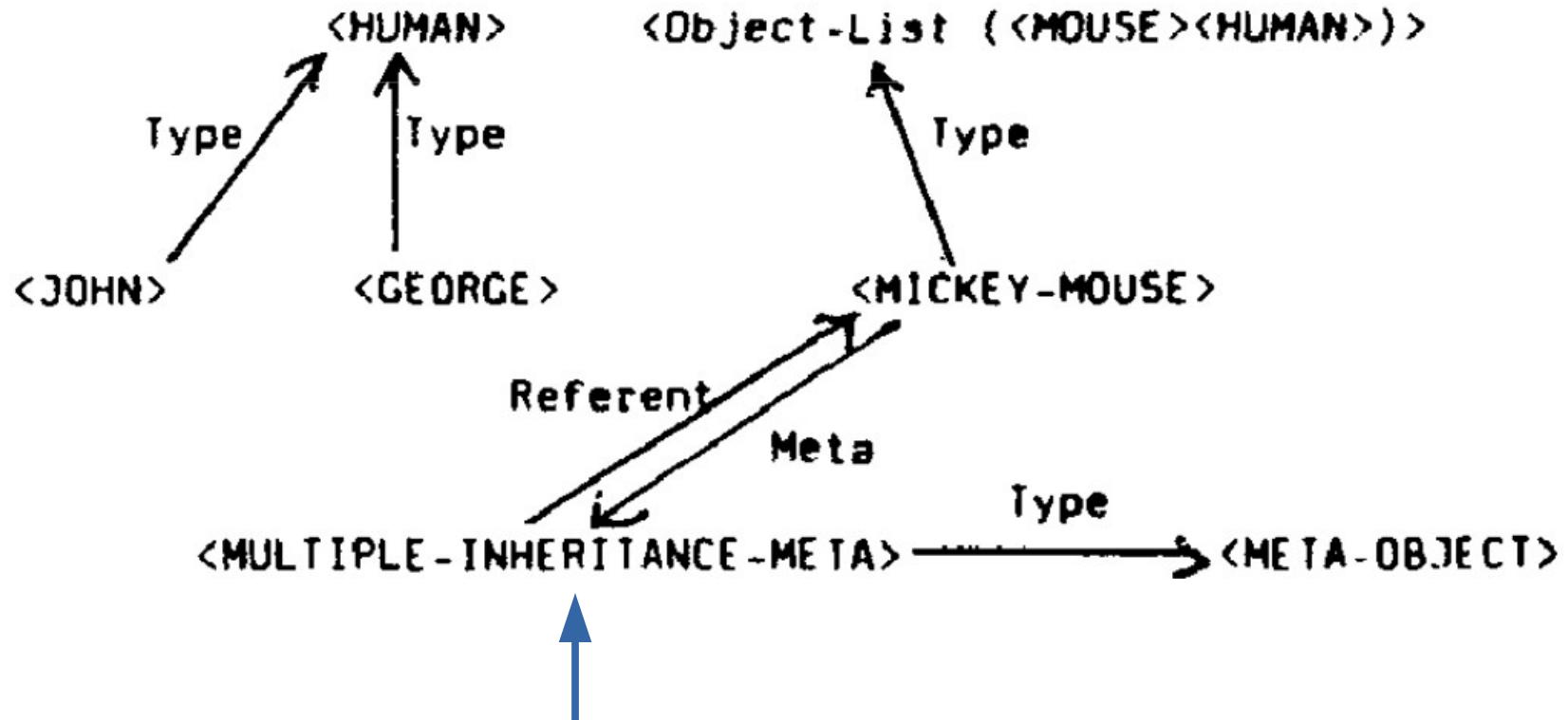
Novo estilo de programação

Exemplo:
desvio de
execução
para
suportar
herança
múltipla



Novo estilo de programação

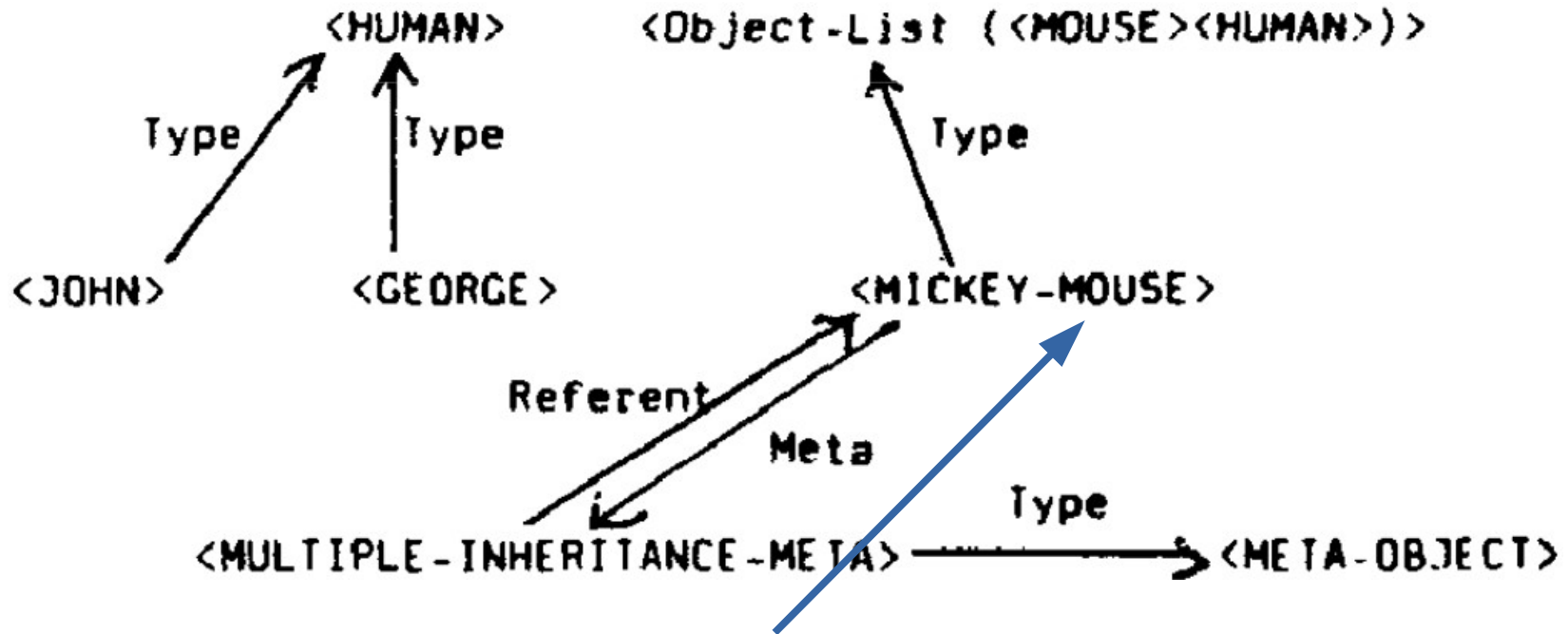
Exemplo:
desvio de
execução
para
suportar
herança
múltipla



Esse novo "meta" sabe tratar objetos diferentes

Novo estilo de programação

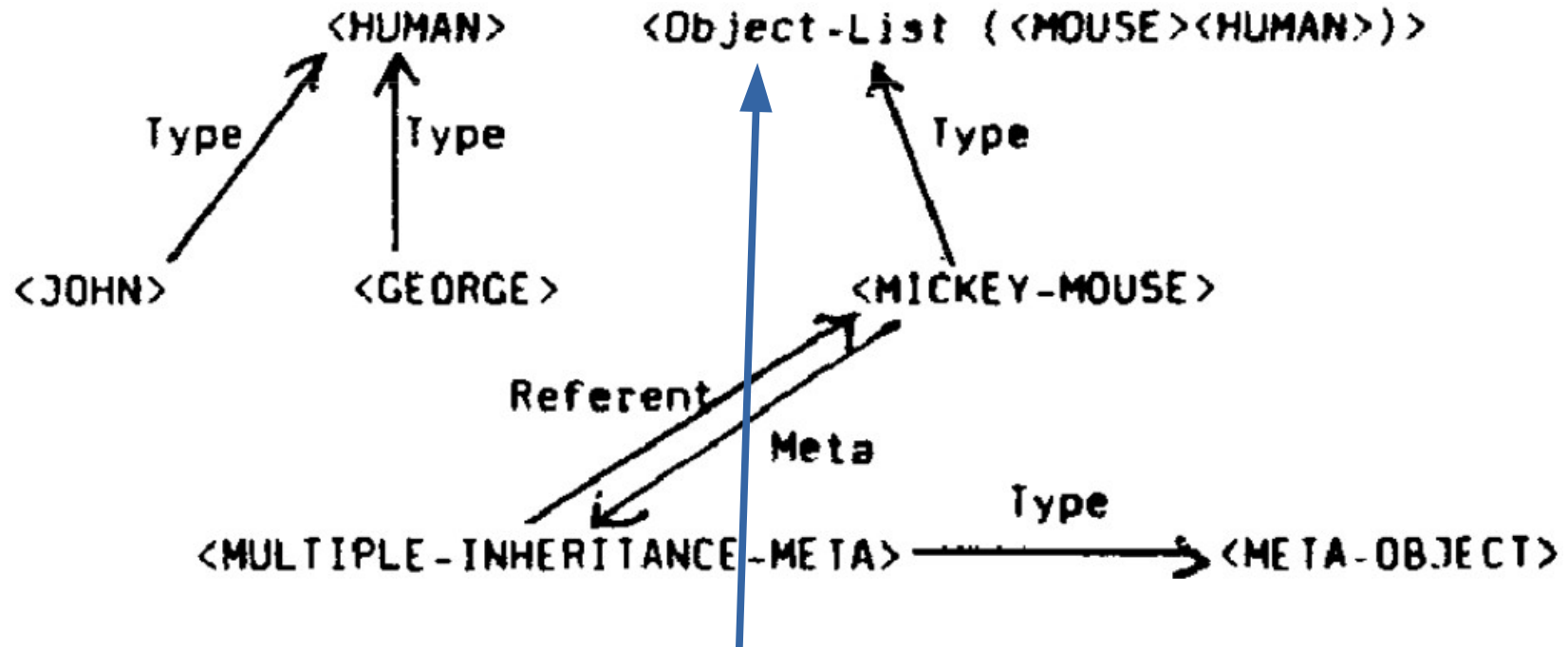
Exemplo:
desvio de
execução
para
suportar
herança
múltipla



Mickey-mouse é implementado
por esse novo "meta"

Novo estilo de programação

Exemplo:
desvio de
execução
para
suportar
herança
múltipla



Pode-se, então, fazer o objeto ser formado
por uma lista de objetos diferentes

Novo estilo de programação

3-KRS contém biblioteca de variações mais utilizadas: impressão amigável, rastreamento, herança múltipla, etc.

Novo estilo de programação

3-KRS contém biblioteca de variações mais utilizadas: impressão amigável, rastreamento, herança múltipla, etc.

*Reflexão local pode ser feita em objetos **específicos** (Exemplos: John, uma mensagem particular, etc)* ← instância

classe → *Reflexão geral pode ser feita em objetos **abstratos** (Exemplo: classe Pessoa)*

FIM



- Obrigado!
- Cada um fala sobre 1 conceito que achou impressionante