

Mecanismos Reflexivos em Smalltalk

Daniel Ribeiro

Seções

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

1 Introdução

2 Um Pouco de Smalltalk

3 Alterando a Resolução de Métodos

4 Mecanismos Reflexivos

5 Aplicações

6 Conclusões e Bibliografia

Seções

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

1 Introdução

2 Um Pouco de Smalltalk

3 Alterando a Resolução de Métodos

4 Mecanismos Reflexivos

5 Aplicações

6 Conclusões e Bibliografia

Seções

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

1 Introdução

2 Um Pouco de Smalltalk

3 Alterando a Resolução de Métodos

4 Mecanismos Reflexivos

5 Aplicações

6 Conclusões e Bibliografia

Seções

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- 1 Introdução
- 2 Um Pouco de Smalltalk
- 3 Alterando a Resolução de Métodos
- 4 Mecanismos Reflexivos
- 5 Aplicações
- 6 Conclusões e Bibliografia

Seções

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- 1 Introdução
- 2 Um Pouco de Smalltalk
- 3 Alterando a Resolução de Métodos
- 4 Mecanismos Reflexivos
- 5 Aplicações
- 6 Conclusões e Bibliografia

Seções

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- 1 Introdução
- 2 Um Pouco de Smalltalk
- 3 Alterando a Resolução de Métodos
- 4 Mecanismos Reflexivos
- 5 Aplicações
- 6 Conclusões e Bibliografia

Definição Formal

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

Reflection is the ability of a program to manipulate as data something representing the state of the program during its own execution. There are two aspects of such manipulation : introspection and intercession. Introspection is the ability of a program to observe and therefore reason about its own state. Intercession is the ability of a program to modify its own execution state or alter its own interpretation or meaning. Both aspects require a mechanism for encoding execution state as data; providing such an encoding is called reification.

Conceitos Iniciais

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Programas reflexivos sabem sobre si mesmos.
- Arquiteturas procedurais completamente reflexivas: onde é possível que um processo acesse e manipule uma representação completa e explícita de seu próprio estado, de modo que toda alteração é percebida imediatamente no estado e comportamento do processo.
- Metaobjeto: contém informação descrevendo o objeto referente

Conceitos Iniciais

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Programas reflexivos sabem sobre si mesmos.
- Arquiteturas procedurais completamente reflexivas: onde é possível que um processo acesse e manipule uma representação completa e explícita de seu próprio estado, de modo que toda alteração é percebida imediatamente no estado e comportamento do processo.
- Metaobjeto: contém informação descrevendo o objeto referente

Conceitos Iniciais

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Programas reflexivos sabem sobre si mesmos.
- Arquiteturas procedurais completamente reflexivas: onde é possível que um processo acesse e manipule uma representação completa e explícita de seu próprio estado, de modo que toda alteração é percebida imediatamente no estado e comportamento do processo.
- Metaobjeto: contém informação descrevendo o objeto referente

Motivação

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Reflexão facilita na implementação de muitas aplicações
- Arquiteturas procedurais completamente reflexivas são difíceis de implementar eficientemente
- É possível implementar de modo eficiente *mecanismos reflexivos*

Motivação

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Reflexão facilita na implementação de muitas aplicações
- Arquiteturas procedurais completamente reflexivas são difíceis de implementar eficientemente
- É possível implementar de modo eficiente *mecanismos reflexivos*

Motivação

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Reflexão facilita na implementação de muitas aplicações
- Arquiteturas procedurais completamente reflexivas são difíceis de implementar eficientemente
- É possível implementar de modo eficiente *mecanismos reflexivos*

Smalltalk-80: De onde partimos?

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- **Metaclasses: classe dos objetos do tipo classe**
 - A pilha de execução é um objeto (variável *thisContext*)
 - Inspeção das variáveis de um objeto
 - Alteração da classe de um objeto para a de outro
 - Troca simultânea das referências de dois objetos (*become*)
 - Lista de que objetos têm referência para um objeto

Smalltalk-80: De onde partimos?

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Metaclasses: classe dos objetos do tipo classe
- A pilha de execução é um objeto (variável *thisContext*)
 - Inspeção das variáveis de um objeto
 - Alteração da classe de um objeto para a de outro
 - Troca simultânea das referências de dois objetos (*become*)
 - Lista de que objetos têm referência para um objeto

Smalltalk-80: De onde partimos?

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Metaclasses: classe dos objetos do tipo classe
- A pilha de execução é um objeto (variável *thisContext*)
- Inspeção das variáveis de um objeto
 - Alteração da classe de um objeto para a de outro
 - Troca simultânea das referências de dois objetos (*become*)
 - Lista de que objetos têm referência para um objeto

Smalltalk-80: De onde partimos?

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Metaclasses: classe dos objetos do tipo classe
- A pilha de execução é um objeto (variável *thisContext*)
- Inspeção das variáveis de um objeto
- Alteração da classe de um objeto para a de outro
 - Troca simultânea das referências de dois objetos (*become*)
 - Lista de que objetos têm referência para um objeto

Smalltalk-80: De onde partimos?

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Metaclasses: classe dos objetos do tipo classe
- A pilha de execução é um objeto (variável *thisContext*)
- Inspeção das variáveis de um objeto
- Alteração da classe de um objeto para a de outro
- Troca simultânea das referências de dois objetos (*become*)
- Lista de que objetos têm referência para um objeto

Smalltalk-80: De onde partimos?

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Metaclasses: classe dos objetos do tipo classe
- A pilha de execução é um objeto (variável *thisContext*)
- Inspeção das variáveis de um objeto
- Alteração da classe de um objeto para a de outro
- Troca simultânea das referências de dois objetos (*become*)
- Lista de que objetos têm referência para um objeto

Fundamentos de Smalltalk

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Tudo é um objeto: números, nil, true, false, classes, a pilha de execução, etc
- Referências a variáveis podem apontar para qualquer objeto, independente da classe (não é fortemente tipada)
- Chamadas de métodos: também são chamados de envio de mensagens
- *Receiver* recebe a mensagem, e *selector* é o nome da mensagem
- Geralmente implementado com uma máquina virtual, que interpreta byte-code
- Sem herança múltipla
- Classe raiz: *Object*

Fundamentos de Smalltalk

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Tudo é um objeto: números, nil, true, false, classes, a pilha de execução, etc
- Referências a variáveis podem apontar para qualquer objeto, independente da classe (não é fortemente tipada)
- Chamadas de métodos: também são chamados de envio de mensagens
- *Receiver* recebe a mensagem, e *selector* é o nome da mensagem
- Geralmente implementado com uma máquina virtual, que interpreta byte-code
- Sem herança múltipla
- Classe raiz: *Object*

Fundamentos de Smalltalk

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Tudo é um objeto: números, nil, true, false, classes, a pilha de execução, etc
- Referências a variáveis podem apontar para qualquer objeto, independente da classe (não é fortemente tipada)
- Chamadas de métodos: também são chamados de envio de mensagens
 - *Receiver* recebe a mensagem, e *selector* é o nome da mensagem
 - Geralmente implementado com uma máquina virtual, que interpreta byte-code
 - Sem herança múltipla
 - Classe raiz: *Object*

Fundamentos de Smalltalk

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Tudo é um objeto: números, nil, true, false, classes, a pilha de execução, etc
- Referências a variáveis podem apontar para qualquer objeto, independente da classe (não é fortemente tipada)
- Chamadas de métodos: também são chamados de envio de mensagens
- *Receiver* recebe a mensagem, e *selector* é o nome da mensagem
- Geralmente implementado com uma máquina virtual, que interpreta byte-code
- Sem herança múltipla
- Classe raiz: *Object*

Fundamentos de Smalltalk

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Tudo é um objeto: números, nil, true, false, classes, a pilha de execução, etc
- Referências a variáveis podem apontar para qualquer objeto, independente da classe (não é fortemente tipada)
- Chamadas de métodos: também são chamados de envio de mensagens
- *Receiver* recebe a mensagem, e *selector* é o nome da mensagem
- Geralmente implementado com uma máquina virtual, que interpreta byte-code
- Sem herança múltipla
- Classe raiz: *Object*

Fundamentos de Smalltalk

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Tudo é um objeto: números, nil, true, false, classes, a pilha de execução, etc
- Referências a variáveis podem apontar para qualquer objeto, independente da classe (não é fortemente tipada)
- Chamadas de métodos: também são chamados de envio de mensagens
- *Receiver* recebe a mensagem, e *selector* é o nome da mensagem
- Geralmente implementado com uma máquina virtual, que interpreta byte-code
- Sem herança múltipla
- Classe raiz: *Object*

Fundamentos de Smalltalk

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Tudo é um objeto: números, nil, true, false, classes, a pilha de execução, etc
- Referências a variáveis podem apontar para qualquer objeto, independente da classe (não é fortemente tipada)
- Chamadas de métodos: também são chamados de envio de mensagens
- *Receiver* recebe a mensagem, e *selector* é o nome da mensagem
- Geralmente implementado com uma máquina virtual, que interpreta byte-code
- Sem herança múltipla
- Classe raiz: *Object*

Resolução de Métodos

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Pode-se enviar qualquer mensagem para qualquer objeto
 - Se classe do objeto em questão implementa o método invocado, ele é executado.
 - Caso contrário, busca-se na super classe da classe do objeto. Se ela também não implementa o método, vai-se subindo na hierarquia de classes.
 - Se mensagem não é encontrada, é enviada a mensagem *doesNotUnderstand* para o objeto.

Resolução de Métodos

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Pode-se enviar qualquer mensagem para qualquer objeto
- Se classe do objeto em questão implementa o método invocado, ele é executado.
- Caso contrário, busca-se na super classe da classe do objeto. Se ela também não implementa o método, vai-se subindo na hierarquia de classes.
- Se mensagem não é encontrada, é enviada a mensagem *doesNotUnderstand* para o objeto.

Resolução de Métodos

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Pode-se enviar qualquer mensagem para qualquer objeto
- Se classe do objeto em questão implementa o método invocado, ele é executado.
- Caso contrário, busca-se na super classe da classe do objeto. Se ela também não implementa o método, vai-se subindo na hierarquia de classes.
- Se mensagem não é encontrada, é enviada a mensagem *doesNotUnderstand* para o objeto.

Resolução de Métodos

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Pode-se enviar qualquer mensagem para qualquer objeto
- Se classe do objeto em questão implementa o método invocado, ele é executado.
- Caso contrário, busca-se na super classe da classe do objeto. Se ela também não implementa o método, vai-se subindo na hierarquia de classes.
- Se mensagem não é encontrada, é enviada a mensagem *doesNotUnderstand* para o objeto.

O que Podemos Mudar

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

Leitura e Escrita de Variáveis

- Métodos *setters* and *getters*: altera-se mecanismos de envio de mensagens.
- Encapsulamento de variáveis em *variáveis ativas*: as referências se tornam um objeto em si, e elas controlam o acesso aos valores das variáveis através de mensagens.

Mensagens

- Interceptação de mensagens enviadas **por** um objeto.
- Interceptação de mensagens enviadas **à** um objeto.
mensagens.

O que Podemos Mudar

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

Leitura e Escrita de Variáveis

- Métodos *setters* and *getters*: altera-se mecanismos de envio de mensagens.
- Encapsulamento de variáveis em *variáveis ativas*: as referências se tornam um objeto em si, e elas controlam o acesso aos valores das variáveis através de mensagens.

Mensagens

- Interceptação de mensagens enviadas **por** um objeto.
- Interceptação de mensagens enviadas **à** um objeto.

O que Podemos Mudar

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

Leitura e Escrita de Variáveis

- Métodos *setters* and *getters*: altera-se mecanismos de envio de mensagens.
- Encapsulamento de variáveis em *variáveis ativas*: as referências se tornam um objeto em si, e elas controlam o acesso aos valores das variáveis através de mensagens.

Mensagens

- Interceptação de mensagens enviadas por um objeto.
 - Interceptação de mensagens enviadas à um objeto.
- mensagens.

O que Podemos Mudar

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

Leitura e Escrita de Variáveis

- Métodos *setters* and *getters*: altera-se mecanismos de envio de mensagens.
- Encapsulamento de variáveis em *variáveis ativas*: as referências se tornam um objeto em si, e elas controlam o acesso aos valores das variáveis através de mensagens.

Mensagens

- Interceptação de mensagens enviadas **por** um objeto.
- Interceptação de mensagens enviadas *à* um objeto. mensagens.

O que Podemos Mudar

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

Leitura e Escrita de Variáveis

- Métodos *setters* and *getters*: altera-se mecanismos de envio de mensagens.
- Encapsulamento de variáveis em *variáveis ativas*: as referências se tornam um objeto em si, e elas controlam o acesso aos valores das variáveis através de mensagens.

Mensagens

- Interceptação de mensagens enviadas **por** um objeto.
- Interceptação de mensagens enviadas **à** um objeto. mensagens.

O que Podemos Mudar

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

Valor Retornado

Se um objeto intercepta mensagens enviadas a um outro objeto, ele também deve poder alterar o resultado da mensagem interceptada.

Conclusão

Alterar o algoritmo de resolução de métodos permite implementar as alterações acima.

O que Podemos Mudar

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

Valor Retornado

Se um objeto intercepta mensagens enviadas a um outro objeto, ele também deve poder alterar o resultado da mensagem interceptada.

Conclusão

Alterar o algoritmo de resolução de métodos permite implementar as alterações acima.

O que Podemos Mudar

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

Valor Retornado

Se um objeto intercepta mensagens enviadas a um outro objeto, ele também deve poder alterar o resultado da mensagem interceptada.

Conclusão

Alterar o algoritmo de resolução de métodos permite implementar as alterações acima.

O que Podemos Mudar

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

Valor Retornado

Se um objeto intercepta mensagens enviadas a um outro objeto, ele também deve poder alterar o resultado da mensagem interceptada.

Conclusão

Alterar o algoritmo de resolução de métodos permite implementar as alterações acima.

Proposta Inicial: *doesNotUnderstand*

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Idéia: *wrapper* que implementa apenas *doesNotUnderstand*
 - Mensagens são repassadas ao objeto encapsulado, possivelmente alteradas
 - Resultados recebidos do objeto encapsulado são devolvidos, possivelmente alterados
 - Problema 1: referência do objeto sai do contexto ⇒ quebrando de encapsulamento
 - Problema 2: *doesNotUnderstand* é ineficiente

Proposta Inicial: *doesNotUnderstand*

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Idéia: *wrapper* que implementa apenas *doesNotUnderstand*
- Mensagens são repassadas ao objeto encapsulado, possivelmente alteradas
- Resultados recebidos do objeto encapsulado são devolvidos, possivelmente alterados
- Problema 1: referência do objeto sai do contexto ⇒ quebrando de encapsulamento
- Problema 2: *doesNotUnderstand* é ineficiente

Proposta Inicial: *doesNotUnderstand*

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Idéia: *wrapper* que implementa apenas *doesNotUnderstand*
- Mensagens são repassadas ao objeto encapsulado, possivelmente alteradas
- Resultados recebidos do objeto encapsulado são devolvidos, possivelmente alterados
- Problema 1: referência do objeto sai do contexto ⇒ quebrando de encapsulamento
- Problema 2: *doesNotUnderstand* é ineficiente

Proposta Inicial: *doesNotUnderstand*

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Idéia: *wrapper* que implementa apenas *doesNotUnderstand*
- Mensagens são repassadas ao objeto encapsulado, possivelmente alteradas
- Resultados recebidos do objeto encapsulado são devolvidos, possivelmente alterados
- Problema 1: referência do objeto sai do contexto ⇒ quebrando de encapsulamento
- Problema 2: *doesNotUnderstand* é ineficiente

Proposta Inicial: *doesNotUnderstand*

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Idéia: *wrapper* que implementa apenas *doesNotUnderstand*
- Mensagens são repassadas ao objeto encapsulado, possivelmente alteradas
- Resultados recebidos do objeto encapsulado são devolvidos, possivelmente alterados
- Problema 1: referência do objeto sai do contexto \Rightarrow quebrando de encapsulamento
- Problema 2: *doesNotUnderstand* é ineficiente

Outra proposta: Resoluções personalizadas

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Idéia: *in-line caching*
 - Técnica: é feito cache da última classe e do último método usados
 - Alteração é feita no próprio código
 - Justificativa: localidade da execução de código
 - Vantagens: muito eficiente em 95% do casos

Outra proposta: Resoluções personalizadas

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Idéia: *in-line caching*
- Técnica: é feito cache da última classe e do último método usados
 - Alteração é feita no próprio código
 - Justificativa: localidade da execução de código
 - Vantagens: muito eficiente em 95% do casos

Outra proposta: Resoluções personalizadas

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Idéia: *in-line caching*
- Técnica: é feito cache da última classe e do último método usados
- Alteração é feita no próprio código
 - Justificativa: localidade da execução de código
 - Vantagens: muito eficiente em 95% do casos

Outra proposta: Resoluções personalizadas

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Idéia: *in-line caching*
- Técnica: é feito cache da última classe e do último método usados
- Alteração é feita no próprio código
- Justificativa: localidade da execução de código
- Vantagens: muito eficiente em 95% do casos

Outra proposta: Resoluções personalizadas

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Idéia: *in-line caching*
- Técnica: é feito cache da última classe e do último método usados
- Alteração é feita no próprio código
- Justificativa: localidade da execução de código
- Vantagens: muito eficiente em 95% do casos

Resoluções personalizadas: Sem cache

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

```
push      receiver
push      arg1,...,argn
call      unlinkedDispatch
<message selector>
<unused>
;Resultado aqui...
```

Resoluções personalizadas: Sem cache

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

```
push      receiver
push      arg1,...,argn
call      unlinkedDispatch
<message selector>
<unused>
;Resultado aqui...
```

Resoluções personalizadas: Sem cache

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

```
push      receiver
push      arg1,...,argn
call      unlinkedDispatch
<message selector>
<unused>
;Resultado aqui...
```

Resoluções personalizadas: Sem cache

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

```
push      receiver
push      arg1,...,argn
call      unlinkedDispatch
<message selector>
<unused>
;Resultado aqui...
```

Resoluções personalizadas: Sem cache

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

```
push      receiver
push      arg1,...,argn
call      unlinkedDispatch
<message selector>
<unused>
;Resultado aqui...
```

Resoluções personalizadas: Sem cache

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

```
push      receiver
push      arg1,...,argn
call      unlinkedDispatch
<message selector>
<unused>
;Resultado aqui...
```

Resoluções personalizadas: *unlinkedDispatch*

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Quando invocado, faz resolução de métodos normal
- Associa o método invocado com a classe do objeto e o seletor da mensagem
- Altera o código da resolução de métodos para...

Resoluções personalizadas: *unlinkedDispatch*

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Quando invocado, faz resolução de métodos normal
- Associa o método invocado com a classe do objeto e o seletor da mensagem
- Altera o código da resolução de métodos para...

Resoluções personalizadas: *unlinkedDispatch*

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Quando invocado, faz resolução de métodos normal
- Associa o método invocado com a classe do objeto e o seletor da mensagem
- Altera o código da resolução de métodos para...

Resoluções personalizadas: Comparação

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

Com Cache

```
push receiver  
push arg1,...,argn  
call linkedDispatch  
last class  
method  
;Resultado aqui...
```

Sem Cache

```
push receiver  
push arg1,...,argn  
call unlinkedDispatch  
<message selector>  
<unused>  
;Resultado aqui...
```

Resoluções personalizadas: Comparação

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

Com Cache

```
push receiver
push arg1,...,argn
call linkedDispatch
last class
method
;Resultado aqui...
```

Sem Cache

```
push receiver
push arg1,...,argn
call unlinkedDispatch
<message selector>
<unused>
;Resultado aqui...
```

Resoluções personalizadas: Comparação

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

Com Cache

```
push receiver
push arg1,...,argn
call linkedDispatch
last class
method
;Resultado aqui...
```

Sem Cache

```
push receiver
push arg1,...,argn
call unlinkedDispatch
<message selector>
<unused>
;Resultado aqui...
```

Resoluções personalizadas: Comparação

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

Com Cache

```
push receiver
push arg1,...,argn
call linkedDispatch
last class
method
;Resultado aqui...
```

Sem Cache

```
push receiver
push arg1,...,argn
call unlinkedDispatch
<message selector>
<unused>
;Resultado aqui...
```

Resoluções personalizadas: Comparação

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

Com Cache

```
push receiver
push arg1,...,argn
call linkedDispatch
last class
method
;Resultado aqui...
```

Sem Cache

```
push receiver
push arg1,...,argn
call unlinkedDispatch
<message selector>
<unused>
;Resultado aqui...
```

Resoluções personalizadas: Comparação

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

Com Cache

```
push receiver  
push arg1,...,argn  
call linkedDispatch  
last class  
method  
;Resultado aqui...
```

Sem Cache

```
push receiver  
push arg1,...,argn  
call unlinkedDispatch  
<message selector>  
<unused>  
;Resultado aqui...
```

Resoluções personalizadas: *linkedDispatch*

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Resolução alterada é dita *linkada*
 - Se classe e métodos forem usados novamente, ele é executado diretamente
 - Se houver algum conflito com o cache, a resolução completa de métodos é realizada

Resoluções personalizadas: *linkedDispatch*

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Resolução alterada é dita *linkada*
- Se classe e métodos forem usados novamente, ele é executado diretamente
- Se houver algum conflito com o cache, a resolução completa de métodos é realizada

Resoluções personalizadas: *linkedDispatch*

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- Resolução alterada é dita *linkada*
- Se classe e métodos forem usados novamente, ele é executado diretamente
- Se houver algum conflito com o cache, a resolução completa de métodos é realizada

Mecanismos Reflexivos

Mecanismos Reflexivos em Smalltalk

Daniel Ribeiro

Introdução

Um Pouco de Smalltalk

Alterando a Resolução de Métodos

Mecanismos Reflexivos

Aplicações

Conclusões e Bibliografia

- Classes de *dispatch*: tipo de classe na qual todos os métodos que são invocados são passados para *dispatchMessage: aMessage* (como em *doesNotUnderstand*, mas é a primeira coisa a ser feita, e não a última).
- Classes sem pai: semelhante à idéia acima, mas primeiro olha os métodos da classe antes de invocar *dispatchMessage*. Efetivamente é o equivalente a remover a superclasse do mecanismo de resolução de métodos.

Mecanismos Reflexivos

Mecanismos Reflexivos em Smalltalk

Daniel Ribeiro

Introdução

Um Pouco de Smalltalk

Alterando a Resolução de Métodos

Mecanismos Reflexivos

Aplicações

Conclusões e Bibliografia

- Classes de *dispatch*: tipo de classe na qual todos os métodos que são invocados são passados para *dispatchMessage: aMessage* (como em *doesNotUnderstand*, mas é a primeira coisa a ser feita, e não a última).
- Classes sem pai: semelhante à idéia acima, mas primeiro olha os métodos da classe antes de invocar *dispatchMessage*. Efetivamente é o equivalente a remover a superclasse do mecanismo de resolução de métodos.

Metaobjetos em Smalltalk

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- São classes de um objeto só que fazem apenas referência ao objeto que encapsulam
 - Podem ser introduzidos e removidos dinamicamente
 - Transparente para o resto do sistema
 - Não podem ser referenciados diretamente
 - Implementar com *doesNotUnderstand* não é eficiente
 - As classes dos metaobjetos são *sem pai*, ou dispatching classes
 - Implementar com resolução de métodos personalizada com *in-line caching* torna eficiente

Metaobjetos em Smalltalk

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- São classes de um objeto só que fazem apenas referência ao objeto que encapsulam
- Podem ser introduzidos e removidos dinamicamente
 - Transparente para o resto do sistema
 - Não podem ser referenciados diretamente
 - Implementar com *doesNotUnderstand* não é eficiente
 - As classes dos metaobjetos são *sem pai*, ou dispatching classes
 - Implementar com resolução de métodos personalizada com *in-line caching* torna eficiente

Metaobjetos em Smalltalk

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- São classes de um objeto só que fazem apenas referência ao objeto que encapsulam
- Podem ser introduzidos e removidos dinamicamente
- Transparente para o resto do sistema
 - Não podem ser referenciados diretamente
 - Implementar com *doesNotUnderstand* não é eficiente
 - As classes dos metaobjetos são *sem pai*, ou dispatching classes
 - Implementar com resolução de métodos personalizada com *in-line caching* torna eficiente

Metaobjetos em Smalltalk

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- São classes de um objeto só que fazem apenas referência ao objeto que encapsulam
- Podem ser introduzidos e removidos dinamicamente
- Transparente para o resto do sistema
- Não podem ser referenciados diretamente
 - Implementar com *doesNotUnderstand* não é eficiente
 - As classes dos metaobjetos são *sem pai*, ou dispatching classes
 - Implementar com resolução de métodos personalizada com *in-line caching* torna eficiente

Metaobjetos em Smalltalk

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- São classes de um objeto só que fazem apenas referência ao objeto que encapsulam
- Podem ser introduzidos e removidos dinamicamente
- Transparente para o resto do sistema
- Não podem ser referenciados diretamente
- Implementar com *doesNotUnderstand* não é eficiente
 - As classes dos metaobjetos são *sem pai*, ou dispatching classes
 - Implementar com resolução de métodos personalizada com *in-line caching* torna eficiente

Metaobjetos em Smalltalk

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- São classes de um objeto só que fazem apenas referência ao objeto que encapsulam
- Podem ser introduzidos e removidos dinamicamente
- Transparente para o resto do sistema
- Não podem ser referenciados diretamente
- Implementar com *doesNotUnderstand* não é eficiente
- As classes dos metaobjetos são *sem pai*, ou dispatching classes
- Implementar com resolução de métodos personalizada com *in-line caching* torna eficiente

Metaobjetos em Smalltalk

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia

- São classes de um objeto só que fazem apenas referência ao objeto que encapsulam
- Podem ser introduzidos e removidos dinamicamente
- Transparente para o resto do sistema
- Não podem ser referenciados diretamente
- Implementar com *doesNotUnderstand* não é eficiente
- As classes dos metaobjetos são *sem pai*, ou dispatching classes
- Implementar com resolução de métodos personalizada com *in-line caching* torna eficiente

Mecanismos Reflexivos: Exemplo

Mecanismos Reflexivos em Smalltalk

Daniel Ribeiro

Introdução

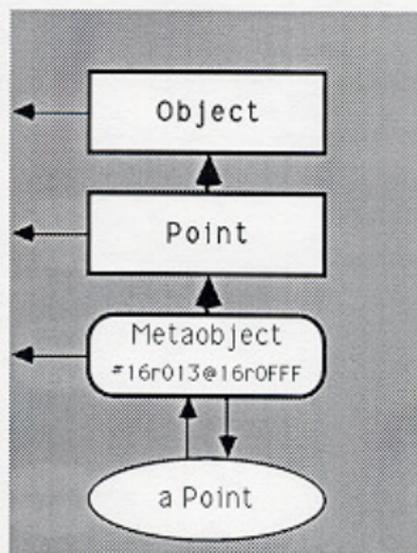
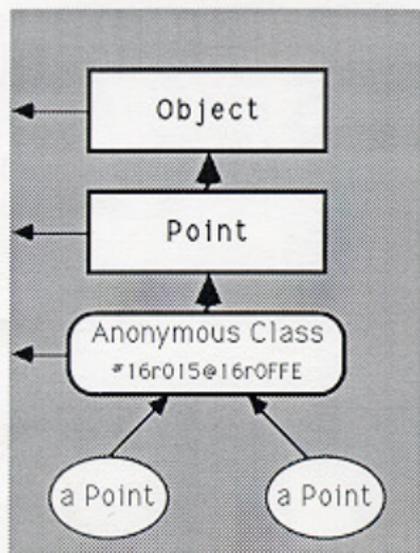
Um Pouco de Smalltalk

Alterando a Resolução de Métodos

Mecanismos Reflexivos

Aplicações

Conclusões e Bibliografia



- **Objetos atômicos**
- Futures
- Campos dinâmicos
- Proteção dinâmica
- Múltiplas visões de um objeto
- Adaptadores de protocolo dinâmicos

- **Objetos atômicos**
- **Futures**
- Campos dinâmicos
- Proteção dinâmica
- Múltiplas visões de um objeto
- Adaptadores de protocolo dinâmicos

- **Objetos atômicos**
- **Futures**
- **Campos dinâmicos**
 - Proteção dinâmica
 - Múltiplas visões de um objeto
 - Adaptadores de protocolo dinâmicos

- Objetos atômicos
- Futures
- Campos dinâmicos
- Proteção dinâmica
 - Múltiplas visões de um objeto
 - Adaptadores de protocolo dinâmicos

- Objetos atômicos
- Futures
- Campos dinâmicos
- Proteção dinâmica
- Múltiplas visões de um objeto
- Adaptadores de protocolo dinâmicos

- Objetos atômicos
- Futures
- Campos dinâmicos
- Proteção dinâmica
- Múltiplas visões de um objeto
- Adaptadores de protocolo dinâmicos

Conclusões

Mecanismos Reflexivos em Smalltalk

Daniel Ribeiro

Introdução

Um Pouco de Smalltalk

Alterando a Resolução de Métodos

Mecanismos Reflexivos

Aplicações

Conclusões e Bibliografia

- Reflexão é um framework que permite tratar facilmente alguns problemas
- Sobrescrever o método de resolução de métodos fornece grande parte do poder da reflexão
- Mecanismos reflexivos não são obrigatoriamente ineficientes
- Os mecanismos reflexivos podem ser utilizados com pouco impacto no código de um programa
- Smalltalk-80 não é muito adequada para programação paralela e distribuída, mas com reflexão ela pode ser estendida para atender tais requisitos
- Reflexão permite sistemas complexos se adaptarem a novos requisitos ao permitir objetos se adequarem à novas situações

Conclusões

Mecanismos Reflexivos em Smalltalk

Daniel Ribeiro

Introdução

Um Pouco de Smalltalk

Alterando a Resolução de Métodos

Mecanismos Reflexivos

Aplicações

Conclusões e Bibliografia

- Reflexão é um framework que permite tratar facilmente alguns problemas
- Sobrescrever o método de resolução de métodos fornece grande parte do poder da reflexão
- Mecanismos reflexivos não são obrigatoriamente ineficientes
- Os mecanismos reflexivos podem ser utilizados com pouco impacto no código de um programa
- Smalltalk-80 não é muito adequada para programação paralela e distribuída, mas com reflexão ela pode ser estendida para atender tais requisitos
- Reflexão permite sistemas complexos se adaptarem a novos requisitos ao permitir objetos se adequarem à novas situações

Conclusões

Mecanismos Reflexivos em Smalltalk

Daniel Ribeiro

Introdução

Um Pouco de Smalltalk

Alterando a Resolução de Métodos

Mecanismos Reflexivos

Aplicações

Conclusões e Bibliografia

- Reflexão é um framework que permite tratar facilmente alguns problemas
- Sobrescrever o método de resolução de métodos fornece grande parte do poder da reflexão
- Mecanismos reflexivos não são obrigatoriamente ineficientes
- Os mecanismos reflexivos podem ser utilizados com pouco impacto no código de um programa
- Smalltalk-80 não é muito adequada para programação paralela e distribuída, mas com reflexão ela pode ser estendida para atender tais requisitos
- Reflexão permite sistemas complexos se adaptarem a novos requisitos ao permitir objetos se adequarem à novas situações

Conclusões

Mecanismos Reflexivos em Smalltalk

Daniel Ribeiro

Introdução

Um Pouco de Smalltalk

Alterando a Resolução de Métodos

Mecanismos Reflexivos

Aplicações

Conclusões e Bibliografia

- Reflexão é um framework que permite tratar facilmente alguns problemas
- Sobrescrever o método de resolução de métodos fornece grande parte do poder da reflexão
- Mecanismos reflexivos não são obrigatoriamente ineficientes
- Os mecanismos reflexivos podem ser utilizados com pouco impacto no código de um programa
- Smalltalk-80 não é muito adequada para programação paralela e distribuída, mas com reflexão ela pode ser estendida para atender tais requisitos
- Reflexão permite sistemas complexos se adaptarem a novos requisitos ao permitir objetos se adequarem à novas situações

Conclusões

Mecanismos Reflexivos em Smalltalk

Daniel Ribeiro

Introdução

Um Pouco de Smalltalk

Alterando a Resolução de Métodos

Mecanismos Reflexivos

Aplicações

Conclusões e Bibliografia

- Reflexão é um framework que permite tratar facilmente alguns problemas
- Sobrescrever o método de resolução de métodos fornece grande parte do poder da reflexão
- Mecanismos reflexivos não são obrigatoriamente ineficientes
- Os mecanismos reflexivos podem ser utilizados com pouco impacto no código de um programa
- Smalltalk-80 não é muito adequada para programação paralela e distribuída, mas com reflexão ela pode ser estendida para atender tais requisitos
- Reflexão permite sistemas complexos se adaptarem a novos requisitos ao permitir objetos se adequarem à novas situações

Conclusões

Mecanismos Reflexivos em Smalltalk

Daniel Ribeiro

Introdução

Um Pouco de Smalltalk

Alterando a Resolução de Métodos

Mecanismos Reflexivos

Aplicações

Conclusões e Bibliografia

- Reflexão é um framework que permite tratar facilmente alguns problemas
- Sobrescrever o método de resolução de métodos fornece grande parte do poder da reflexão
- Mecanismos reflexivos não são obrigatoriamente ineficientes
- Os mecanismos reflexivos podem ser utilizados com pouco impacto no código de um programa
- Smalltalk-80 não é muito adequada para programação paralela e distribuída, mas com reflexão ela pode ser estendida para atender tais requisitos
- Reflexão permite sistemas complexos se adaptarem a novos requisitos ao permitir objetos se adequarem à novas situações

Bibliografia

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia



Brian Foote and Ralph E. Johnson.
Reflective facilities in smalltalk-80.

In Norman Meyrowitz, editor, *Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA)*, volume 24, pages 327–336, New York, NY, 1989. ACM Press.



Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides.

Design patterns: elements of reusable object-oriented software.

Addison-Wesley Professional, 1995.

Bibliografia

Mecanismos
Reflexivos
em Smalltalk

Daniel
Ribeiro

Introdução

Um Pouco
de Smalltalk

Alterando a
Resolução de
Métodos

Mecanismos
Reflexivos

Aplicações

Conclusões e
Bibliografia



Fred Rivard.

Smalltalk: a reflective language.

Technical report, Laboratoire Jules Verne, Ecole des Mines de Nantes & Object Technology International Inc., France, October 1996.



Daniel G. Bobrow, Richard P. Gabriel, and Jon L. White.

Clos in context: the shape of the design space.
pages 29–61, 1993.