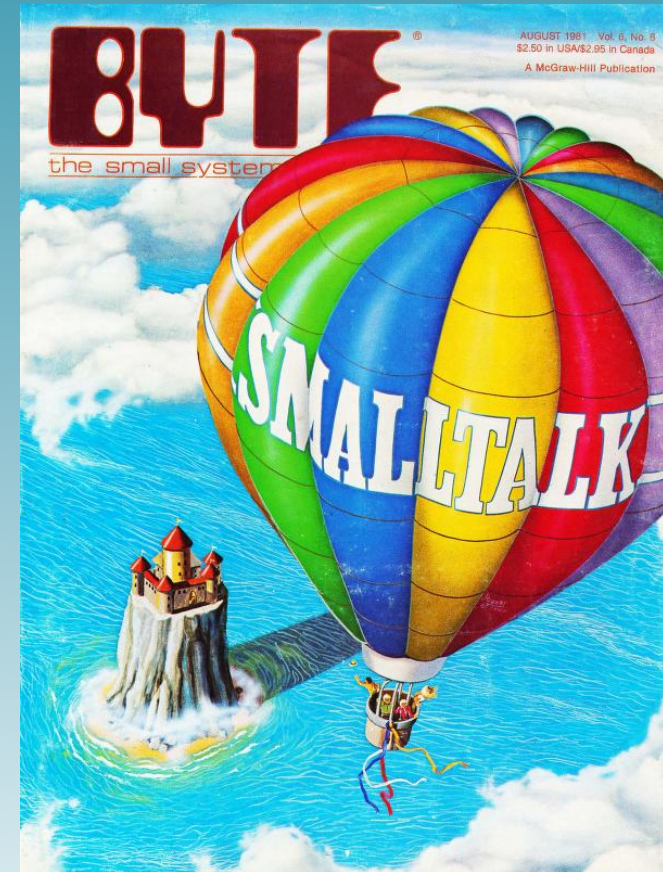


# Cuis Smalltalk Method Finder

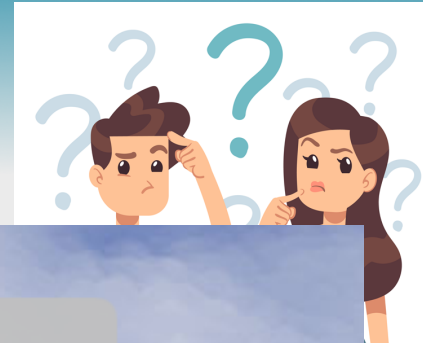
R. Mark Volkmann  
Object Computing, Inc.  
<https://objectcomputing.com>  
@mark\_volkman



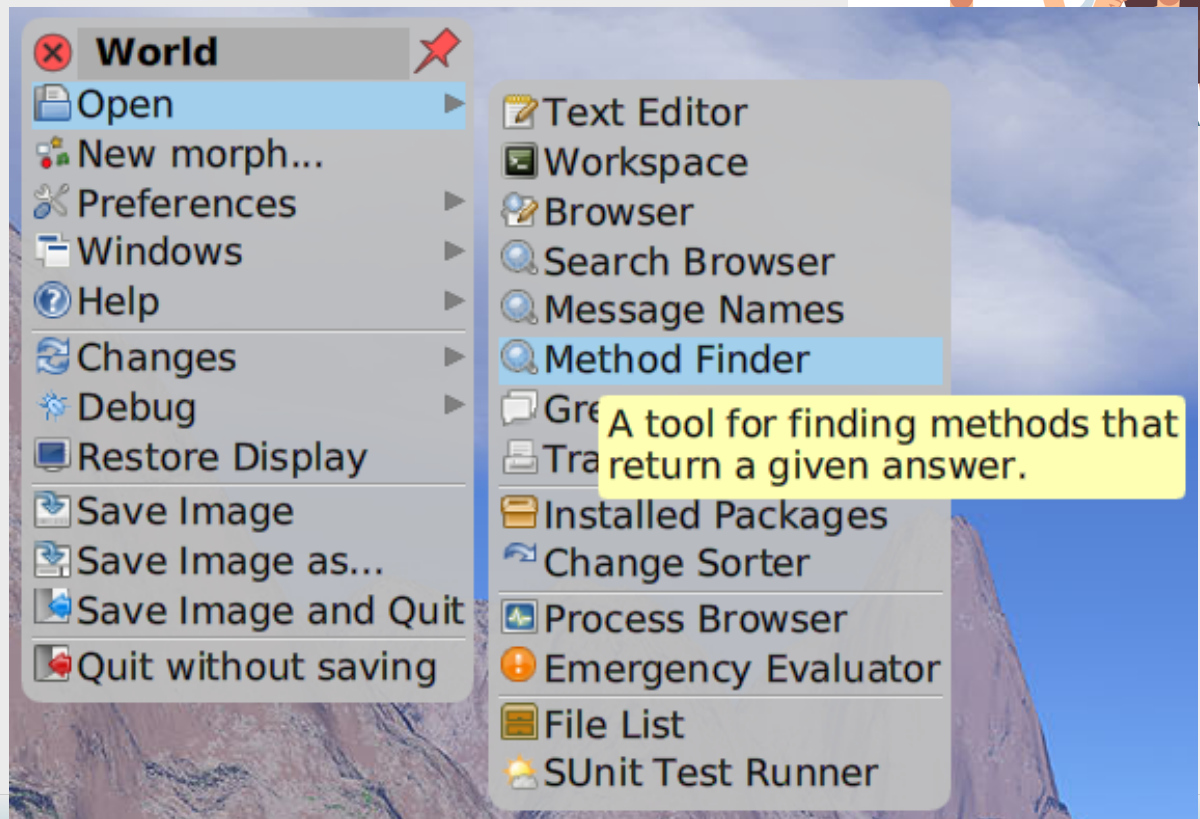
**OBJECT COMPUTING**  
YOUR OUTCOMES ENGINEERED

Slides at <https://github.com/mvolkmann/talks/>

# What is it?



- Tool for finding methods
  - on given receiver
  - that take given arguments
  - and produce given result
- Open World menu
- Hover over “Open”
- Click “Method Finder”



# The GUI

Receiver: 2  
Arguments (dot-separated): 2  
Expected: 4  
Find Methods 7 matching methods were found.  
Integer digitAdd:  
Number ^  
Number raisedTo:  
Number raisedToInteger:  
Number raisedToNegativeInteger:  
SmallInteger \*  
SmallInteger +

Utilizes classes from **SearchBrowser** by Mariano Montone in <https://github.com/Cuis-Smalltalk/Cuis-Smalltalk-Tools>  
**MethodFinderWindow** is a subclass of **SearchBrowserWindow** which is a subclass of **MethodSetWindow** which is a subclass of **CodeWindow** which is a subclass of **SystemWindow**

jmv 8/1/2019 19:23:05 ° 20 message sends ° mathematical functions ° 3 implementors ° 24 direct senders ° in no package ° in no change set  
browse senders implementors versions inheritance hierarchy inst vars class vars  
**raisedTo: exponent**  
*"Answer the receiver raised to aNumber."*  
exponent isInteger ifTrue: [  
*"Do the special case of integer power"*  
↑ self raisedToInteger: exponent].  
exponent isFraction ifTrue: [  
*"Special case for fraction power by Nicolas Cellier:*  
*If aNumber is a fraction, but result must be a Float, learn it as quickly as possible, and give quick Float*  
answer  
*Allows evaluating:*  
*(2009/2000) raisedTo: (3958333/100000)*  
"  
↑ self raisedToFraction: exponent].  
self < 0 ifTrue: [  
↑ **NegativePowerError** new signalReceiver: self selector: **#raisedTo:** argument: exponent].  
0.0 = exponent ifTrue: [↑ self class one] "Special case of exponent=0"



# From Workspace

- Open Workspace
- Evaluate expression like  
`MethodFinder methodFor: #(2 2 4)`
- Prints to Transcript

```
7 matching methods were found.  
- SmallInteger *  
- SmallInteger +  
- Number ^  
- Integer digitAdd:  
- Number raisedTo:  
- Number raisedToInteger:  
- Number raisedToNegativeInteger:
```



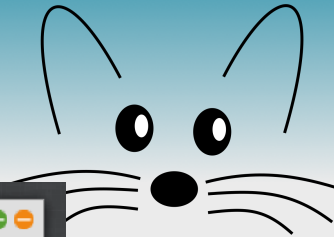
# Installing



- Clone repository
  - `git clone https://github.com/mvolkmann/Cuis-Smalltalk-MethodFinder`
- Open a Cuis image
- Open a Workspace
- Evaluate `Feature require #MethodFinder`



# In Squeak



The screenshot shows the Squeak IDE interface. On the left, the 'World' menu is open, displaying various options. The 'open...' option is highlighted with a red box. To its right, the 'open...' submenu is visible, listing various browsers and tools. The 'method finder' option in the submenu is also highlighted with a red box. On the right side of the interface, the 'Selector Browser' window is open. It contains a list of selectors, with '2 raisedTo: 2 --> 4' selected. Below the list, there is a text area with instructions on how to use the Selector Browser to find methods.

**World**

- Browser
- Workspace
- Transcript
- Test Runner
- previous project
- jump to project...
- save project on file...
- load project from file...
- undo "move Selector Bro
- restore display (r)
- open...**
- windows...
- changes...
- help...
- appearance...
- do...
- objects (o)
- new morph...
- authoring tools...
- playfield options...
- flaps...
- projects...
- telemorphic...
- make screenshot
- debug...
- save
- save as...
- save as new version
- save and quit
- quit

**open...**

- class browser
- workspace
- file list
- package pane browser
- process browser
- method finder**
- message names
- simple change sorter
- dual change sorter
- file...
- transcript (t)
- Dependency Browser
- Font Importer
- Help Browser
- http proxy editor
- Monticello Browser
- Monticello Configurations
- Preference Browser
- Preference Wizard
- Services Browser
- SqueakMap Catalog
- SqueakMap Categories
- Test Runner
- mvc project
- morphic project

**Selector Browser**

2. 2. 4

Collection raisedTo:  
Complex raisedTo:  
\*Number raisedTo:

2 \* 2 --> 4  
2 + 2 --> 4  
**2 raisedTo: 2 --> 4**  
2 raisedToInteger: 2 --> 4

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 and choose "print it".

MethodFinder methodFor: #( (4 3) 7 (0 5) 5 (5 5) 10).  
This will discover (data1 + data2).

You supply inputs and answers and the system will find the method. Each inner array is a list of inputs. It contains the receiver and zero or more arguments. For Booleans and any computed arguments, use brace notation.

MethodFinder methodFor: { {1. 3}. true. {20. 10}. false}.  
This will discover the expressions (data1 < data2), (data2 > data1), and many others.

# How Does It Work? ...



- Takes string expressions for receiver, arguments, and expected
- Compiles string expressions to objects
  - `Compiler evaluate: expressionString`
    - can detect when an expression is invalid, but can't describe why
- Gets all selectors for receiver found in its class and all superclasses
  - `receiverObject class allSelectors`

# ... How Does It Work? ...

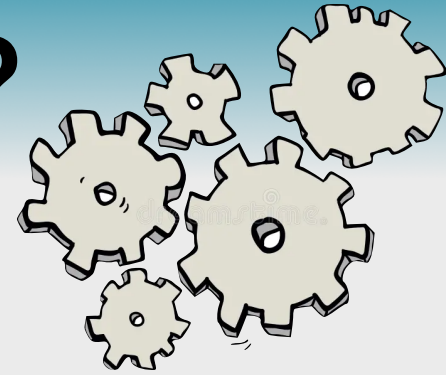


- Rejects selectors that
  - do not take specified number of arguments
  - are in **SkipSet** (a **Set** initialized in **MethodFinder** class method **initialize**)
  - are understood by the **Behavior**, **ChangeSet**, or **ClassDescription** classes
  - include the substring '**halt**' in any case
  - begin with '**fileOut**' or '**handleFailing**'

**PROBLEM:** We can't anticipate dangerous methods that might be added in the future.



# ... How Does It Work?



- Iterates over remaining selectors to find **matches**
  - makes **veryDeepCopy** of receiver
  - makes **veryDeepCopy** of arguments array

```
actualObject :=  
    [receiverCopy perform: selector withArguments: argumentsCopy]  
    on: Error  
    do: [ :ex | nil ].  
  
actualObject isKindOf: Float :: and: [ expectedObject isKindOf: Float ] ::  
    ifTrue: [ actualObject - expectedObject :: abs <= 0.001 ]  
    ifFalse: [ actualObject = expectedObject ].
```

- Returns **matches** asSortedCollection

# Potentially Dangerous

- Blindly executing methods is dangerous
  - could modify existing classes and methods in bad ways
  - could prompt for input
  - could produce unwanted output
  - could be very slow
  - could crash image

- Examples

- **removeFromSystem**

"Forget the receiver from the Smalltalk global dictionary.  
Any existing instances will refer to an obsolete version of the receiver."

- **shutDown** `self terminateScreenUpdater`

**Assumptions  
are made and  
most  
assumptions  
are wrong.**

*- Albert Einstein*

# How Does Squeak Differ?



- **MethodFinder** instance method **initialize** creates a **Set** of “approved” selectors
  - also see instance method **noteDangerous**
  - class variables **Approved**, **AddAndRemove**, and **Dangerous** are used in instance method **organizationFiltered**
- The Good
  - no possibility of sending dangerous messages
- The Bad
  - can only send messages for registered selectors, so new ones won't be found

# Wrap Up

It's an interesting and potentially dangerous tool. How can it be improved?

## Question:

Recently I noticed that Cuis locks up when an image is started in macOS by double clicking and the window is dragged to another monitor. Can someone verify if this happens for them?

Slides at <https://github.com/mvolkmann/talks/>

Package at <https://github.com/mvolkmann/Cuis-Smalltalk-MethodFinder>