Finden statt Suchen:

Der Method-Finder wird abgesichert

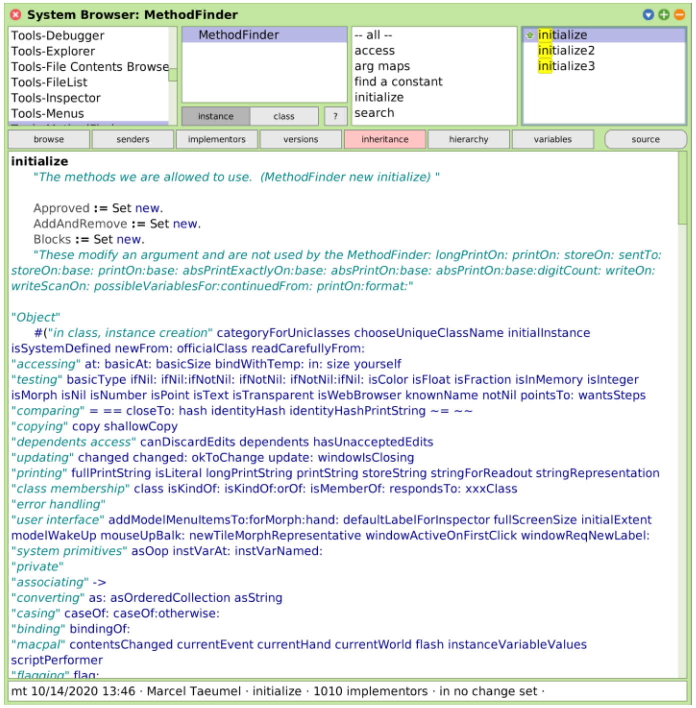
Christoph Thiede

Squeak Demos ’22

2022-11-19

State of the art – Einschränkungen

- Whitelist von Selektoren
- Manuelle Pflege notwendig
- unbekannte Domänen nicht durchsuchbar

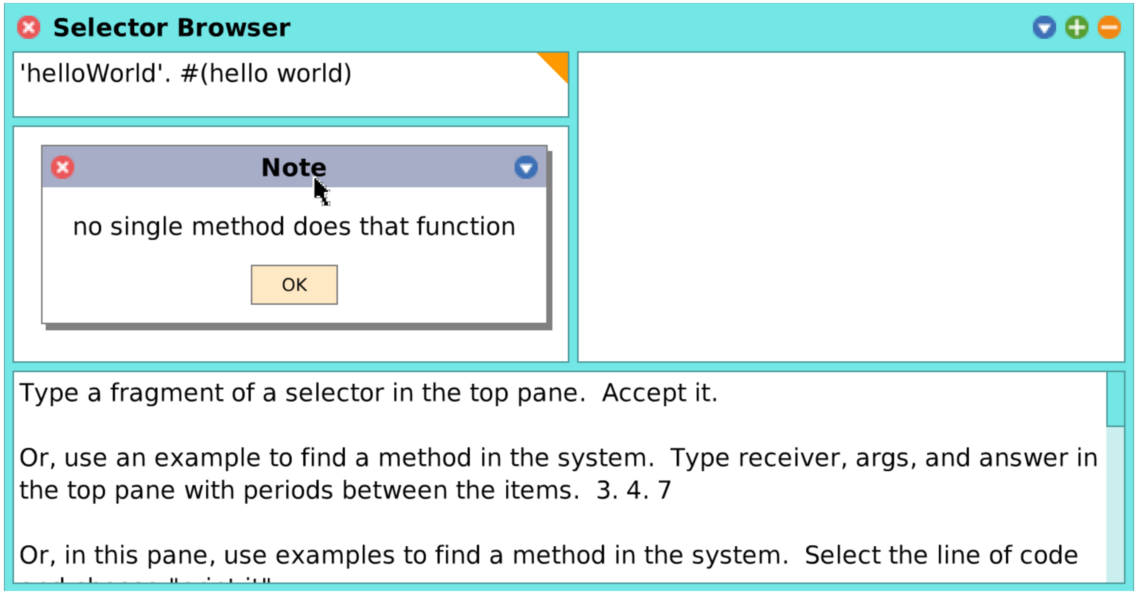


```
initialize
	"The methods we are allowed to use.  (MethodFinder new initialize) "

	Approved := Set new.
	AddAndRemove := Set new.
	Blocks := Set new.
	"These modify an argument and are not used by the MethodFinder: longPrintOn: printOn: storeOn: sentTo:
storeOn:base: printOn:base: absPrintExactlyOn:base: absPrintOn:base: absPrintOn:base:digitCount: writeOn:
writeScanOn: possibleVariablesFor:continuedFrom: printOn:format:"

"Object"
	#("in class, instance creation" categoryForUniclasses chooseUniqueClassName initialInstance
isSystemDefined newFrom: officialClass readCarefullyFrom:
"accessing" at: basicAt: basicSize bindWithTemp: in: size yourself
"testing" basicType ifNil: ifNil:ifNotNil: ifNotNil: ifNotNil:ifNil: isColor isFloat isFraction isInMemory isInteger
isMorph isNil isNumber isPoint isText isTransparent isWebBrowser knownName notNil pointsTo: wantsSteps
"comparing" = == closeTo: hash identityHash identityHashPrintString ~= ~~
"copying" copy shallowCopy
"dependents access" canDiscardEdits dependents hasUnacceptedEdits
"updating" changed changed: okToChange update: windowIsClosing
"printing" fullPrintString isLiteral longPrintString printString storeString stringForReadout stringRepresentation
"class membership" class isKindOf: isKindOf:orOf: isMemberOf: respondsTo: xxxClass
"error handling"
"user interface" addModelMenuItemsTo:forMorph:hand: defaultLabelForInspector fullScreenSize initialExtent
modelWakeUp mouseUpBalk: newTileMorphRepresentative windowActiveOnFirstClick windowReqNewLabel:
"system primitives" asOop instVarAt: instVarNamed:
"private"
"associating" ->
"converting" as: asOrderedCollection asString
"casing" caseOf: caseOf:otherwise:
"binding" bindingOf:
"macpal" contentsChanged currentEvent currentHand currentWorld flash instanceVariableValues
scriptPerformer
"flagging" flag:
```

mt 10/14/2020 13:46 · Marcel Taeumel · initialize · 1010 implementors · in no change set ·



'helloWorld'. #(hello world)

no single method does that function

Type a fragment of a selector in the top pane. Accept it.

Or, use an example to find a method in the system. Type receiver, args, and answer in the top pane with periods between the items. 3. 4. 7

Or, in this pane, use examples to find a method in the system. Select the line of code

Warum Whitelisting?

- Ausführen beliebiger Methoden könnte unerwartete/ unerwünschte Konsequenzen haben
  - ActiveWorld lock
  - 1 hour wait
  - self inform: ‘Please close this dialog’
  - Smalltalk quitPrimitive
  - FileDirectory default deleteRecursive
- allgemein: Seiteneffekte
  - globaler Zustand wird verändert (ActiveWorld, …)
  - Systemumgebung wird beeinflusst (Dateisystem, …)

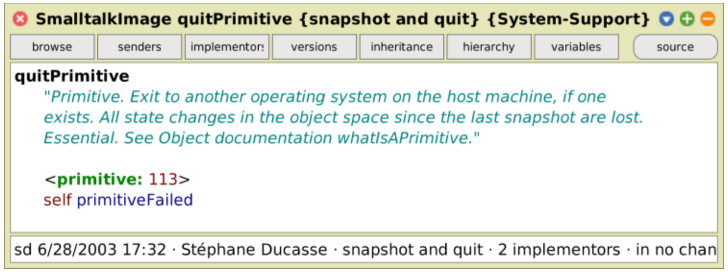
Idee: Seiteneffekte erkennen und verhindern

Props an Eliot Miranda

- Code Simulation

```
thisContext
    runSimulated: [2 + 3]
    contextAtEachStep: [:ctx |
        ctx willReallyStore
            ifTrue: [self error: 'forbidden’]]
```

- Arten von Seiteneffekten
  - Primitiven
    - primitiveSuspend, primitiveQuit, …
    - primitiveAtPut, …
  - Instruktionen
    - popIntoRcvr, …

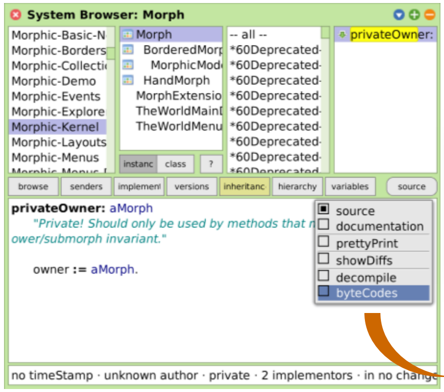


SmalltalkImage quitPrimitive {snapshot and quit} {System-Support}

```
quitPrimitive
	"Primitive. Exit to another operating system on the host machine, if one
	exists. All state changes in the object space since the last snapshot are lost.
	Essential. See Object documentation whatIsAPrimitive."

	<primitive: 113>
	self primitiveFailed
```

sd 6/28/2003 17:32 · Stéphane Ducasse · snapshot and quit · 2 implementors · in no chan



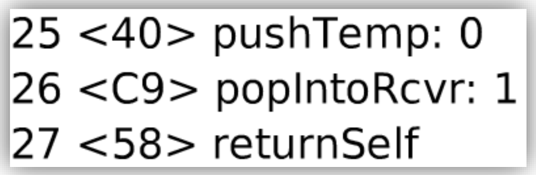
System Browser: Morph

```
privateOwner: aMorph
	"Private! Should only be used by methods that n
owner/submorph invariant."

	owner := aMorph.
```

- ▣ source
- □ documentation
- □ prettyPrint
- □ showDiffs
- □ decompile
- □ byteCodes

no timeStamp · unknown author · private · 2 implementors · in no change



```
25 <40> pushTemp: 0
26 <C9> popIntoRcvr: 1
27 <58> returnSelf
```

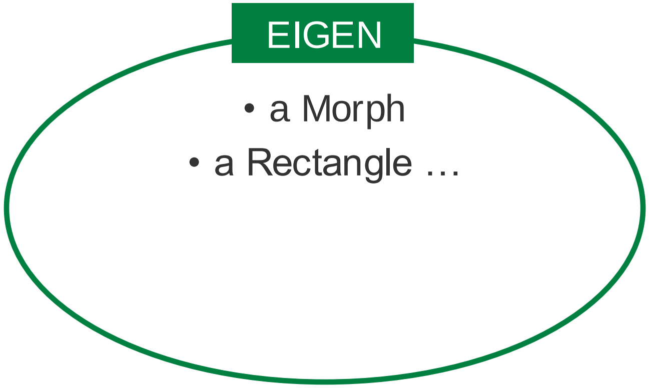
Idee: Seiteneffekte erkennen und verhindern

- Einschränkungen:
  - Lokale Seiteneffekte (z. B. Collection>>#collect:)
  - Temporäre Seiteneffekte (z. B. ClassBuilder beSilentDuring:)
  - Irrelevante Seiteneffekte (z. B. Caches)
  - Interessante Seiteneffekte (z. B. OrderedCollection>>#add:)

Idee 2: Seiteneffekte erkennen und isolieren

- Object Ownership tracken
  - Eigene Objekte: Voller Zugriff
  - Fremde Objekte: transparente Umleitung auf eigene Kopie

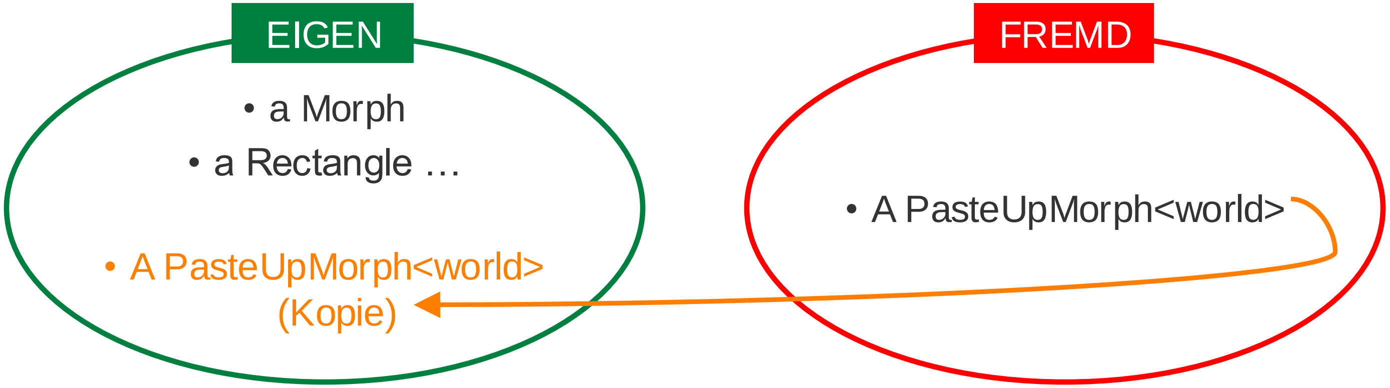
  - Morph new



Idee 2: Seiteneffekte erkennen und isolieren

- Object Ownership tracken
  - Eigene Objekte: Voller Zugriff
  - Fremde Objekte: transparente Umleitung auf eigene Kopie

  - Morph new openInWorld



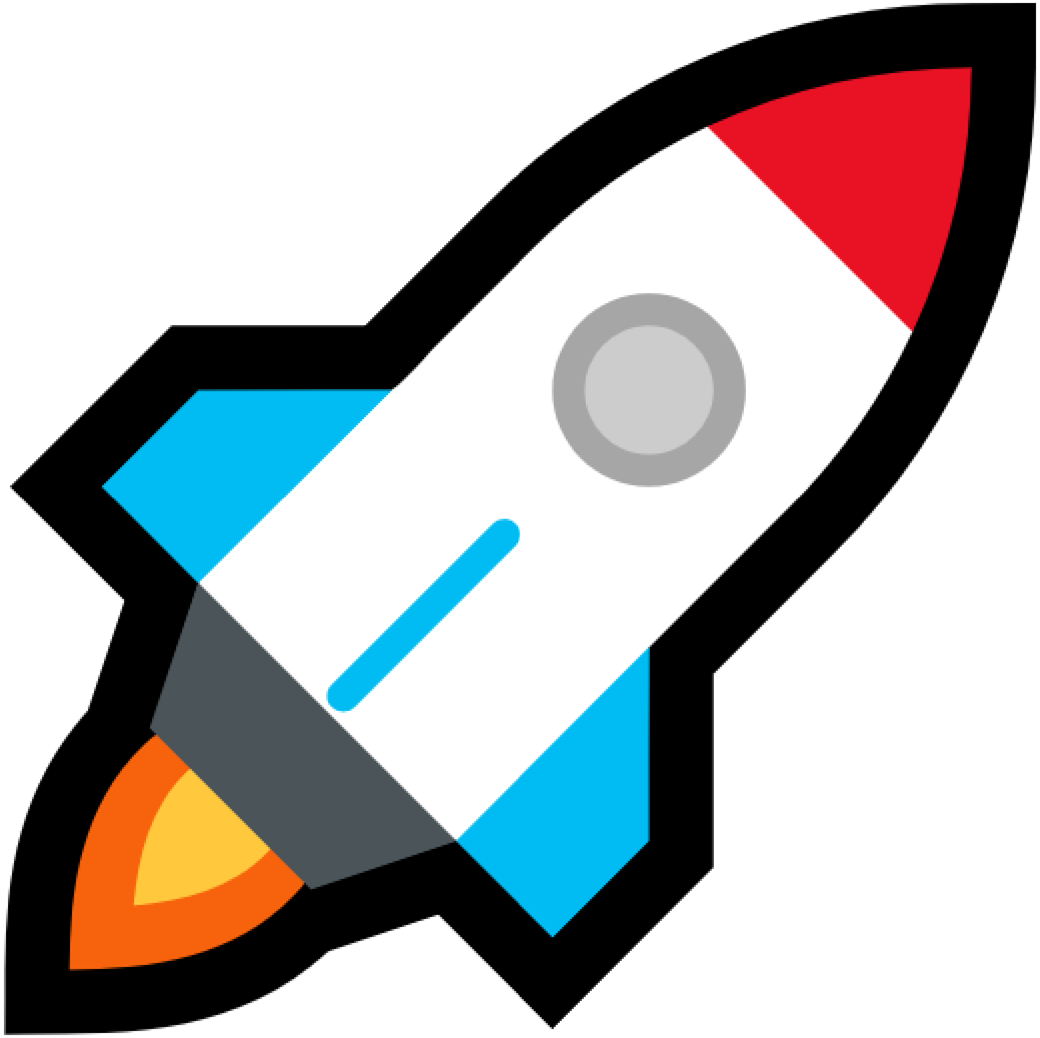
- Primitiven mit Einfluss auf Systemumgebung deaktivieren → Fallbackcode auslösen

Sandbox

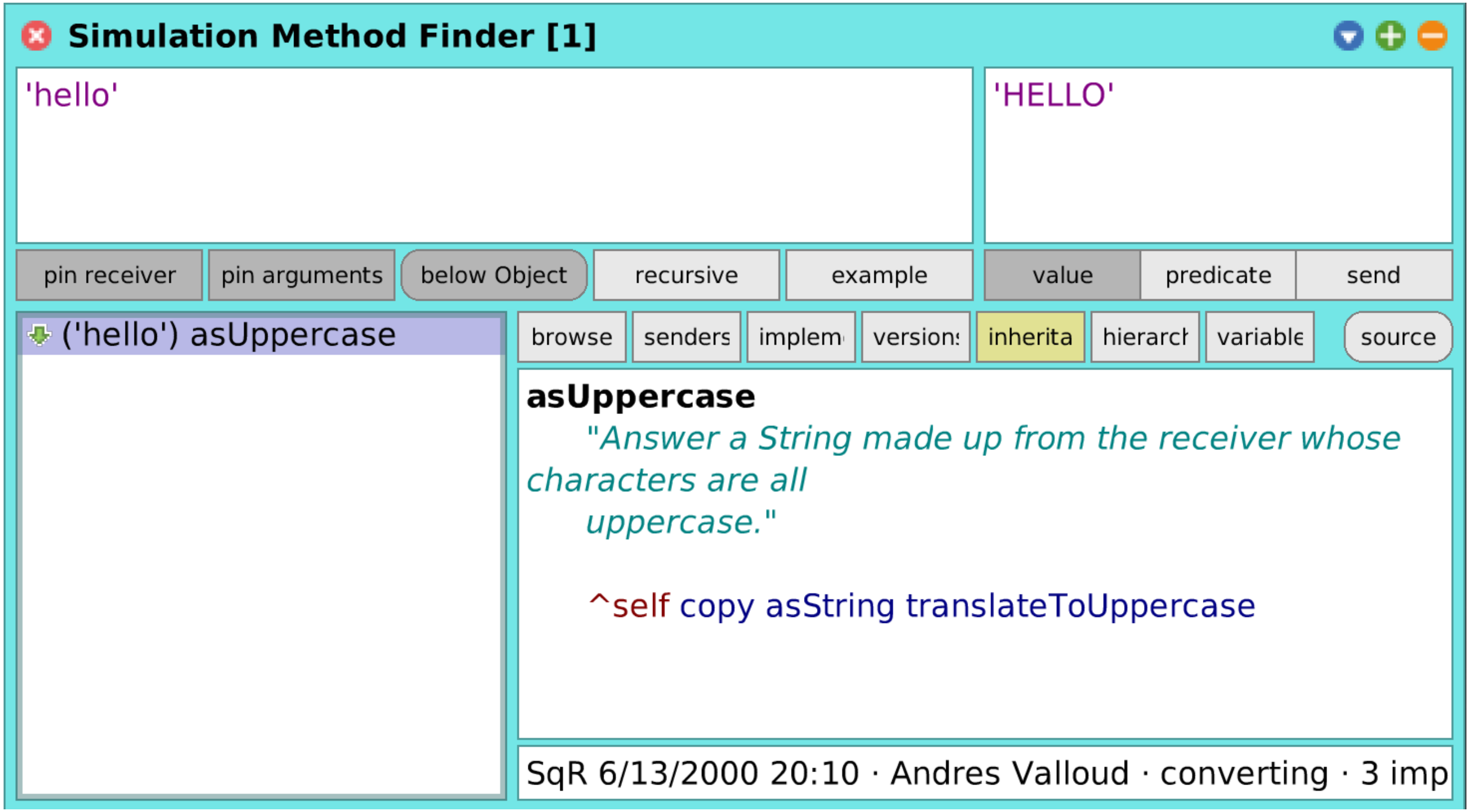
```
array := {1. 2. 3}.

Sandbox evaluate:
        [array at: 1 put: 10.
        array first]. "10"

array first. "1"
```



Method Finder 2



'hello'

'HELLO'

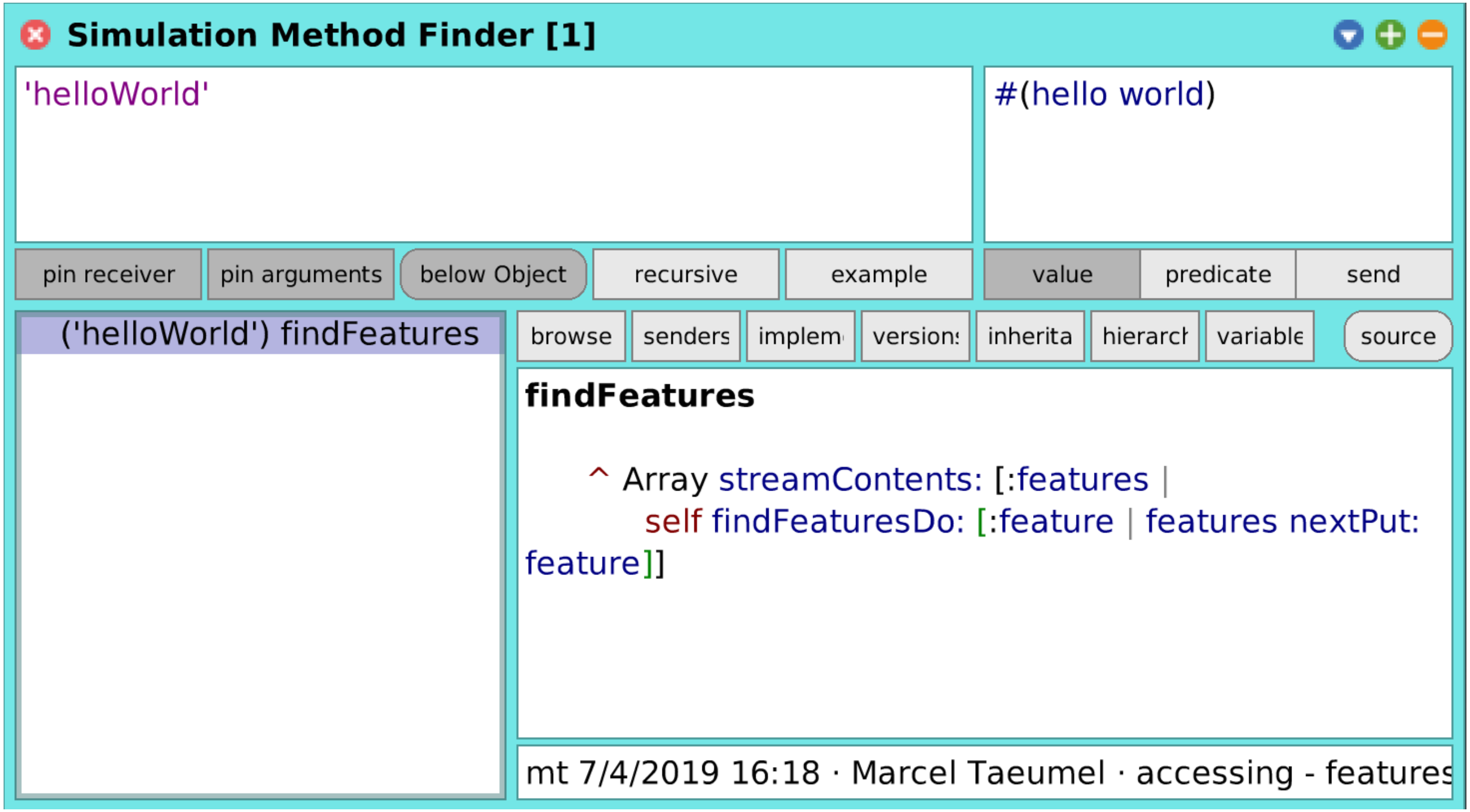
('hello') asUppercase

```
asUppercase
	"Answer a String made up from the receiver whose characters are all
	uppercase."

	^self copy asString translateToUppercase
```

SqR 6/13/2000 20:10 · Andres Valloud · converting · 3 imp

Method Finder 2

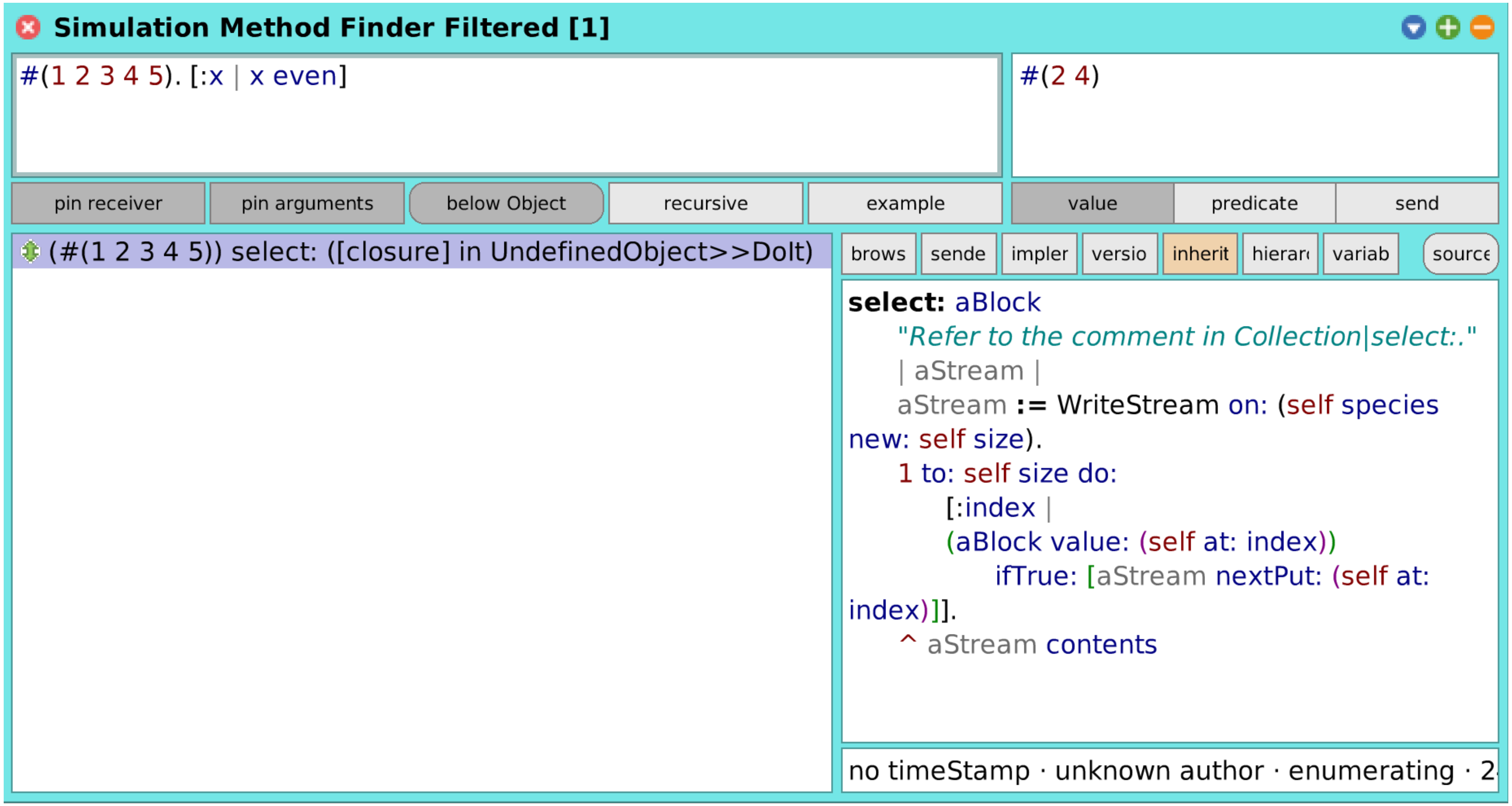


```
findFeatures

	^ Array streamContents: [:features |
		self findFeaturesDo: [:feature | features nextPut: feature]]
```

mt 7/4/2019 16:18 · Marcel Taeumel · accessing - features

Method Finder 2: Blocksuche



```
#(1 2 3 4 5). [:x | x even]
```

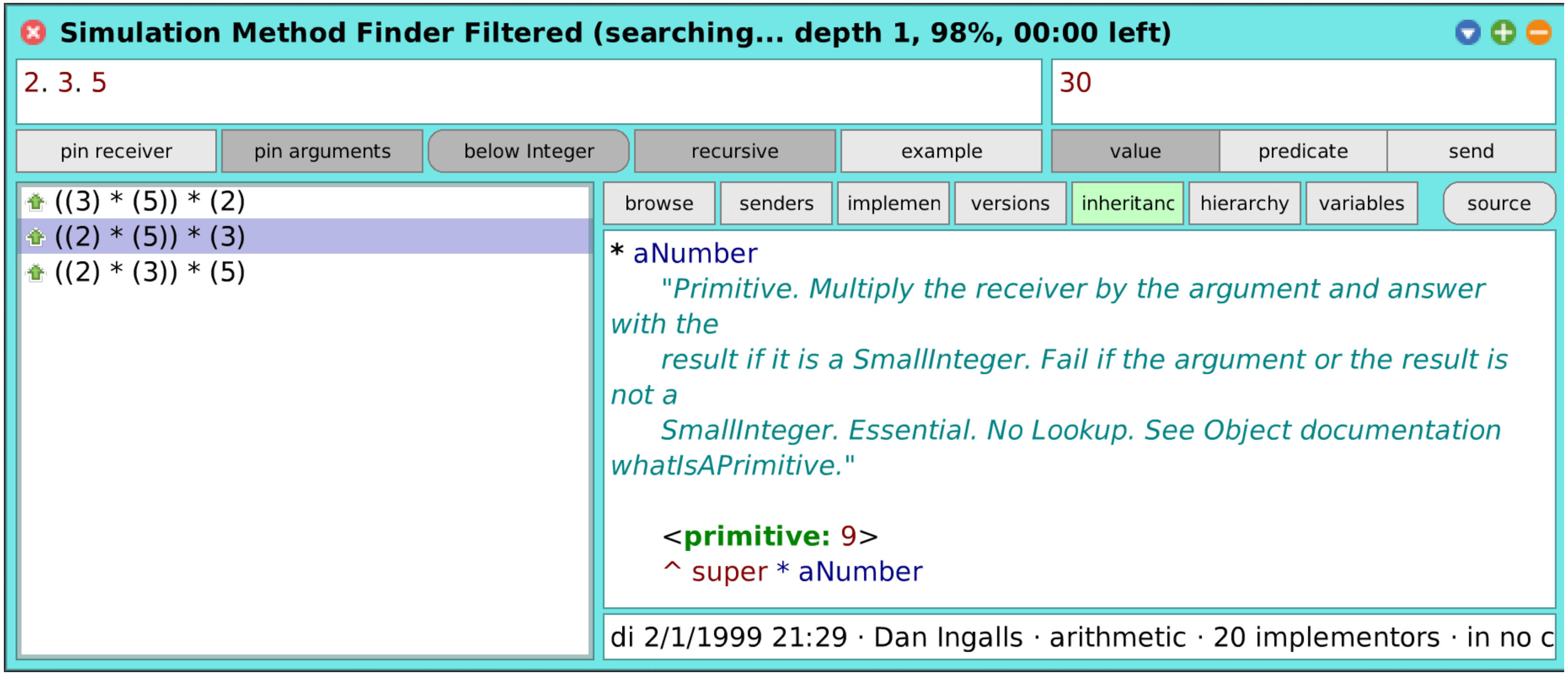
```
#(2 4)
```

```
(#(1 2 3 4 5)) select: ([closure] in UndefinedObject>>DoIt)
```

```
select: aBlock
	"Refer to the comment in Collection|select:."
	| aStream |
	aStream := WriteStream on: (self species
new: self size).
	1 to: self size do:
		[:index |
		(aBlock value: (self at: index))
			ifTrue: [aStream nextPut: (self at:
index)]].
	^ aStream contents
```

no timeStamp · unknown author · enumerating · 2

Method Finder 2: Rekursive Suche



Simulation Method Finder Filtered (searching... depth 1, 98%, 00:00 left)

2. 3. 5

30

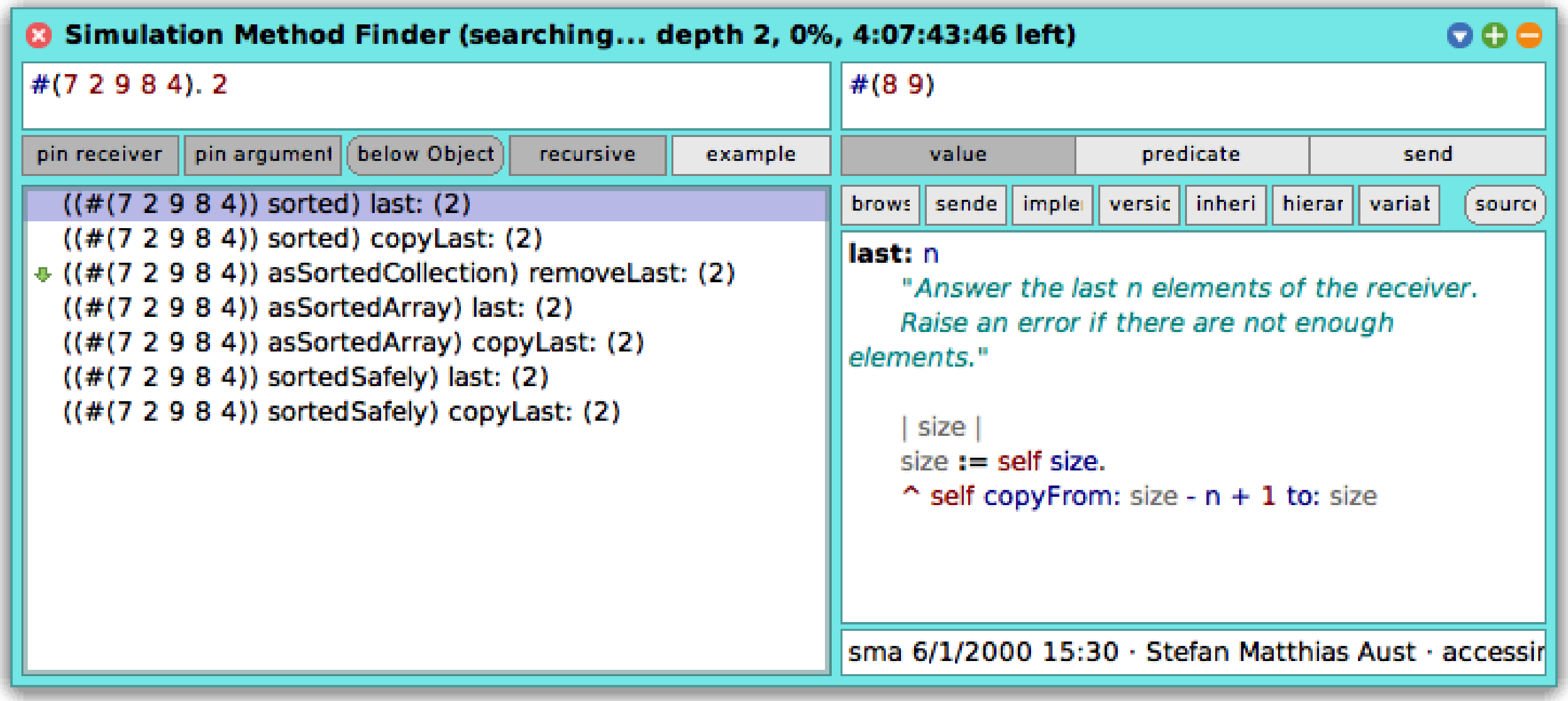
```
((3) * (5)) * (2)
((2) * (5)) * (3)
((2) * (3)) * (5)
```

```
* aNumber
	"Primitive. Multiply the receiver by the argument and answer with the
	result if it is a SmallInteger. Fail if the argument or the result is not a
	SmallInteger. Essential. No Lookup. See Object documentation whatIsAPrimitive."

	<primitive: 9>
	^ super * aNumber
```

di 2/1/1999 21:29 · Dan Ingalls · arithmetic · 20 implementors · in no c

Method Finder 2: Rekursive Suche



```
#(7 2 9 8 4). 2
```

```
#(8 9)
```

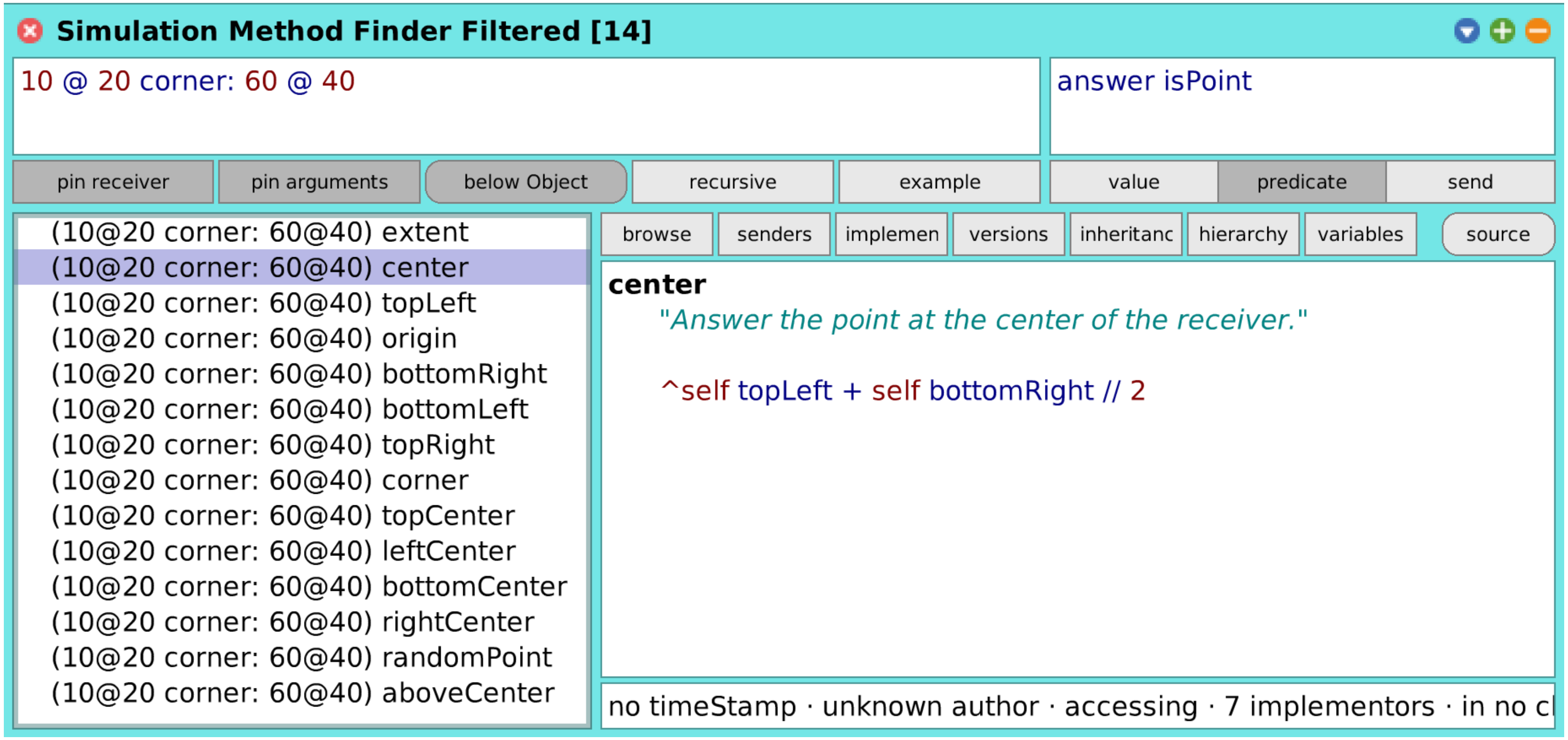
```
((#(7 2 9 8 4)) sorted) last: (2)
((#(7 2 9 8 4)) sorted) copyLast: (2)
((#(7 2 9 8 4)) asSortedCollection) removeLast: (2)
((#(7 2 9 8 4)) asSortedArray) last: (2)
((#(7 2 9 8 4)) asSortedArray) copyLast: (2)
((#(7 2 9 8 4)) sortedSafely) last: (2)
((#(7 2 9 8 4)) sortedSafely) copyLast: (2)
```

```
last: n
	"Answer the last n elements of the receiver.
	Raise an error if there are not enough
elements."

	| size |
	size := self size.
	^ self copyFrom: size - n + 1 to: size
```

sma 6/1/2000 15:30 · Stefan Matthias Aust · accessir

Method Finder 2: Prädikatsuche



```
10 @ 20 corner: 60 @ 40
```

```
answer isPoint
```

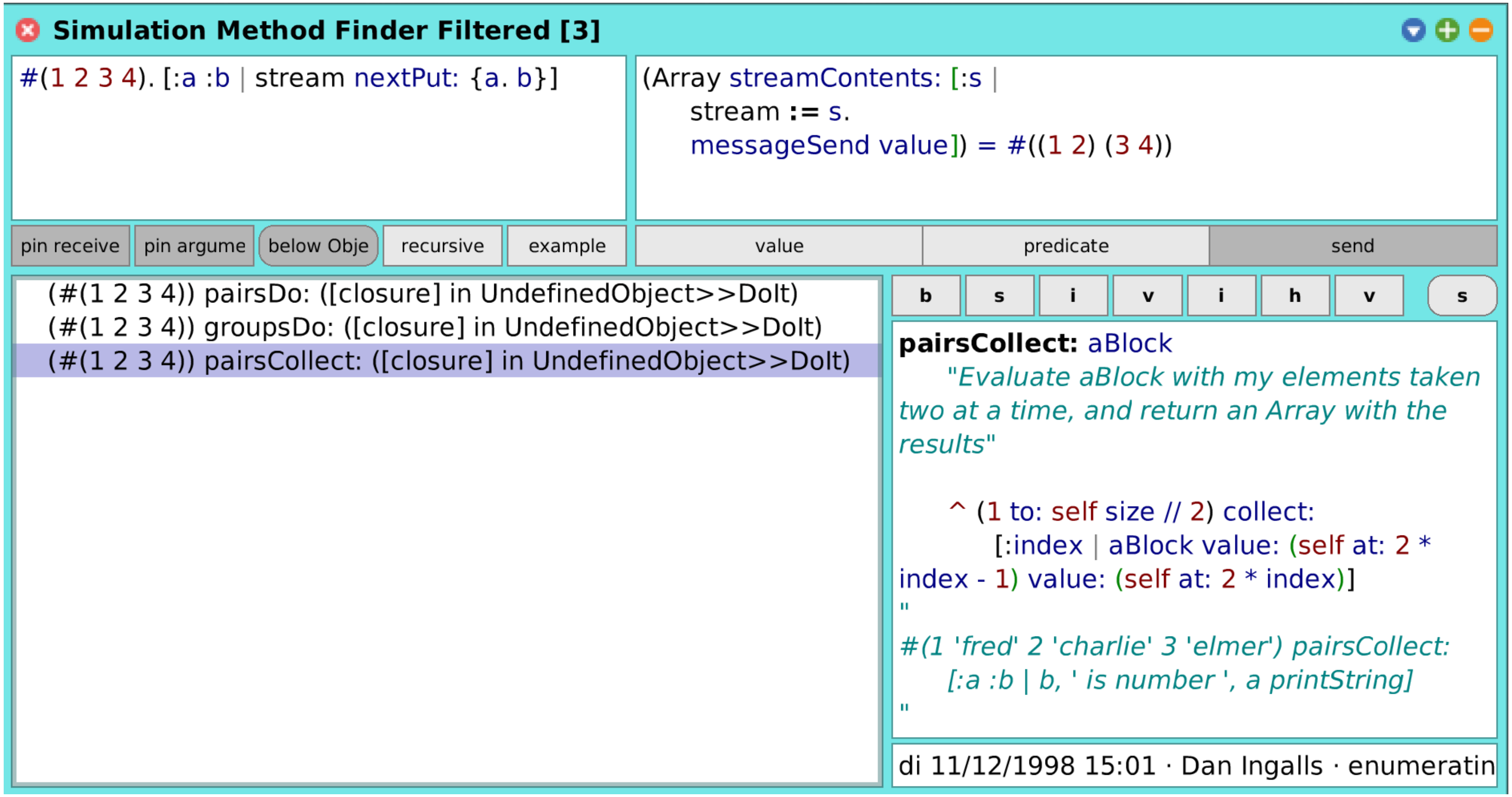
```
(10@20 corner: 60@40) extent
(10@20 corner: 60@40) center
(10@20 corner: 60@40) topLeft
(10@20 corner: 60@40) origin
(10@20 corner: 60@40) bottomRight
(10@20 corner: 60@40) bottomLeft
(10@20 corner: 60@40) topRight
(10@20 corner: 60@40) corner
(10@20 corner: 60@40) topCenter
(10@20 corner: 60@40) leftCenter
(10@20 corner: 60@40) bottomCenter
(10@20 corner: 60@40) rightCenter
(10@20 corner: 60@40) randomPoint
(10@20 corner: 60@40) aboveCenter
```

```
center
	"Answer the point at the center of the receiver."

	^self topLeft + self bottomRight // 2
```

no timeStamp · unknown author · accessing · 7 implementors · in no cl

Method Finder 2: Kontextsuche



```
#(1 2 3 4). [:a :b | stream nextPut: {a. b}]
```

```
(Array streamContents: [:s |
	stream := s.
	messageSend value]) = #((1 2) (3 4))
```

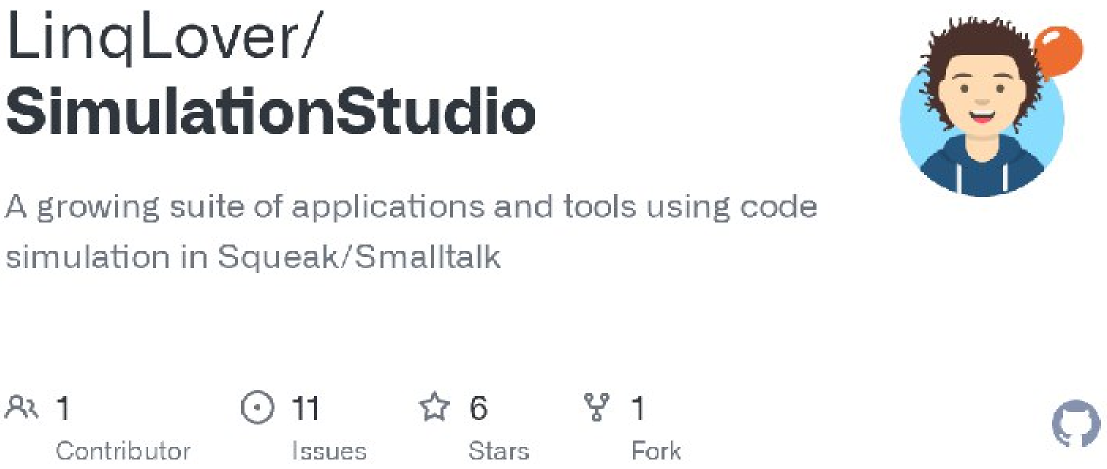
```
(#(1 2 3 4)) pairsDo: ([closure] in UndefinedObject>>DoIt)
(#(1 2 3 4)) groupsDo: ([closure] in UndefinedObject>>DoIt)
(#(1 2 3 4)) pairsCollect: ([closure] in UndefinedObject>>DoIt)
```

```
pairsCollect: aBlock
	"Evaluate aBlock with my elements taken two at a time, and return an Array with the results"

	^ (1 to: self size // 2) collect:
		[:index | aBlock value: (self at: 2 * index - 1) value: (self at: 2 * index)]
"
#(1 'fred' 2 'charlie' 3 'elmer') pairsCollect:
	[:a :b | b, ' is number ', a printString]
"
```

di 11/12/1998 15:01 · Dan Ingalls · enumeratin

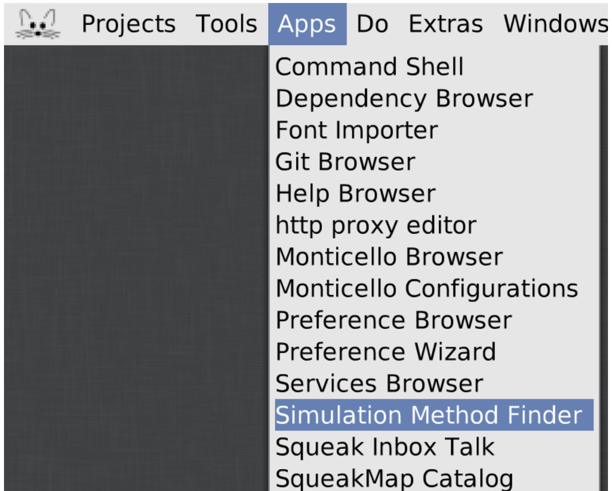
Selbst ausprobieren!



LinqLover/
SimulationStudio

A growing suite of applications and tools using code simulation in Squeak/Smalltalk

```
Metacello new
    baseline: 'SimulationStudio';
    githubUser: 'LinqLover’
        project: 'SimulationStudio’
        path: 'packages';
    load.
```



Zum Nachlesen

- https://github.com/LinqLover/SimulationStudio
- [squeak-dev] [ANN] News from SimulationStudio: Method Finder 2
- [squeak-dev] [ANN] SimulationStudio and sandboxed execution for Squeak
- [squeak-dev] Re: MethodFinder.Blocks