

# The Moldable Debugger

## A Framework for Developing Domain-Specific Debuggers

Andrei Chiș  
University of Bern

@chis\_andrei  
andreichis.com

Tudor Gîrba  
CompuGroup Medical  
Schweiz AG

@girba  
tudorgirba.com

Oscar Nierstrasz  
University of Bern

@onierstrasz  
scg.unibe.ch/staff/oscar

# Roadmap

1 Debugging needs

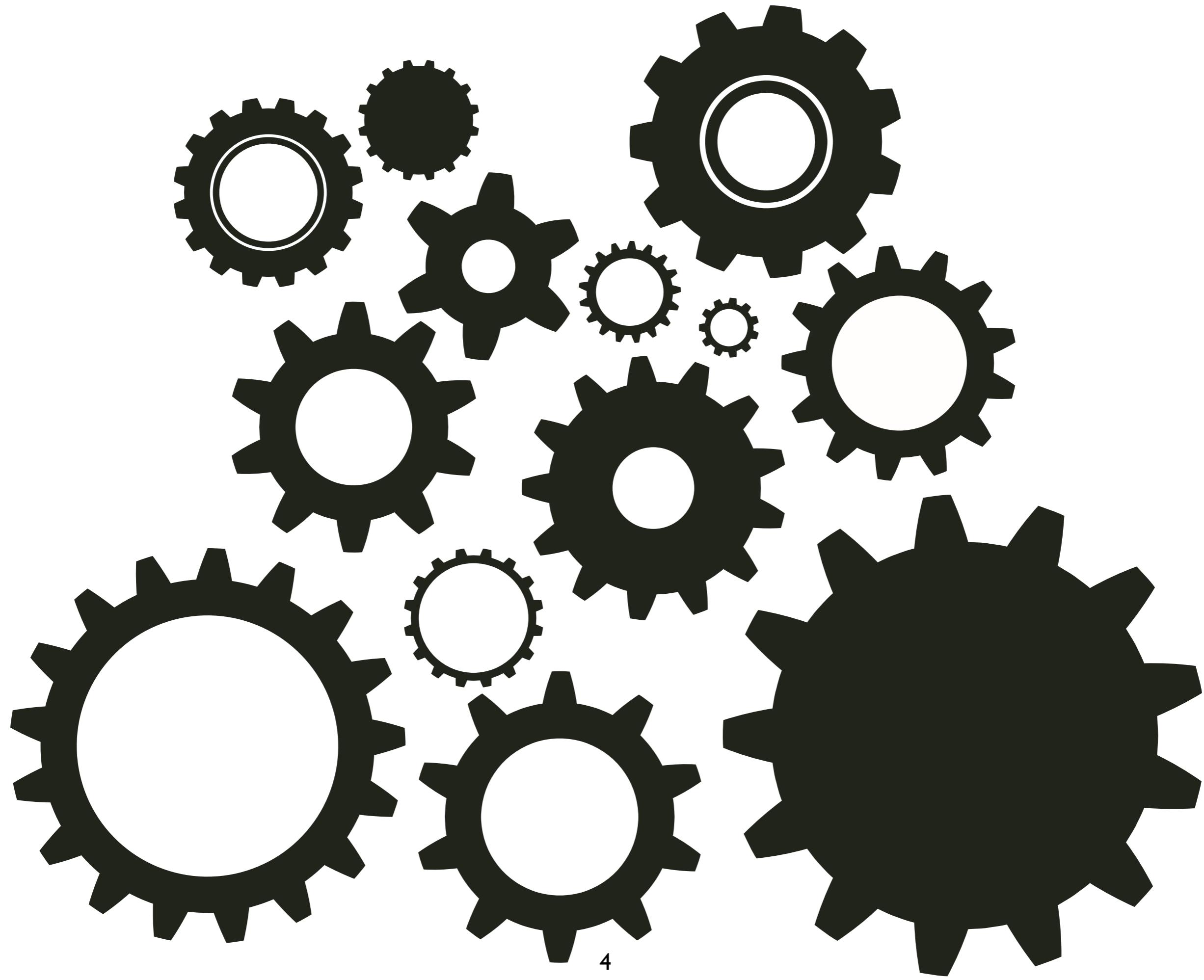
2 Motivating examples

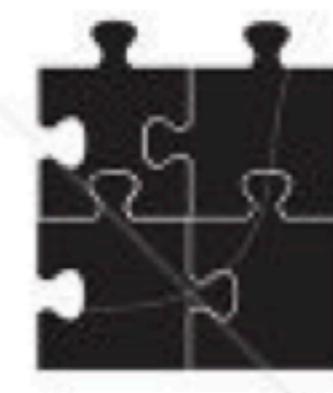
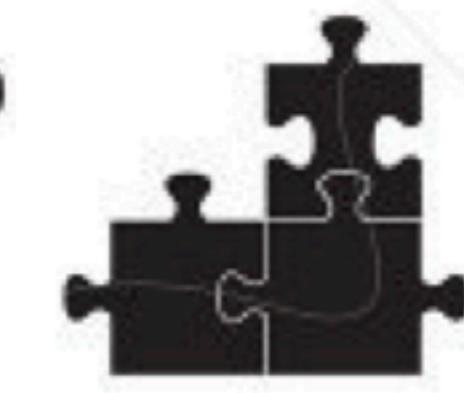
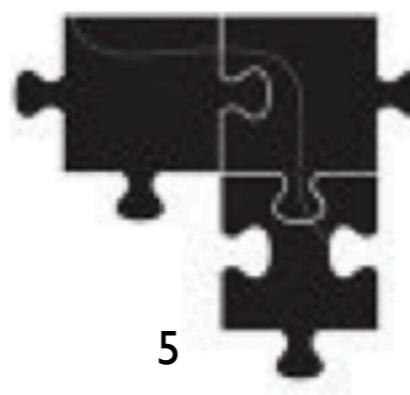
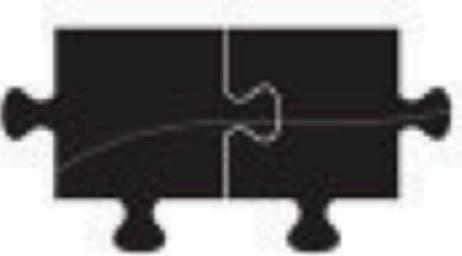
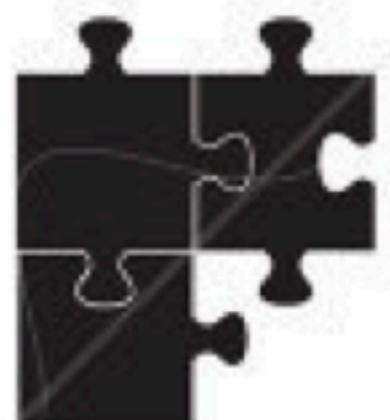
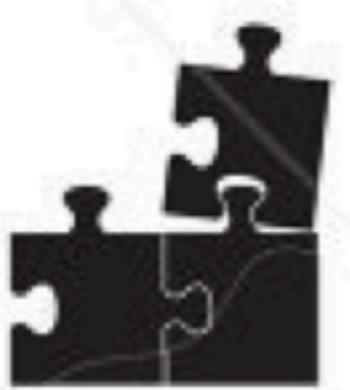
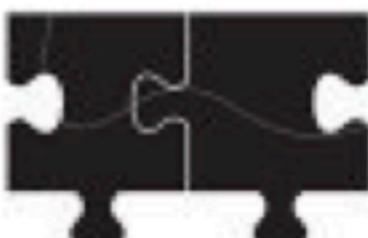
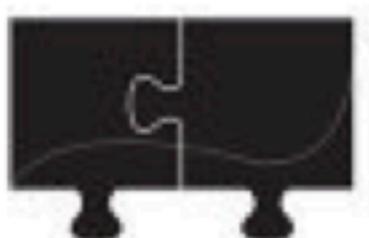
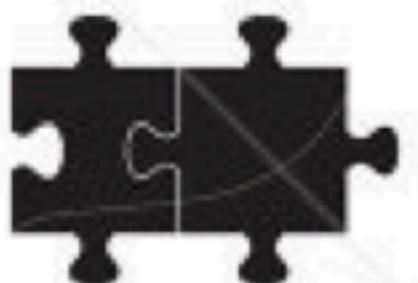
3 The Moldable Debugger

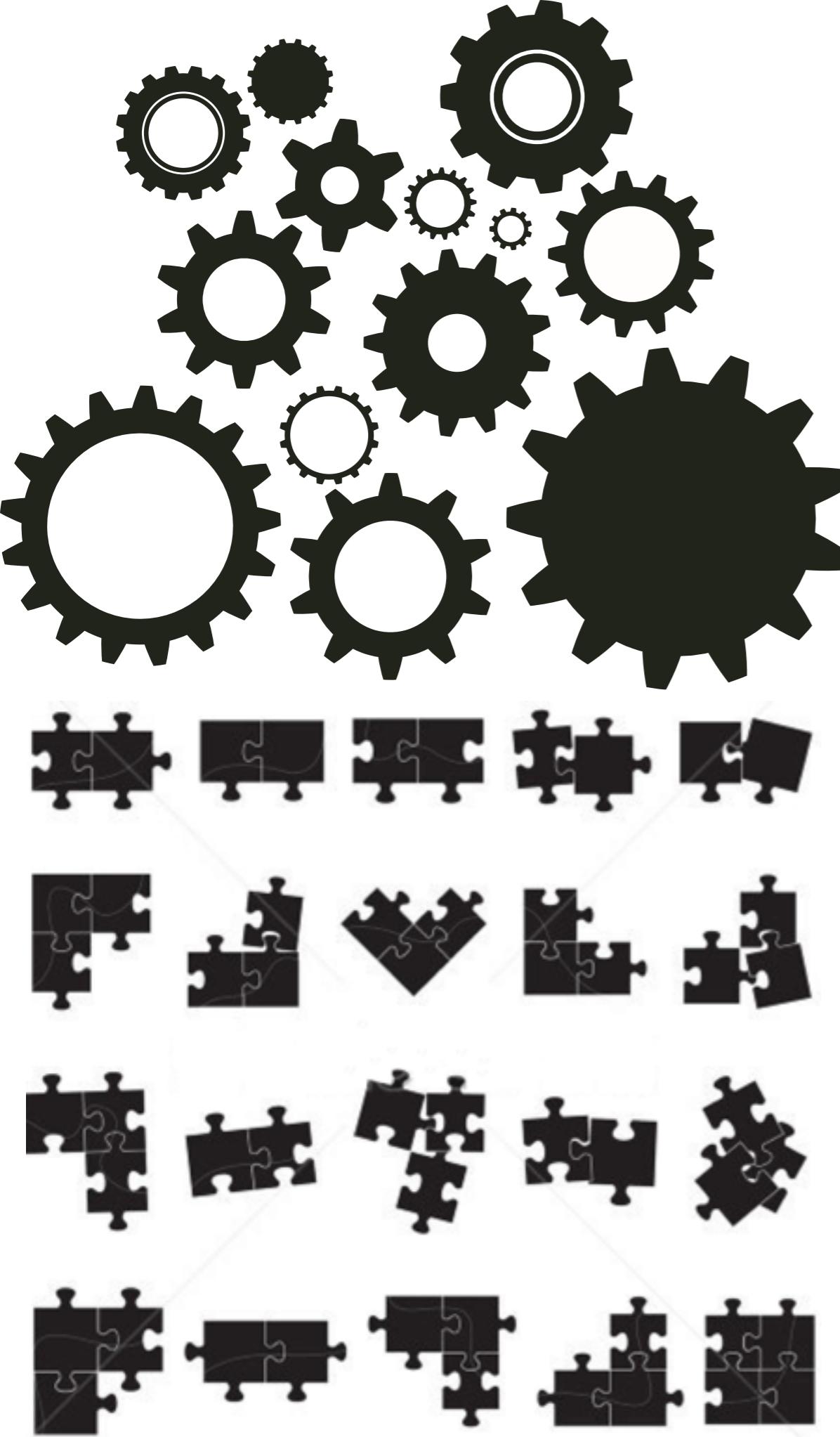
4 Domain-specific  
extensions

5 Activation predicates

Object-oriented applications use objects to express a model of the business domain







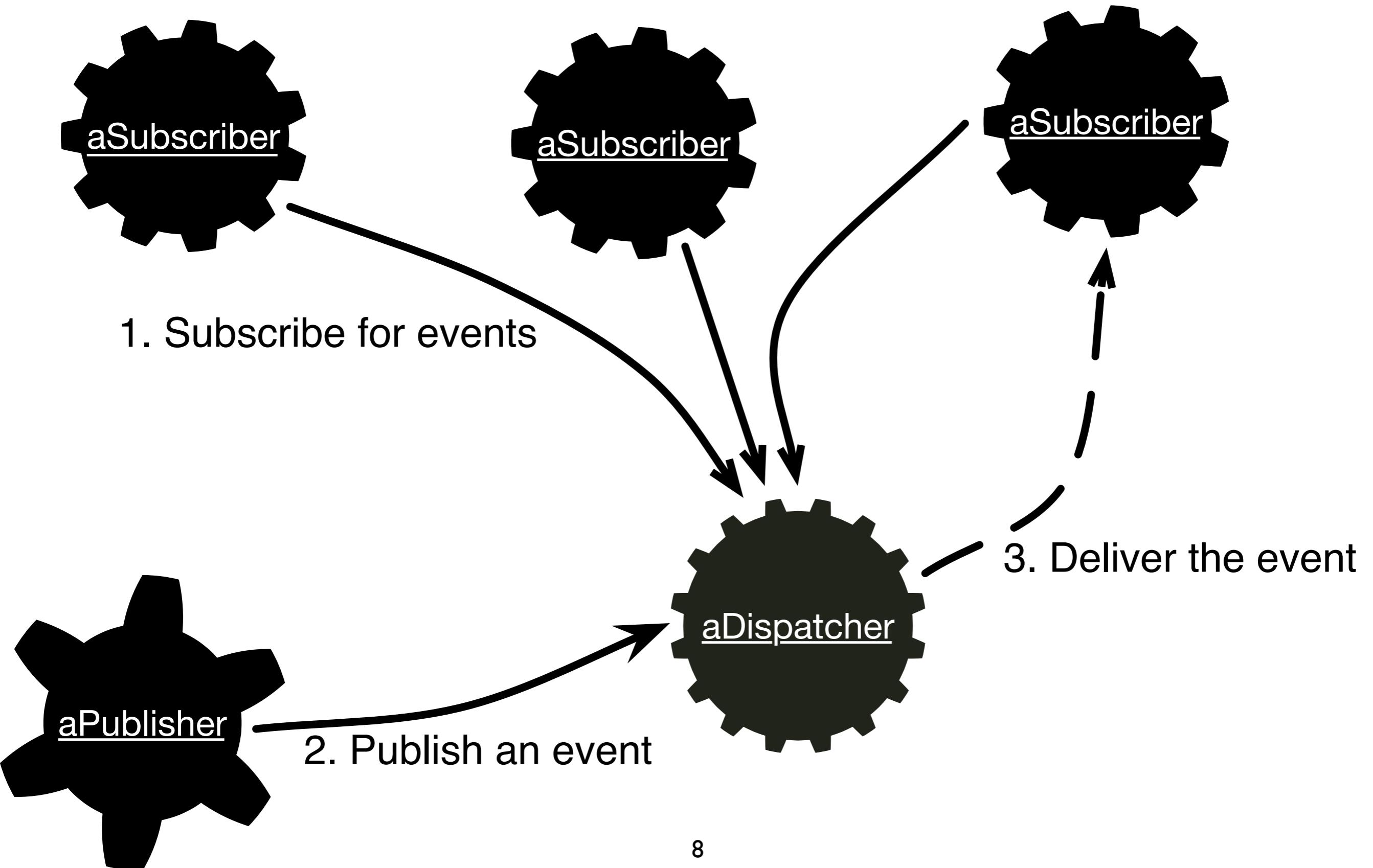
Stack

```
RubScrolledTextMorph>>whenTextAcceptRequest:  
MessageSend>>value:  
MessageSend>>cull:  
MessageSend>>cull:cull:  
AnnouncementSubscription>>deliver: in Block: [ action cull: anAnnouncement cull  
BlockClosure>>on:do:  
BlockClosure>>on:fork:  
AnnouncementSubscription>>deliver:  
SubscriptionRegistry>>deliver:to:startingAt: in Block: [ subscription deliver: anAnnounce  
BlockClosure>>ifCurtailed:  
SubscriptionRegistry>>deliver:to:startingAt:  
SubscriptionRegistry>>deliver:to:  
SubscriptionRegistry>>deliver:  
Announcer>>announce:  
RubEditingArea(RubAbstractTextArea)>>announce:  
RubEditingArea(RubAbstractTextArea)>>acceptContents  
RubSmalltalkEditor(RubTextEditor)>>accept  
RubTextEditor class>>buildShortcutsOn: in Block: [ :target | target editor accept ]  
BlockClosure>>cull:  
BlockClosure>>cull:cull:  
BlockClosure>>cull:cull:cull:  
KMCategoryBinding>>completeMatch:buffer:  
KMKeymap>>notifyCompleteMatchTo:buffer: in Block: [:l | l completeMatch: self buffer ]
```

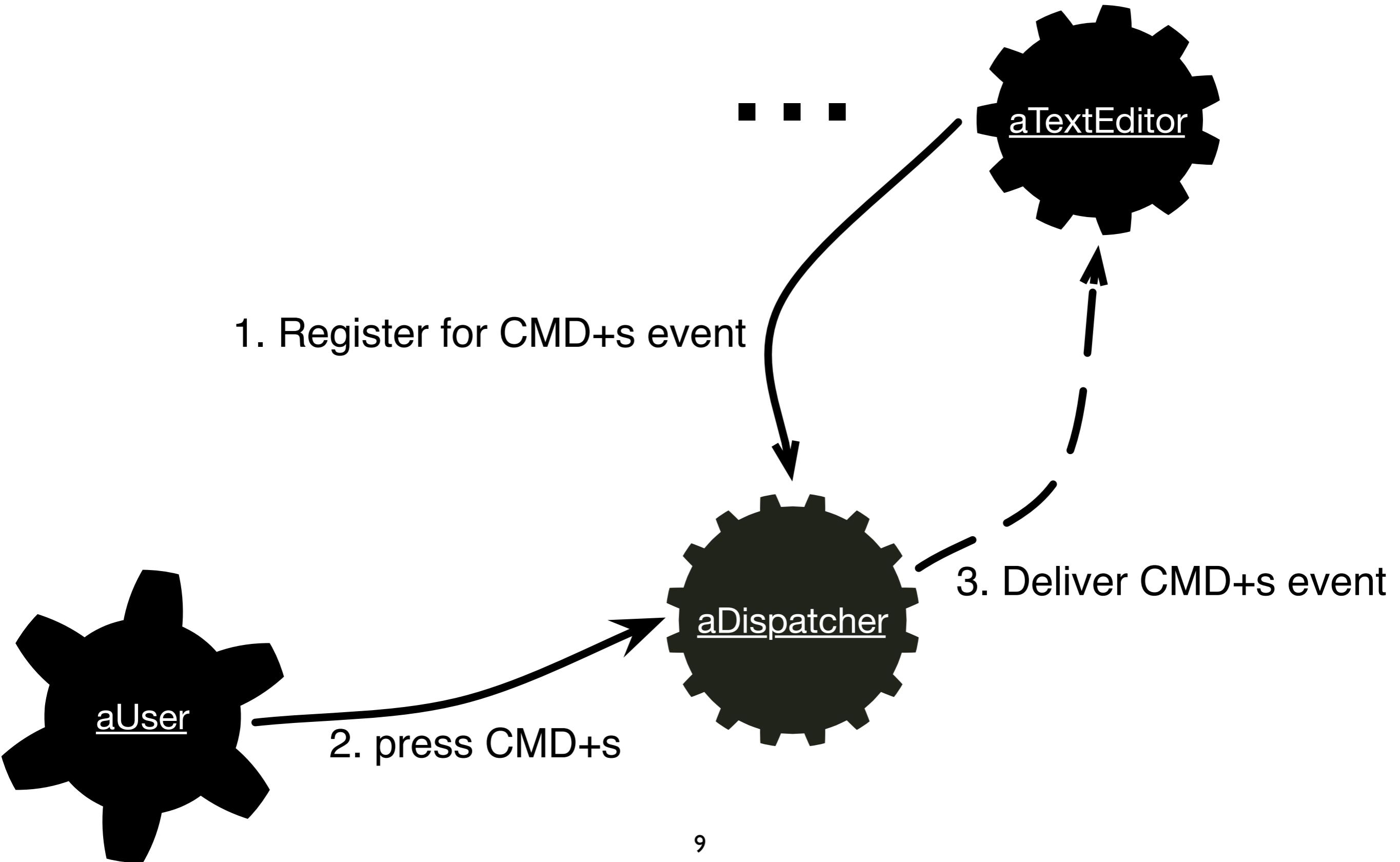


Mind the abstraction gap

# A publisher/subscriber framework



# Pressing CMD+s in an editor



# Halt

Stack

RubScrolledTextMorph&gt;&gt;whenTextAcceptRequest:

MessageSend&gt;&gt;value:

MessageSend&gt;&gt;cull:

MessageSend&gt;&gt;cull:cull:

AnnouncementSubscription&gt;&gt;deliver: in Block: [ acti

BlockClosure&gt;&gt;on:do:

BlockClosure&gt;&gt;on:fork:

AnnouncementSubscription&gt;&gt;deliver:

SubscriptionRegistry&gt;&gt;deliver:to:startingAt: in Block

BlockClosure&gt;&gt;ifCurtailed:

SubscriptionRegistry&gt;&gt;deliver:to:startingAt:

SubscriptionRegistry&gt;&gt;deliver:to:

SubscriptionRegistry&gt;&gt;deliver:

50 / 59

Source

```
whenTextAcceptRequest: anAnnouncement
    self halt.
    self announcer announce: anAnnouncement.
    self acceptContents
```

Type	Variable	Value
attribute	_self	a RubScrolledTextMorph(369885184)
parameter	_stack top	a RubTextAcceptRequest
attribute	_thisContext	RubScrolledTextMorph>>whenTextAcceptRequest:
attribute	alwaysAccept	nil
parameter	anAnnouncement	a RubTextAcceptRequest
attribute	autoAccept	false
temp	l	(600000)----- (670000770)

## Halt

Stack



Source



RubScrolledTextMorph&gt;&gt;whenTextAcceptRequest:

MessageSend&gt;&gt;value:

MessageSend&gt;&gt;cull:

MessageSend&gt;&gt;cull:cull:

AnnouncementSubscription&gt;&gt;deliver: in Block: [ acti

BlockClosure&gt;&gt;on:do:

BlockClosure&gt;&gt;on:fork:

AnnouncementSubscription&gt;&gt;deliver:

SubscriptionRegistry&gt;&gt;deliver:to:startingAt: in Block

BlockClosure&gt;&gt;ifCurtailed:

SubscriptionRegistry&gt;&gt;deliver:to:startingAt:

SubscriptionRegistry&gt;&gt;deliver:to:

SubscriptionRegistry&gt;&gt;deliver:

50 / 59

```
whenTextAcceptRequest: anAnnouncement
    self halt.
    self announcer announce: anAnnouncement.
    self acceptContents
```

Type	Variable	Value
attribute	_self	a RubScrolledTextMorph(369885184)
parameter	_stack top	a RubTextAcceptRequest
attribute	_thisContext	RubScrolledTextMorph>>whenTextAcceptRequest:
parameter	alwaysAccept	nil
attribute	anAnnouncement	a RubTextAcceptRequest
attribute	autoAccept	false
temp	l	(600000)----- (670000770)

# Halt

Stack



Source

**RubScrolledTextMorph>>whenTextAcceptRequest:**

MessageSend&gt;&gt;value:

MessageSend&gt;&gt;cull:

MessageSend&gt;&gt;cull:cull:

AnnouncementSubscription&gt;&gt;deliver: in Block: [ acti

BlockClosure&gt;&gt;on:do:

BlockClosure&gt;&gt;on:fork:

AnnouncementSubscription&gt;&gt;deliver:

SubscriptionRegistry&gt;&gt;deliver:to:startingAt: in Block

BlockClosure&gt;&gt;ifCurtailed:

SubscriptionRegistry&gt;&gt;deliver:to:startingAt:

SubscriptionRegistry&gt;&gt;deliver:to:

SubscriptionRegistry&gt;&gt;deliver:

50 / 59

```
whenTextAcceptRequest: anAnnouncement
    self halt.
    self announcer announce: anAnnouncement.
    self acceptContents
```

Type	Variable	Value
attribute	_self	a RubScrolledTextMorph(369885184)
parameter	_stack top	a RubTextAcceptRequest
attribute	_thisContext	RubScrolledTextMorph>>whenTextAcceptRequest:
parameter	alwaysAccept	nil
attribute	anAnnouncement	a RubTextAcceptRequest
attribute	autoAccept	false
temp	l	(600000)----- (670000770)

# Halt

Stack

```
BlockClosure>>on:do:  
BlockClosure>>on:fork:  
AnnouncementSubscription>>deliver:  
SubscriptionRegistry>>deliver:to:startingAt: in Block  
BlockClosure>>ifCurtailed:  
SubscriptionRegistry>>deliver:to:startingAt:  
SubscriptionRegistry>>deliver:to:  
SubscriptionRegistry>>deliver:  
Announcer>>announce:  
RubEditingArea(RubAbstractTextArea)>>announce:  
RubEditingArea(RubAbstractTextArea)>>acceptContent
```

50 / 59

Source

## acceptContents

"The message is sent when the user hits enter or Cmd-S.  
Accept the current contents and end editing"

```
self announce: (RubTextAcceptRequest morph: self).  
self changed
```

Type	Variable	Value
	_self	a RubEditingArea(541589504)
temp	_stack top	nil
temp	_thisContext	RubEditingArea(RubAbstractTextArea)>>acceptContents
temp	bounds	(0@0) corner: (664@31)
temp	color	Color transparent
temp	cursor	a RubCursor(101711872)
	selection	a Selection(101711872)

## Halt

Stack

BlockClosure>>on:do:  
 BlockClosure>>on:fork:  
 AnnouncementSubscription>>deliver:  
 SubscriptionRegistry>>deliver:to:startingAt: in Block  
 BlockClosure>>ifCurtailed:  
 SubscriptionRegistry>>deliver:to:startingAt:  
**SubscriptionRegistry>>deliver:to:**  
 SubscriptionRegistry>>deliver:  
 Announcer>>announce:  
 RubEditingArea(RubAbstractTextArea)>>announce:  
 RubEditingArea(RubAbstractTextArea)>>acceptCont  
 RubSmalltalkEditor(RubTextEditor)>>accept  
 RubTextEditor class>>buildShortcutsOn: in Block: [ :  
 < 50 / 59

Source

```
deliver: anAnnouncement to: subs
  ^ self deliver: anAnnouncement to: subs startingAt: 1
```

Type	Variable	Value
	_self	a SubscriptionRegistry
	_stack top	an Array(an Announcement...nouncementSubscription)
	_thisContext	SubscriptionRegistry>>deliver:to:
parameter	anAnnouncement	a RubTextAcceptRequest
attribute	monitor	a Semaphore()
parameter	subs	an Array(an Announcement...nouncementSubscription)
attribute		

# Halt

Stack

```
BlockClosure>>on:do:  
BlockClosure>>on:fork:  
AnnouncementSubscription>>deliver:  
SubscriptionRegistry>>deliver:to:startingAt: in Block  
BlockClosure>>ifCurtailed:  
SubscriptionRegistry>>deliver:to:startingAt:  
SubscriptionRegistry>>deliver:to:  
SubscriptionRegistry>>deliver:  
Announcer>>announce:  
RubEditingArea(RubAbstractTextArea)>announce:  
RubEditingArea(RubAbstractTextArea)>acceptCont  
RubSmalltalkEditor(RubTextEditor)>>accept  
RubTextEditor class>>buildShortcutsOn: in Block: [:  
50 / 59
```

Source

```
deliver: anAnnouncement to: subs  
  ^ self deliver: anAnnouncement to: subs startingAt: 1
```

Type	Variable	Value
	_self	a SubscriptionRegistry
	_stack top	an Array(an Announcement...nouncementSubscription)
	_thisContext	SubscriptionRegistry>>deliver:to:
parameter	anAnnouncement	a RubTextAcceptRequest
attribute	monitor	a Semaphore()
parameter	<b>subs</b>	<b>an Array(an Announcement...nouncementSubscription)</b>

# Halt

Stack



Source



```
BlockClosure>>on:do:  
BlockClosure>>on:fork:  
AnnouncementSubscription>>deliver:  
SubscriptionRegistry>>deliver:to:startingAt: in Block  
BlockClosure>>ifCurtailed:  
SubscriptionRegistry>>deliver:to:startingAt:  
SubscriptionRegistry>>deliver:to:  
SubscriptionRegistry>>deliver:  
Announcer>>announce:  
RubEditingArea(RubAbstractTextArea)>>announce:  
RubEditingArea(RubAbstractTextArea)>>acceptCont  
RubSmalltalkEditor(RubTextEditor)>>accept  
RubTextEditor class>>buildShortcutsOn: in Block: [:  
]
```

50 / 59

```
deliver: anAnnouncement to: subs  
  ^ self deliver: anAnnouncement to: subs startingAt: 1
```

Type	Variable	Value	State	Evaluator	2 items	Meta
parameter	_stackTop	an Array(an AnnouncementSubscription)	Item			Index
attribute	_thisContext	SubscriptionRegistry	an AnnouncementSubscription		1	
parameter	anAnnouncement	a RubTextAccept	an AnnouncementSubscription		2	
attribute	monitor	a Semaphore()				
parameter	subs	an Array(an AnnouncementSubscription)				

# Announcement Centric Debugger

a RubScrolledTextMorph

✓ ▶ ✕ ⏪ ⏴ ⏵ ⏶ 🔍 ⌂

**whenTextAcceptRequest:** anAnnouncement

```
self halt.  
self announcer announce: anAnnouncement.  
self acceptContents
```

Subscriptions

Announcement

Announcer

Announcement Type

Action

**RubTextAcceptRequest** **MessageSend(#whenTextAcceptRequest: -> a RubTextAcceptRequest)**

RubTextAcceptRequest **MessageSend(#whenTextAcceptRequest: -> a RubTextAcceptRequest)**

a RubEditingArea

◀ ▶ 🔍 ▾

**acceptContents**

"The message is sent when the user hits enter or  
Cmd-S.

Accept the current contents and endediting"

```
self announce: (RubTextAcceptRequest  
morph: self).  
self changed
```

Stack

**RubScrolledTextMorph>>whenTextAcceptRequest:**

MessageSend>>value:

MessageSend>>cull:

MessageSend>>cull:cull:

AnnouncementSubscription>>deliver: in Block: [ action cu

BlockClosure>>on:do:

BlockClosure>>on:fork:

AnnouncementSubscription>>deliver:

SubscriptionRegistry>>deliver:to:startingAt: in Block: [ sub

BlockClosure>>ifCurtailed:

SubscriptionRegistry>>deliver:to:startingAt:

SubscriptionRegistry>>deliver:to:

SubscriptionRegistry>>deliver:

Announcer>>announce:

RubEditingArea(RubAbstractTextArea)>>announce:

**RubEditingArea(RubAbstractTextArea)>>acceptContent**

RubSmalltalkEditor(RubTextEditor)>>accept

RubTextEditor class>>buildShortcutsOn: in Block: [ :target

BlockClosure>>cull:

BlockClosure>>cull:cull:

BlockClosure>>cull:cull:cull:

# Announcement Centric Debugger

```
a RubScrolledTextMorph ✓ ▶ ✕ ⏪ ⏴ ⏵ ⏷ 🔍 ⏹
```

**whenTextAcceptRequest:** anAnnouncement  
self halt.  
self announcer announce: anAnnouncement.  
self acceptContents

Subscriptions	Announcement	Announcer
Announcement Type	Action	
RubTextAcceptRequest	MessageSend(#whenTextAcceptRequest: -> a F)	
RubTextAcceptRequest	MessageSend(#whenTextAcceptRequest: -> a Ru	

```
a RubEditingArea ← → 🔍 ↻
```

**acceptContents**  
"The message is sent when the user hits enter or  
Cmd-S.  
Accept the current contents and endediting"  
  
self announce: (RubTextAcceptRequest  
morph: self).  
self changed

## Stack

```
RubScrolledTextMorph>>whenTextAcceptRequest:  
MessageSend>>value:  
MessageSend>>cull:  
MessageSend>>cull:cull:  
AnnouncementSubscription>>deliver: in Block: [ action cu  
BlockClosure>>on:do:  
BlockClosure>>on:fork:  
AnnouncementSubscription>>deliver:  
SubscriptionRegistry>>deliver:to:startingAt: in Block: [ sub  
BlockClosure>>ifCurtailed:  
SubscriptionRegistry>>deliver:to:startingAt:  
SubscriptionRegistry>>deliver:to:  
SubscriptionRegistry>>deliver:  
Announcer>>announce:  
RubEditingArea(RubAbstractTextArea)>>announce:  
RubEditingArea(RubAbstractTextArea)>>acceptContent  
RubSmalltalkEditor(RubTextEditor)>>accept  
RubTextEditor class>>buildShortcutsOn: in Block: [ :target  
BlockClosure>>cull:  
BlockClosure>>cull:cull:  
BlockClosure>>cull:cull:cull:
```

# Announcement Centric Debugger

a RubScrolledTextMorph

✓ ▶ ✕ 🔍 ⌂ ⌃ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ⌊ ⌋ ⌊ ⌋

**whenTextAcceptRequest:** anAnnouncement

self halt.

self announcer announce: anAnnouncement.

self acceptContents

Subscriptions

Announcement

Announcer

Announcement Type

Action

RubTextAcceptRequest MessageSend(#whenTextAcceptRequest: -> a F

RubTextAcceptRequest MessageSend(#whenTextAcceptRequest: -> a Ru

a RubEditingArea

◀ ▶ 🔍 ▾

**acceptContents**

"The message is sent when the user hits enter or  
Cmd-S.

Accept the current contents and endediting"

self announce: (RubTextAcceptRequest  
morph: self).

self changed

Stack

RubScrolledTextMorph>>whenTextAcceptRequest:

MessageSend>>value:

MessageSend>>cull:

MessageSend>>cull:cull:

AnnouncementSubscription>>deliver: in Block: [ action cu

BlockClosure>>on:do:

BlockClosure>>on:fork:

AnnouncementSubscription>>deliver:

SubscriptionRegistry>>deliver:to:startingAt: in Block: [ sub

BlockClosure>>ifCurtailed:

SubscriptionRegistry>>deliver:to:startingAt:

SubscriptionRegistry>>deliver:to:

SubscriptionRegistry>>deliver:

Announcer>>announce:

RubEditingArea(RubAbstractTextArea)>>announce:

**RubEditingArea(RubAbstractTextArea)>>acceptContent**

RubSmalltalkEditor(RubTextEditor)>>accept

RubTextEditor class>>buildShortcutsOn: in Block: [ :target

BlockClosure>>cull:

BlockClosure>>cull:cull:

BlockClosure>>cull:cull:cull:

# Announcement Centric Debugger

a RubScrolledTextMorph



**whenTextAcceptRequest:** anAnnouncement

```
self halt.  
self announcer announce: anAnnouncement.  
self acceptContents
```

Subscriptions

Announcement

Announcer

Announcement Type

Action

RubTextAcceptRequest MessageSend(#whenTextAcceptRequest: -> a F

RubTextAcceptRequest MessageSend(#whenTextAcceptRequest: -> a Ru

a RubEditingArea



**acceptContents**

"The message is sent when the user hits enter or  
Cmd-S.

Accept the current contents and endediting"

```
self announce: (RubTextAcceptRequest  
morph: self).  
self changed
```

Stack

RubScrolledTextMorph>>whenTextAcceptRequest:

MessageSend>>value:

MessageSend>>cull:

MessageSend>>cull:cull:

AnnouncementSubscription>>deliver: in Block: [ action cu

BlockClosure>>on:do:

BlockClosure>>on:fork:

AnnouncementSubscription>>deliver:

SubscriptionRegistry>>deliver:to:startingAt: in Block: [ sub

BlockClosure>>ifCurtailed:

SubscriptionRegistry>>deliver:to:startingAt:

SubscriptionRegistry>>deliver:to:

SubscriptionRegistry>>deliver:

Announcer>>announce:

RubEditingArea(RubAbstractTextArea)>>announce:

**RubEditingArea(RubAbstractTextArea)>>acceptContent**

RubSmalltalkEditor(RubTextEditor)>>accept

RubTextEditor class>>buildShortcutsOn: in Block: [ :target

BlockClosure>>cull:

BlockClosure>>cull:cull:

BlockClosure>>cull:cull:cull:

a RubScrolledTextMorph

**whenTextAcceptRequest:** anAnnouncement

```
self halt.  
self announcer announce: anAnnouncement.  
self acceptContents
```

Subscriptions

Announcement

Announcer

Announcement Type

Action

RubTextAcceptRequest MessageSend(#whenTextAcceptRequest: -&gt; a F

RubTextAcceptRequest MessageSend(#whenTextAcceptRequest: -&gt; a Rut

a RubEditingArea

**acceptContents**

"The message is sent when the user hits enter or  
Cmd-S.

Accept the current contents and endediting"

```
self announce: (RubTextAcceptRequest  
morph: self).  
self changed
```

Stack

**RubScrolledTextMorph>>whenTextAcceptRequest:**

MessageSend&gt;&gt;value:

MessageSend&gt;&gt;cull:

MessageSend&gt;&gt;cull:cull:

AnnouncementSubscription&gt;&gt;deliver: in Block: [ action cu

BlockClosure&gt;&gt;on:do:

BlockClosure&gt;&gt;on:fork:

AnnouncementSubscription&gt;&gt;deliver:

SubscriptionRegistry&gt;&gt;deliver:to:startingAt: in Block: [ sub

BlockClosure&gt;&gt;ifCurtailed:

SubscriptionRegistry&gt;&gt;deliver:to:startingAt:

SubscriptionRegistry&gt;&gt;deliver:to:

SubscriptionRegistry&gt;&gt;deliver:

Announcer&gt;&gt;announce:

RubEditingArea(RubAbstractTextArea)&gt;&gt;announce:

**RubEditingArea(RubAbstractTextArea)>>acceptContent**

RubSmalltalkEditor(RubTextEditor)&gt;&gt;accept

RubTextEditor class&gt;&gt;buildShortcutsOn: in Block: [ :target

BlockClosure&gt;&gt;cull:

BlockClosure&gt;&gt;cull:cull:

BlockClosure&gt;&gt;cull:cull:cull:

# Announcement Centric Debugger

a RubScrolledTextMorph

✓ ▶ ✕ 🔍 ⌂ ⌃ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ⌊ ⌋ ⌊ ⌋

**whenTextAcceptRequest:** anAnnouncement

self halt.

self announcer announce: anAnnouncement.

self acceptContents

Subscriptions

Announcement

Announcer



Announcement Type

Action

**RubTextAcceptRequest** **MessageSend(#whenTextAcceptRequest: -> a F**

RubTextAcceptRequest MessageSend(#whenTextAcceptRequest: -> a Ru

a RubEditingArea

◀ ▶ 🔍 ⌂

**acceptContents**

"The message is sent when the user hits enter or

Cmd-S.

Accept the current contents and endediting"

self announce: (RubTextAcceptRequest  
morph: self).

self changed

Stack

RubScrolledTextMorph>>**whenTextAcceptRequest:**

MessageSend>>value:

MessageSend>>cull:

MessageSend>>cull:cull:

AnnouncementSubscription>>deliver: in Block: [ action cu

BlockClosure>>on:do:

BlockClosure>>on:fork:

AnnouncementSubscription>>deliver:

SubscriptionRegistry>>deliver:to:startingAt: in Block: [ sub

BlockClosure>>ifCurtailed:

SubscriptionRegistry>>deliver:to:startingAt:

SubscriptionRegistry>>deliver:to:

SubscriptionRegistry>>deliver:

Announcer>>announce:

RubEditingArea(RubAbstractTextArea)>>announce:

**RubEditingArea(RubAbstractTextArea)>>acceptContent**

RubSmalltalkEditor(RubTextEditor)>>accept

RubTextEditor class>>buildShortcutsOn: in Block: [ :target

BlockClosure>>cull:

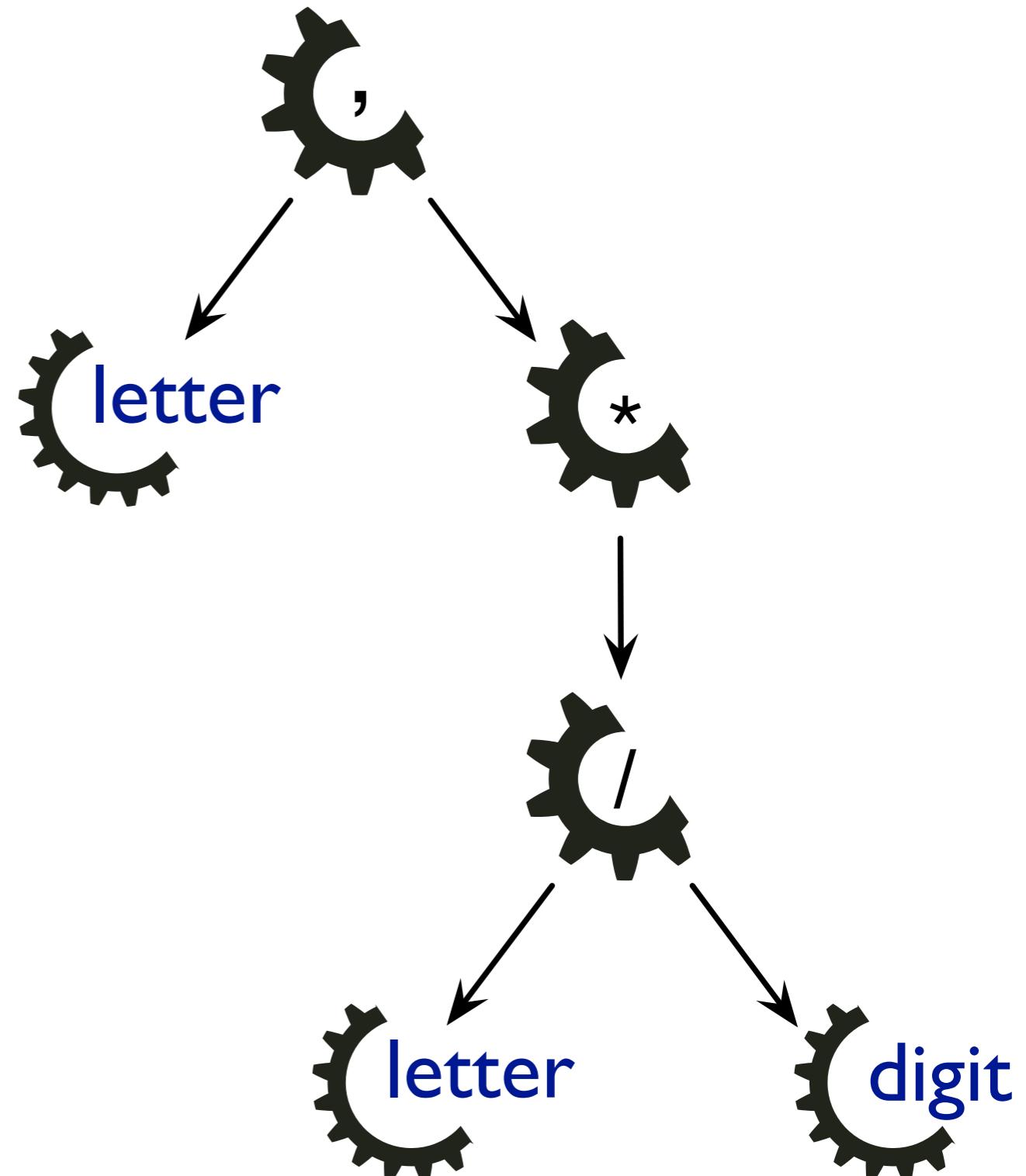
BlockClosure>>cull:cull:

BlockClosure>>cull:cull:cull:

# PetitParser

identifier

letter , (letter / digit) \*



```
IdentifierParser new  
parse: 'aLong32Identifier'
```

# Default Debugger

Stack

PPStream(ReadStream)>>next  
PPContext>>next  
PPPredicateObjectParser>>parseOn:  
PPDelegateParser>>parseOn:  
PPChoiceParser>>parseOn:  
PPPossessiveRepeatingParser>>parseOn:  
PPSequenceParser>>parseOn:  
PPDelegateParser>>parseOn:  
PPEndOfInputParser>>parseOn:  
PPIIdentifierParser(PPDelegateParser)>>parseOn:  
PPIIdentifierParser(PPParser)>>parseWithContext:  
PPIIdentifierParser(PPParser)>>parse:withContext:  
PPIIdentifierParser(PPParser)>>parse:

Source

parseOn: aPPContext  
  ^ parser parseOn: aPPContext

Type	Variable	Value
	_self	a PPDelegateParser(identifier)
	_stack top	a PPContext
	_thisContext	PPDelegateParser>>parseOn:
parameter	aPPContext	a PPContext
attribute	parser	a PPSequenceParser(273678336)
temp	properties	a Dictionary(#name->#identifier )

## Default Debugger

Stack

```

PPStream(ReadStream)>>next
PPContext>>next
PPPredicateObjectParser>>parseOn:
PPDelegateParser>>parseOn:
PPChoiceParser>>parseOn:
PPPossessiveRepeatingParser>>parseOn:
PPSequenceParser>>parseOn:
PPDelegateParser>>parseOn:
PPEndOfInputParser>>parseOn:
PPIIdentifierParser(PPDelegateParser)>>parseOn:
PPIIdentifierParser(PPParser)>>parseWithContext:
PPIIdentifierParser(PPParser)>>parse:withContext:
PPIIdentifierParser(PPParser)>>parse:
  
```

Source

```

parseOn: aPPContext
  ^ parser parseOn: aPPContext
  
```

Type	Variable	Value
	_self	a PPDelegateParser(identifier)
	_stack top	a PPContext
	_thisContext	PPDelegateParser>>parseOn:
parameter	aPPContext	a PPContext
attribute	parser	a PPSequenceParser(273678336)
temp	properties	a Dictionary(#name->#identifier )

## Default Debugger

Stack

```

PPStream(ReadStream)>>next
PPContext>>next
PPPredicateObjectParser>>parseOn:
PPDelegateParser>>parseOn:
PPChoiceParser>>parseOn:
PPPossessiveRepeatingParser>>parseOn:
PPSequenceParser>>parseOn:
PPDelegateParser>>parseOn:
PPEndOfInputParser>>parseOn:
PPIIdentifierParser(PPDelegateParser)>>parseOn:
PPIIdentifierParser(PPParser)>>parseWithContext:
PPIIdentifierParser(PPParser)>>parse:withContext:
PPIIdentifierParser(PPParser)>>parse:

```

Source

```

parseOn: aPPContext
  ^ parser parseOn: aPPContext

```

Type	Variable	Value
	_self	a PPDelegateParser(identifier)
	_stack top	a PPContext
	_thisContext	PPDelegateParser>>parseOn:
parameter	aPPContext	a PPContext
attribute	parser	a PPSequenceParser(273678336)
temp	properties	a Dictionary(#name->#identifier )

# PetitParser Debugger

Stack

```

PPStream(ReadStream)>>next
PPContext>>next
PPPredicateObjectParser(129761280, 'digit expected'):
PPDelegateParser(digit)>>parseOn:
PPChoiceParser(1017118720)>>parseOn:
PPPossessiveRepeatingParser(214958080)>>parseOn:
PPSequenceParser(935854080)>>parseOn:
PPDelegateParser(identifier)>>parseOn:
PPEndOfInputParser(239861760)>>parseOn:
PIdentifierParser(PPDelegateParser)(471334912)>>pa
PIdentifierParser(PPParser)(471334912)>>parseWithC
PIdentifierParser(PPParser)(471334912)>>parse:withI
PIdentifierParser(PPParser)(471334912)>>parse:
IUndefinedObject>>DoIt
    
```

Source

**parseOn: aPPContext**

  ^ parser parseOn: aPPContext

Q

Stream

aLong32Identifier

Source

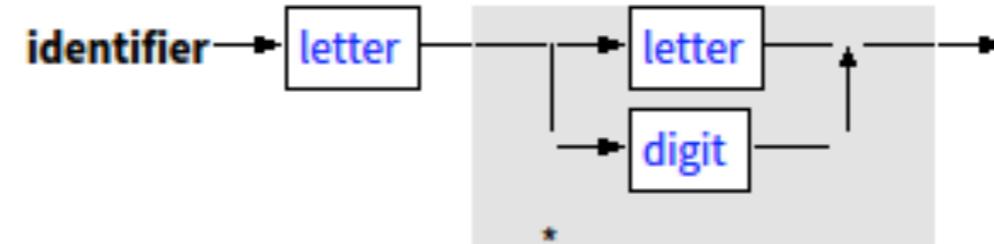
Graph

Map

Example

First

Follow



Type

Variable

Value

_self	a PPDelegateParser(identifier)
_stack top	a PPContext
_thisContext	PPDelegateParser>>parseOn:
parameter	a PPContext
attribute	a PPSequenceParser(935854080)
temp	a Dictionary(#name->#identifier )

# PetitParser Debugger

**Stack**

```

PPStream(ReadStream)>>next
PPContext>>next
PPPredicateObjectParser(129761280, 'digit expected'):
PPDelegateParser(digit)>>parseOn:
PPChoiceParser(1017118720)>>parseOn:
PPPossessiveRepeatingParser(214958080)>>parseOn:
PPSequenceParser(935854080)>>parseOn:
PPDelegateParser(identifier)>>parseOn:
PPEndOfInputParser(239861760)>>parseOn:
PIdentifierParser(PPDelegateParser)(471334912)>>pa
PIdentifierParser(PPParser)(471334912)>>parseWithC
PIdentifierParser(PPParser)(471334912)>>parse:withI
PIdentifierParser(PPParser)(471334912)>>parse:
IUndefinedObject>>DoIt
    
```

**Source**

```

parseOn: aPPContext
  ^ parser parseOn: aPPContext
  
```

**Stream**

```

aLong32Identifier
  
```

Source    Graph    Map    Example    First    Follow

```

graph LR
    identifier[identifier] --> letter1[letter]
    letter1 --> letter2[letter]
    letter1 --> digit[digit]
    digit --> letter2
    
```

Type	Variable	Value
	_self	a PPDelegateParser(identifier)
	_stack top	a PPContext
	_thisContext	PPDelegateParser>>parseOn:
parameter	aPPContext	a PPContext
attribute	parser	a PPSequenceParser(935854080)
temp	properties	a Dictionary(#name->#identifier )

# PetitParser Debugger

Stack

```

PPStream(ReadStream)>>next
PPContext>>next
PPPredicateObjectParser(129761280, 'digit expected'):
PPDelegateParser(digit)>>parseOn:
PPChoiceParser(1017118720)>>parseOn:
PPPossessiveRepeatingParser(214958080)>>parseOn:
PPSequenceParser(935854080)>>parseOn:
PPDelegateParser(identifier)>>parseOn:
PPEndOfInputParser(239861760)>>parseOn:
PIdentifierParser(PPDelegateParser)(471334912)>>parseOn:
PIdentifierParser(PPParser)(471334912)>>parseWithC
PIdentifierParser(PPParser)(471334912)>>parse:withI
PIdentifierParser(PPParser)(471334912)>>parse:
IUndefinedObject>>DoIt

```

Source

```

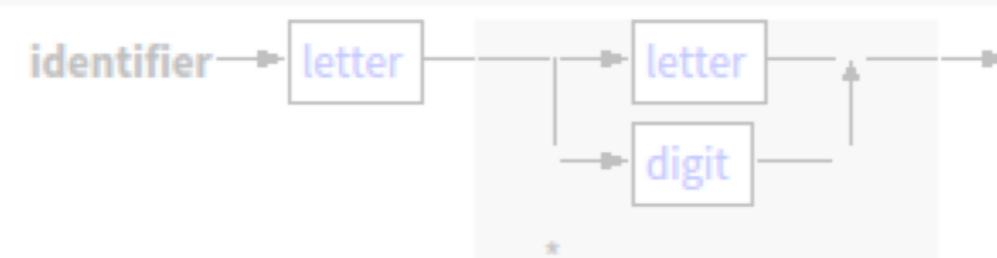
parseOn: aPPContext
^ parser parseOn: aPPContext

```

Stream

aLong32Identifier

Source Graph Map Example First Follow



Type	Variable	Value
	_self	a PPDelegateParser(identifier)
	_stack top	a PPContext
	_thisContext	PPDelegateParser>>parseOn:
parameter	aPPContext	a PPContext
attribute	parser	a PPSequenceParser(935854080)
temp	properties	a Dictionary(#name->#identifier )

# PetitParser Debugger

Stack

```

PPStream(ReadStream)>>next
PPContext>>next
PPPredicateObjectParser(129761280, 'digit expected'):
PPDelegateParser(digit)>>parseOn:
PPChoiceParser(1017118720)>>parseOn:
PPPossessiveRepeatingParser(214958080)>>parseOn:
PPSequenceParser(935854080)>>parseOn:
PPDelegateParser(identifier)>>parseOn:
PPEndOfInputParser(239861760)>>parseOn:
PPIdentifierParser(PPDelegateParser)(471334912)>>parseOn:
PPIdentifierParser(PPParser)(471334912)>>parseWithC:
PPIdentifierParser(PPParser)(471334912)>>parse:withI:
PPIdentifierParser(PPParser)(471334912)>>parse:
UndefinedObject>>DoIt

```

Source

```

parseOn: aPPContext
^ parser parseOn: aPPContext

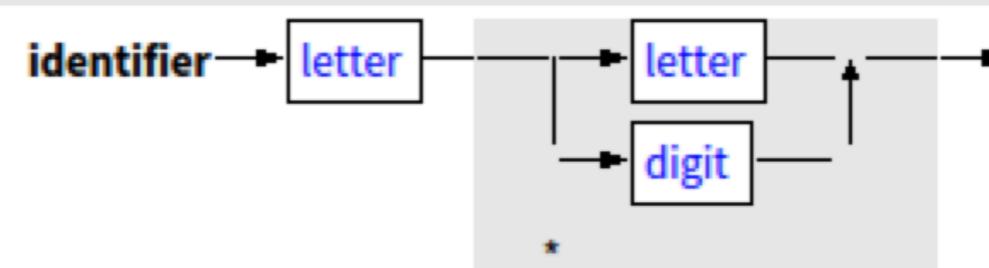
```

✓ ✕ ⌂ ⌂ ⌂

Stream

aLong32Identifier

Source Graph Map Example First Follow



Type	Variable	Value
	_self	a PPDelegateParser(identifier)
	_stack top	a PPContext
	_thisContext	PPDelegateParser>>parseOn:
parameter	aPPContext	a PPContext
attribute	parser	a PPSequenceParser(935854080)
temp	properties	a Dictionary(#name->#identifier )

# The Moldable Debugger

# Halt

Stack

RubScrolledTextMorph&gt;&gt;whenTextAcceptRequest:

MessageSend&gt;&gt;value:

MessageSend&gt;&gt;cull:

MessageSend&gt;&gt;cull:cull:

AnnouncementSubscription&gt;&gt;deliver: in Block: [ acti

BlockClosure&gt;&gt;on:do:

BlockClosure&gt;&gt;on:fork:

AnnouncementSubscription&gt;&gt;deliver:

SubscriptionRegistry&gt;&gt;deliver:to:startingAt: in Block

BlockClosure&gt;&gt;ifCurtailed:

SubscriptionRegistry&gt;&gt;deliver:to:startingAt:

SubscriptionRegistry&gt;&gt;deliver:to:

SubscriptionRegistry&gt;&gt;deliver:

50 / 59

Source

```
whenTextAcceptRequest: anAnnouncement
    self halt.
    self announcer announce: anAnnouncement.
    self acceptContents
```

Type	Variable	Value
attribute	_self	a RubScrolledTextMorph(369885184)
parameter	_stack top	a RubTextAcceptRequest
attribute	_thisContext	RubScrolledTextMorph>>whenTextAcceptRequest:
parameter	alwaysAccept	nil
attribute	anAnnouncement	a RubTextAcceptRequest
attribute	autoAccept	false
temp	l.....	(600000)----- (670000770)

# Debugging widgets

Stack

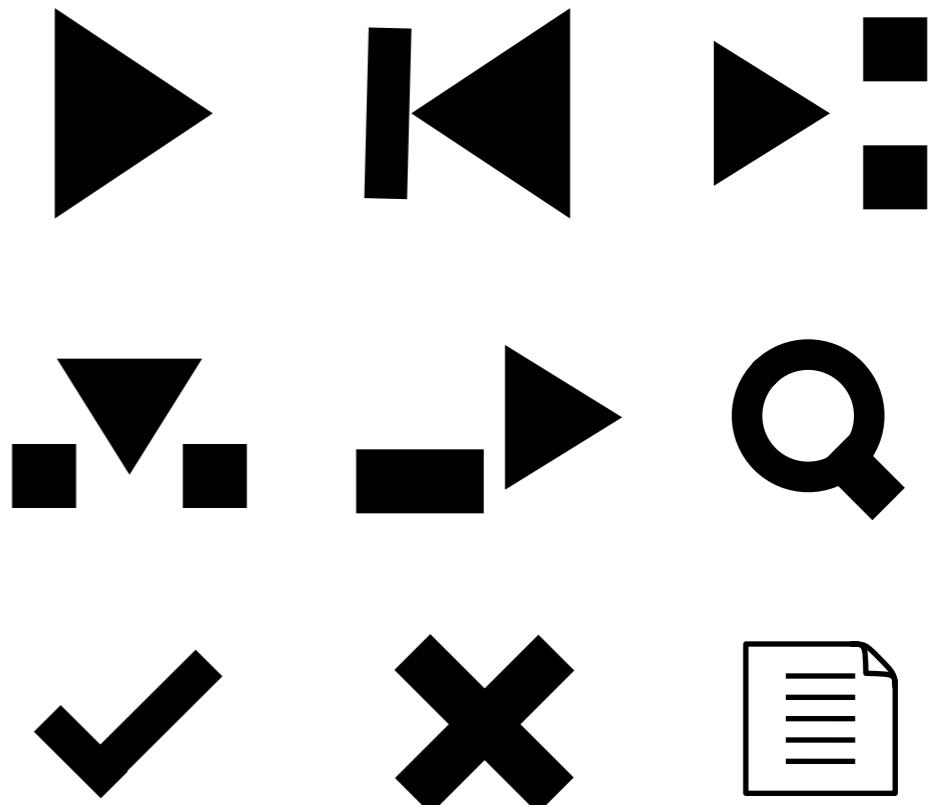
```
RubScrolledTextMorph>>whenTextAcceptRequest:  
MessageSend>>value:  
MessageSend>>cull:  
MessageSend>>cull:cull:  
AnnouncementSubscription>>deliver: in Block: [ act  
BlockClosure>>on:do:  
BlockClosure>>on:fork:  
AnnouncementSubscription>>deliver:  
SubscriptionRegistry>>deliver:to:startingAt: in Bloc  
BlockClosure>>ifCurtailed:  
SubscriptionRegistry>>deliver:to:startingAt:  
SubscriptionRegistry>>deliver:to:  
SubscriptionRegistry>>deliver:  
50 / 59
```

Source

```
whenTextAcceptRequest: anAnnouncement  
    self halt.  
    self announcer announce: anAnnouncement.  
    self acceptContents
```

Type	Variable	Value
	_self	a RubScrolledTextMorph(1019740160)
	_stack top	a RubTextAcceptRequest
	_thisContext	RubScrolledTextMorph>>whenTextAcceptRequest:
attribute	alwaysAccept	nil
parameter	anAnnouncement	a RubTextAcceptRequest
attribute	autoAccept	false
temp	bounds	(6.0@30.0) corner: (697.0@452.0)
temp	color	Color white

# Debugging actions



# Announcement Centric Debugger

a RubScrolledTextMorph

✓ ▶ ✕ ⏪ ⏵ ⏶ ⏷ ⏸ 🔍 ⏹

**whenTextAcceptRequest:** anAnnouncement

```
self halt.  
self announcer announce: anAnnouncement.  
self acceptContents
```

Subscriptions

Announcement

Announcer

Announcement Type

Action

**RubTextAcceptRequest** **MessageSend(#whenTextAcceptRequest: -> a RubTextAcceptRequest)**

RubTextAcceptRequest **MessageSend(#whenTextAcceptRequest: -> a RubTextAcceptRequest)**

a RubEditingArea

◀ ▶ 🔍 ▾

**acceptContents**

"The message is sent when the user hits enter or  
Cmd-S.

Accept the current contents and endediting"

```
self announce: (RubTextAcceptRequest  
morph: self).  
self changed
```

Stack

**RubScrolledTextMorph>>whenTextAcceptRequest:**

MessageSend>>value:

MessageSend>>cull:

MessageSend>>cull:cull:

AnnouncementSubscription>>deliver: in Block: [ action cu

BlockClosure>>on:do:

BlockClosure>>on:fork:

AnnouncementSubscription>>deliver:

SubscriptionRegistry>>deliver:to:startingAt: in Block: [ sub

BlockClosure>>ifCurtailed:

SubscriptionRegistry>>deliver:to:startingAt:

SubscriptionRegistry>>deliver:to:

SubscriptionRegistry>>deliver:

Announcer>>announce:

RubEditingArea(RubAbstractTextArea)>>announce:

**RubEditingArea(RubAbstractTextArea)>>acceptContent**

RubSmalltalkEditor(RubTextEditor)>>accept

RubTextEditor class>>buildShortcutsOn: in Block: [ :target

BlockClosure>>cull:

BlockClosure>>cull:cull:

BlockClosure>>cull:cull:cull:

# Next subscription

# Selected subscription

The screenshot shows the Smalltalk debugger interface with three main panes:

- Top Left:** A code editor pane showing the implementation of `whenTextAcceptRequest` in `RubScrolledTextMorph`. The code halts the receiver and announces to its announcer.
- Top Right:** A stack trace pane titled "Stack" showing the call sequence for `whenTextAcceptRequest`. It includes frames for `RubScrolledTextMorph>>whenTextAcceptRequest`, `MessageSend>>value`, `MessageSend>>cull`, `MessageSend>>cull:cull`, `AnnouncementSubscription>>deliver:in Block: [ action cull: anAnnouncement cull: announcer ]`, `BlockClosure>>on:do`, `BlockClosure>>on:fork`, `AnnouncementSubscription>>deliver`, `SubscriptionRegistry>>deliver:to:startingAt:in Block: [ subscription deliver: anAnnouncement ]`, `BlockClosure>>ifCurtailed`, `SubscriptionRegistry>>deliver:to:startingAt`, `SubscriptionRegistry>>deliver:to`, `SubscriptionRegistry>>deliver`, `Announcer>>announce`, `RubEditingArea(RubAbstractTextArea)>>announce`, and `RubEditingArea(RubAbstractTextArea)>>acceptContents`.
- Bottom Left:** A subscriptions pane titled "Subscriptions" showing two entries under "RubTextAcceptRequest". The first entry is `MessageSend(#whenTextAcceptRequest: -> a RubScrolledTextMorph(1019740160))` and the second is `MessageSend(#whenTextAcceptRequest: -> a RubShoutStylerDecorator)`.
- Bottom Right:** A code editor pane showing the implementation of `acceptContents` in `RubEditingArea`. It describes the message and its effect of accepting current contents and ending editing. It also shows the receiver announcing itself to a `RubTextAcceptRequest` morph and changing itself.

# PetitParser Debugger

Stack

```

PPStream(ReadStream)>>next
PPContext>>next
PPPredicateObjectParser(129761280, 'digit expected'):
PPDelegateParser(digit)>>parseOn:
PPChoiceParser(1017118720)>>parseOn:
PPPossessiveRepeatingParser(214958080)>>parseOn:
PPSequenceParser(935854080)>>parseOn:
PPDelegateParser(identifier)>>parseOn:
PPEndOfInputParser(239861760)>>parseOn:
PIdentifierParser(PPDelegateParser)(471334912)>>pa
PIdentifierParser(PPParser)(471334912)>>parseWithC
PIdentifierParser(PPParser)(471334912)>>parse:withI
PIdentifierParser(PPParser)(471334912)>>parse:
IUndefinedObject>>DoIt
    
```

Source

**parseOn: aPPContext**

  ^ parser parseOn: aPPContext

Q

Stream

aLong32Identifier

Source

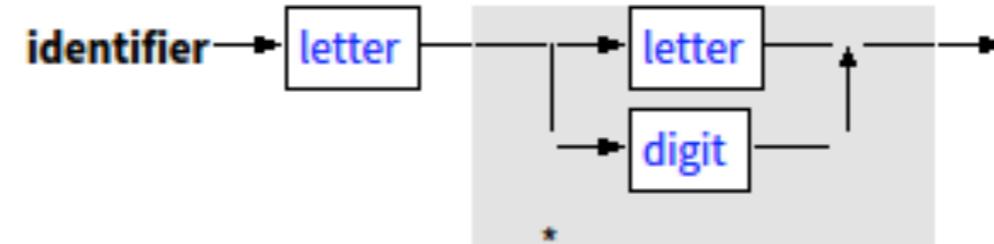
Graph

Map

Example

First

Follow



Type

Variable

Value

\_self

a PPDelegateParser(identifier)

\_stack top

a PPContext

\_thisContext

PPDelegateParser>>parseOn:

parameter

aPPContext

a PPContext

attribute

parser

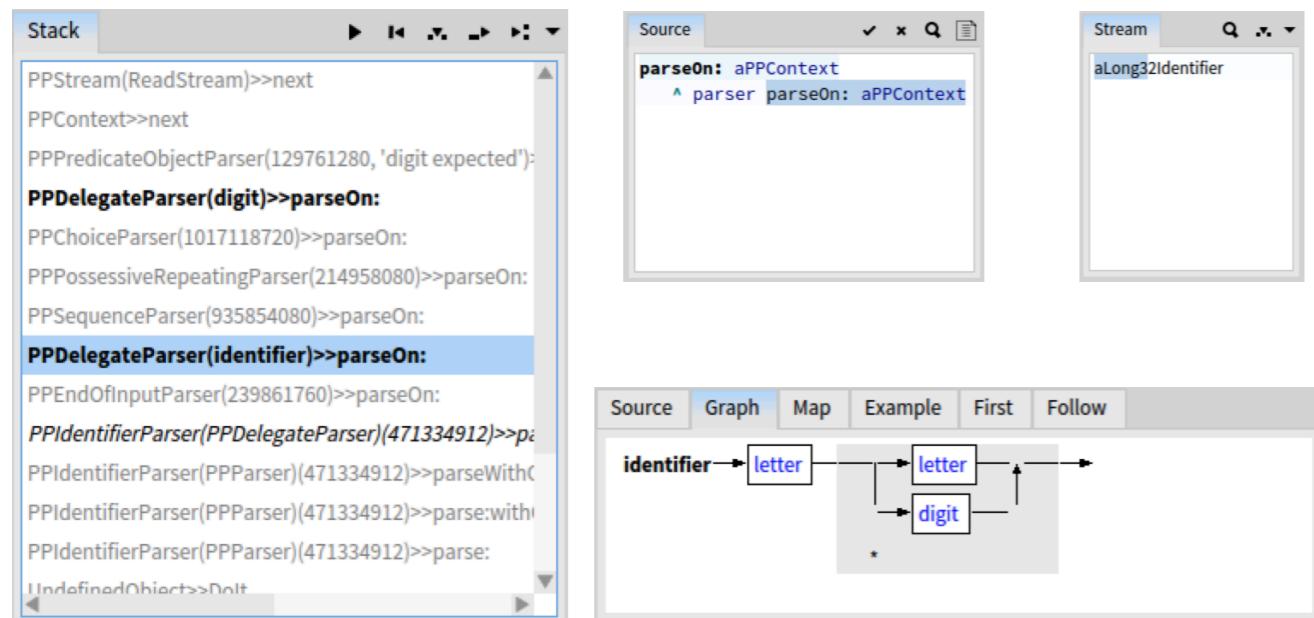
a PPSequenceParser(935854080)

temp

properties

a Dictionary(#name->#identifier )

## Next parser



## Next production

## Production(aproduction)

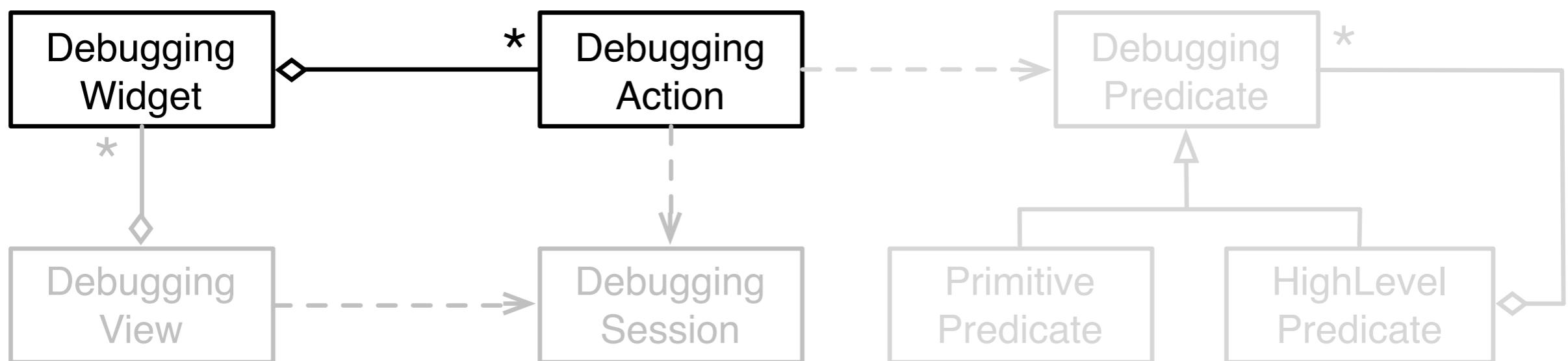
## Next failure

Type	Variable	Value
	<code>_self</code>	a PPDelegateParser(identifier)
	<code>_stack top</code>	a PPContext
	<code>_thisContext</code>	PPDelegateParser>>parseOn:
parameter	<code>aPPContext</code>	a PPContext
attribute	<code>parser</code>	a PPSequenceParser(935854080)
temp	<code>properties</code>	a Dictionary(#name->#identifier )

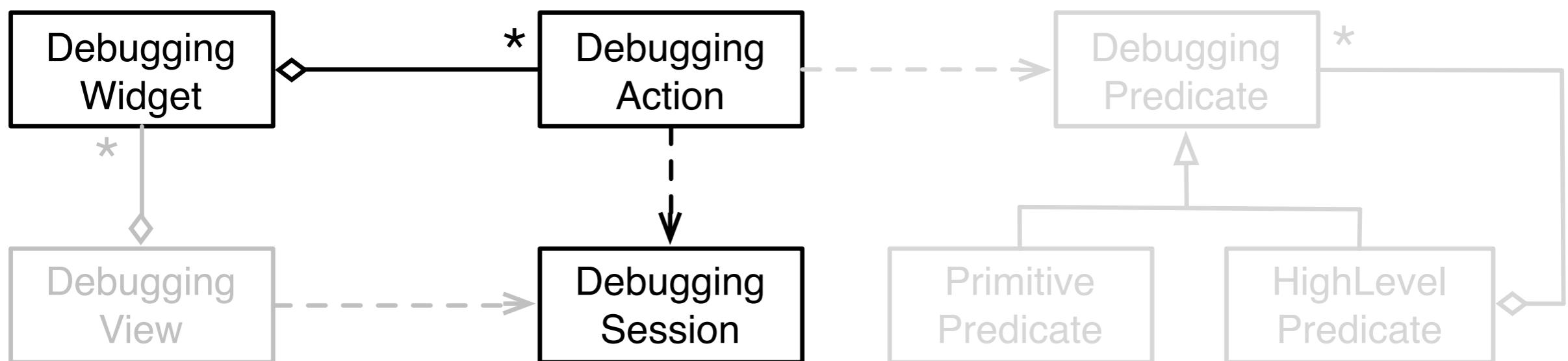
## Stream position(anInteger)

## Stream position changed

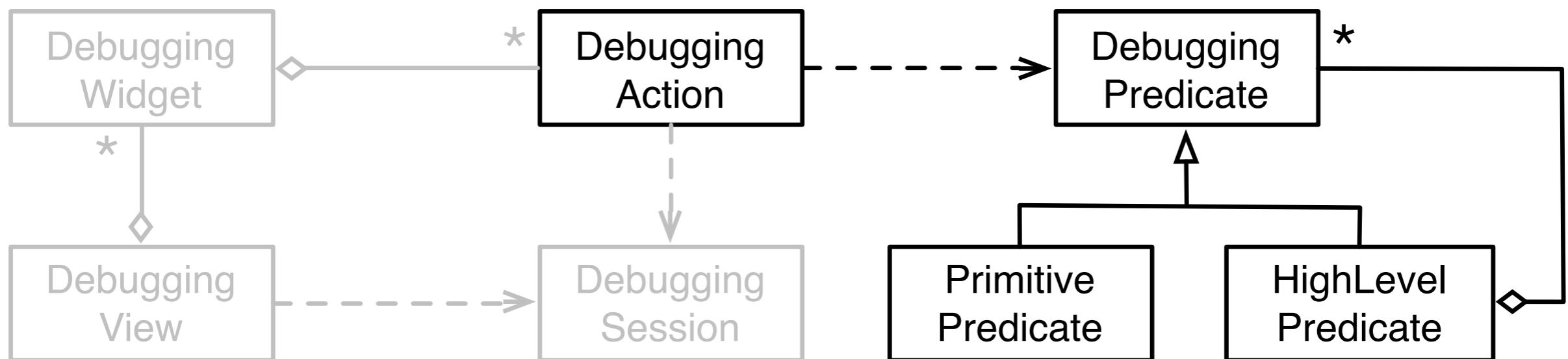
# Domain specific-extension



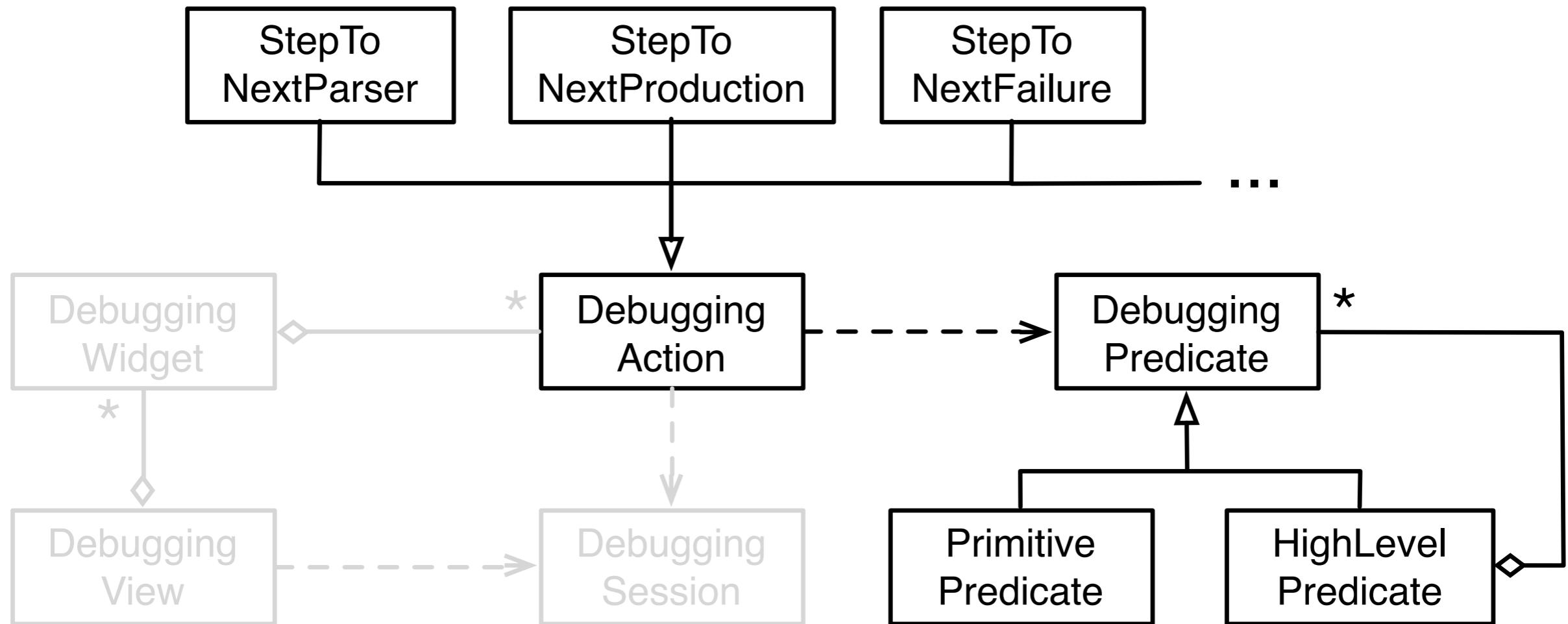
# Domain specific-extension



# Domain specific-extension

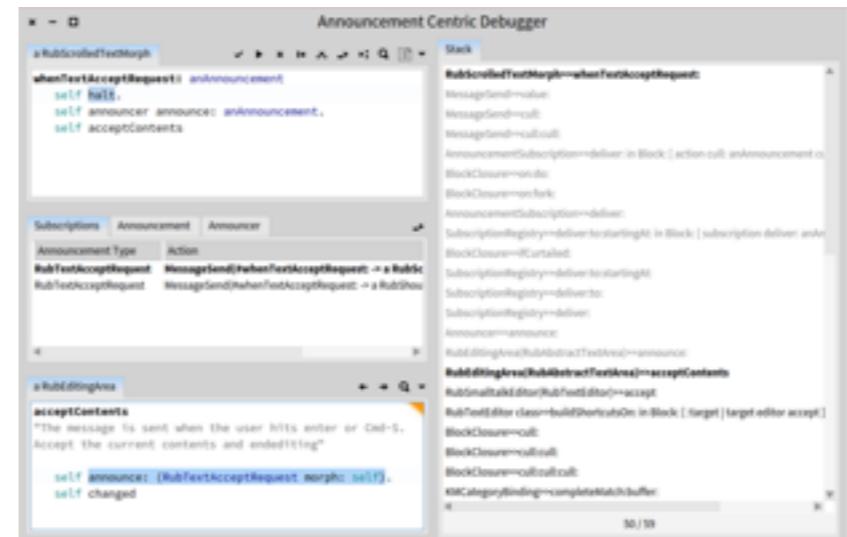
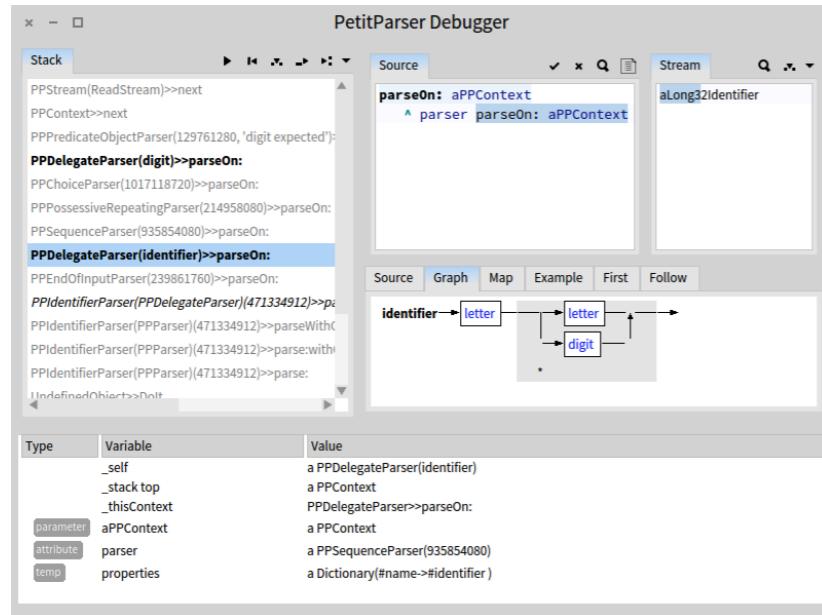


# Domain specific-extension



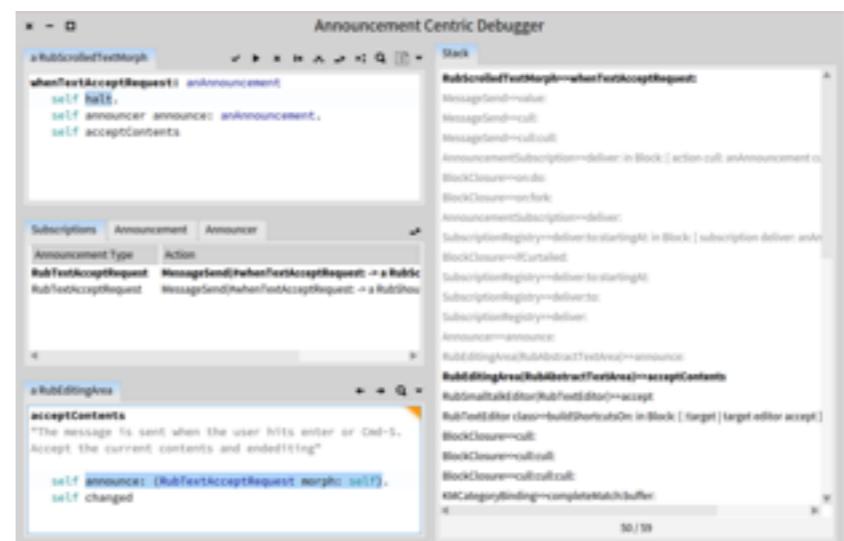
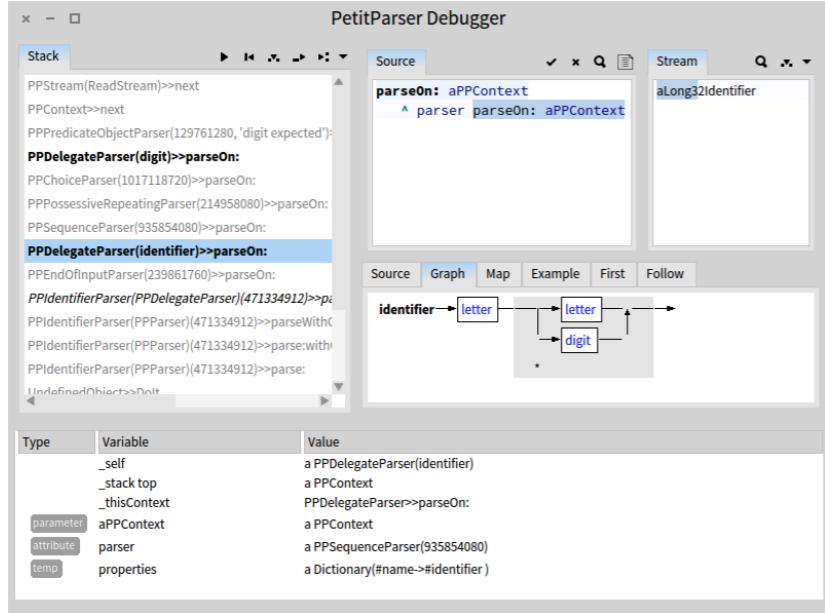
# Petit Parser

# Events

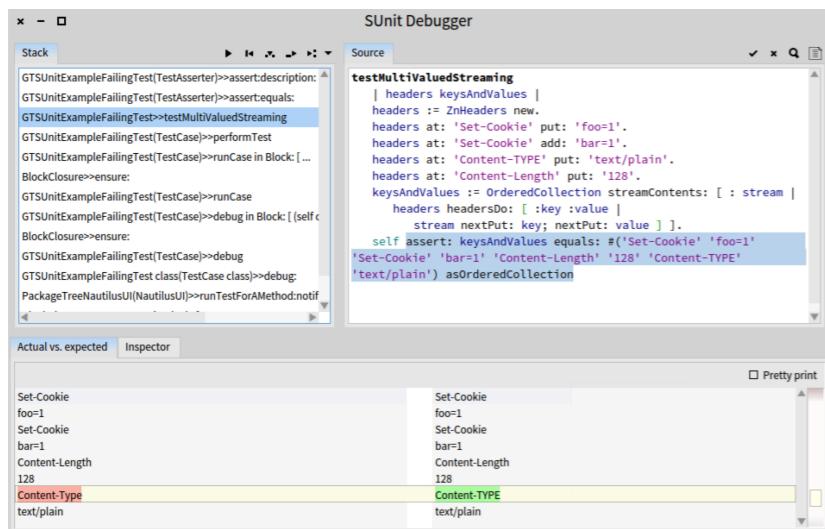


# Petit Parser

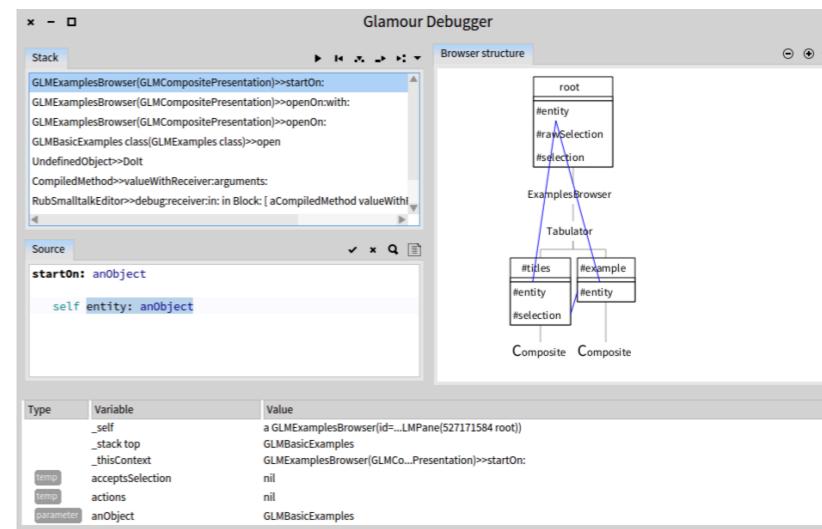
# Events



# SUnit

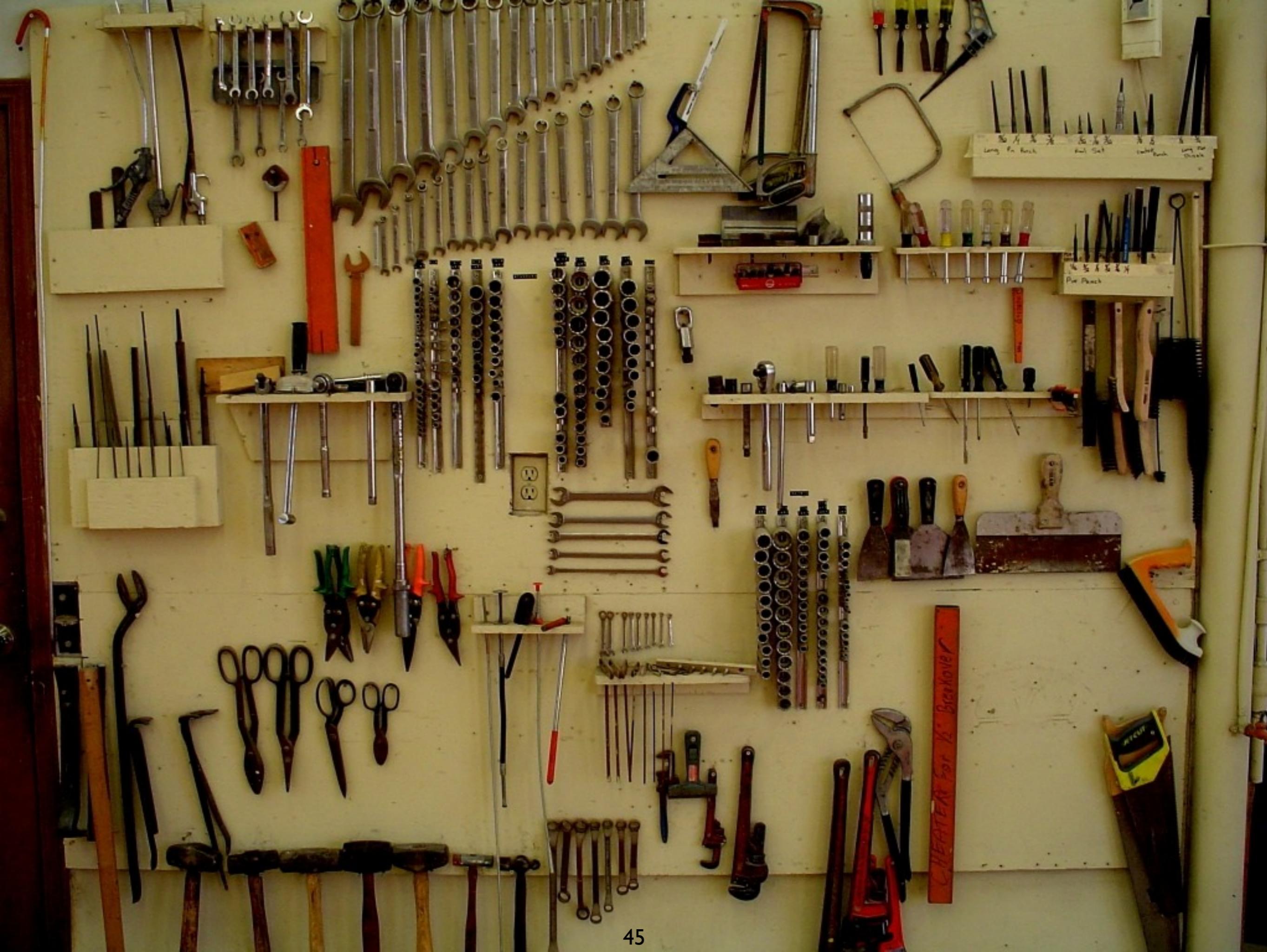


# Glamour

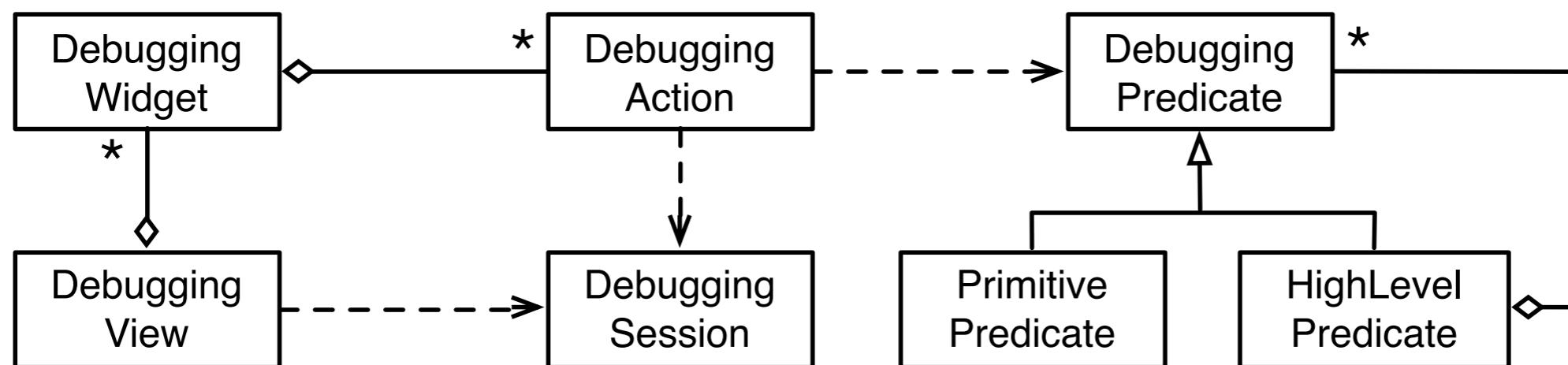


# Size of extensions in LOC

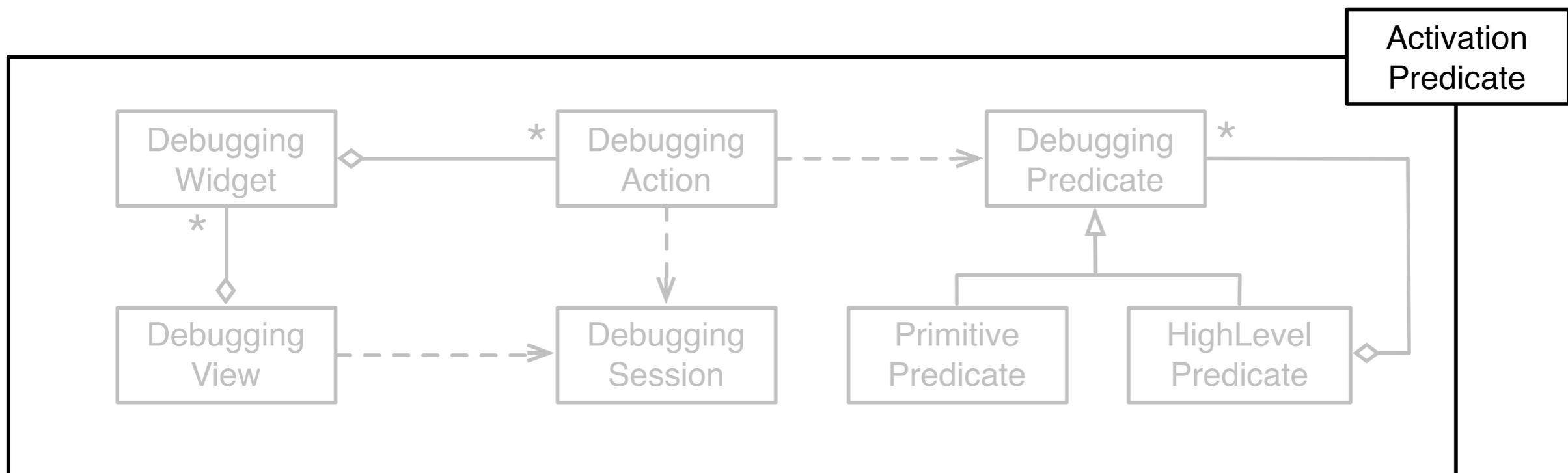
	Session	Operations	View	Total
Base model	800	700	-	1500
Default Debugger	-	100	400	500
Announcements	200	50	200	450
Petit Parser	100	300	200	600
Glamour	150	100	50	300
SUnit	100	-	50	150



# Activation predicate

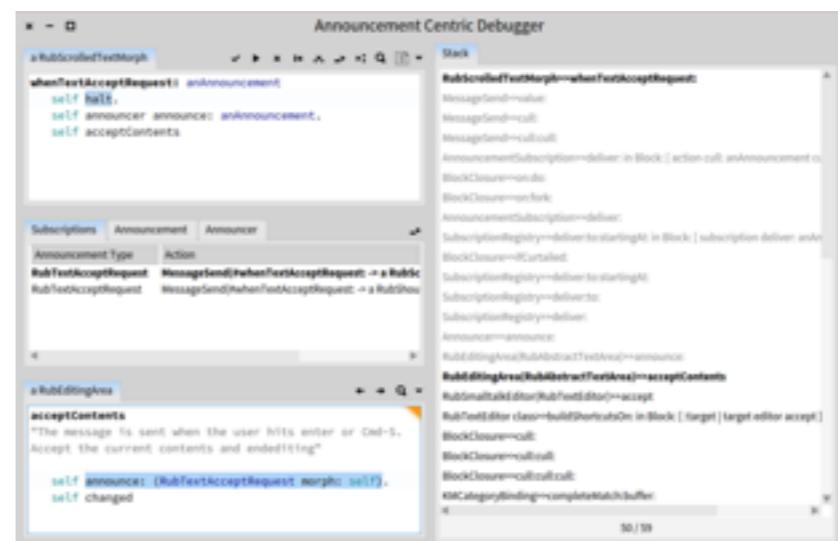
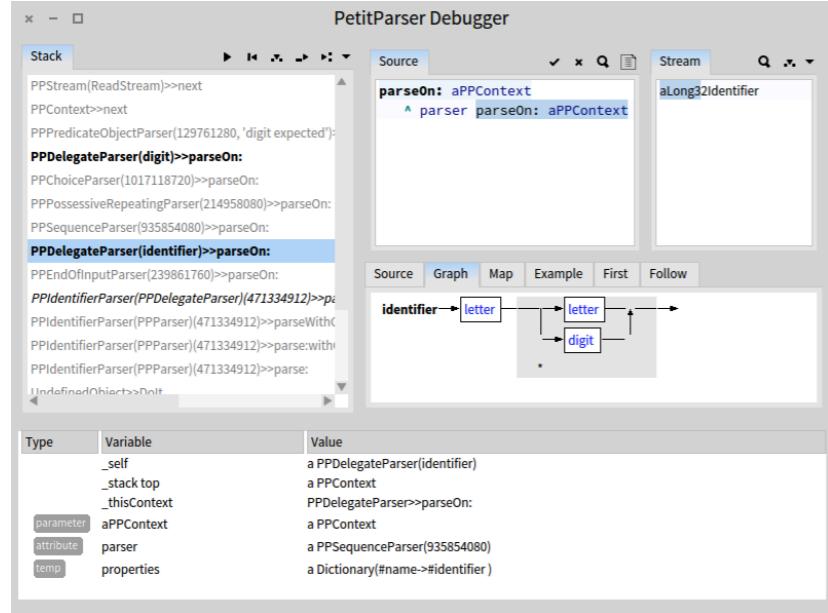


# Activation predicate

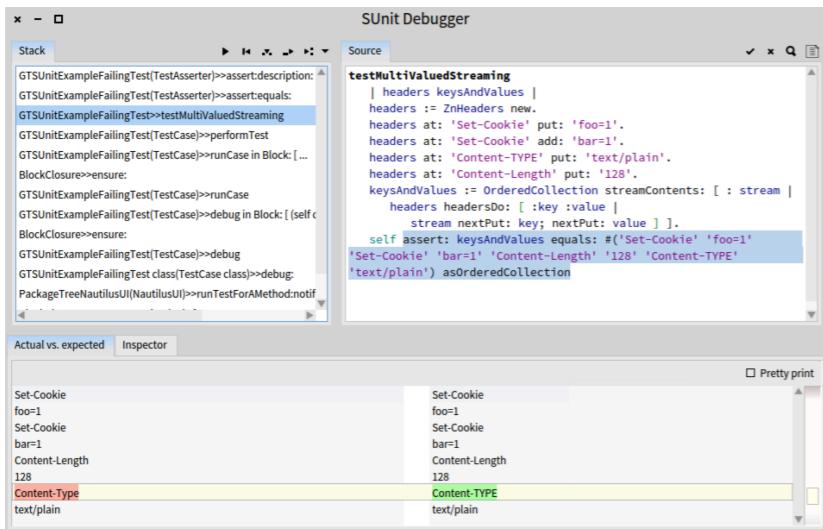


# Petit Parser

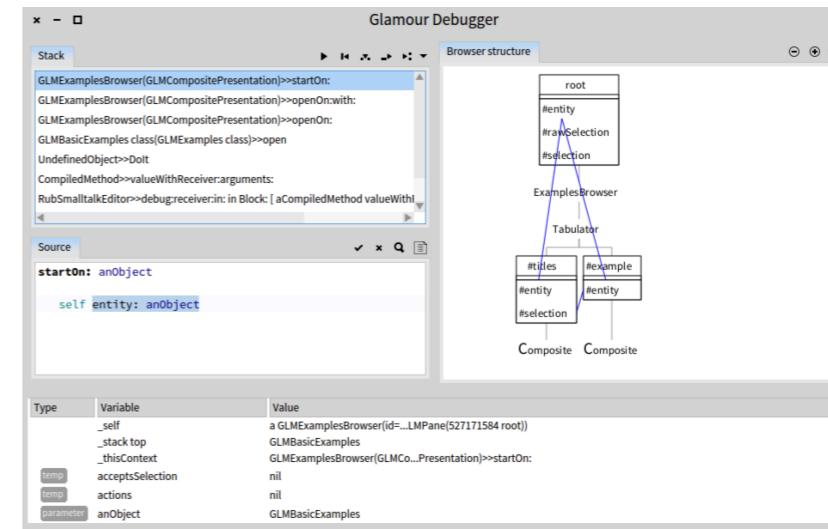
# Events



# SUnit

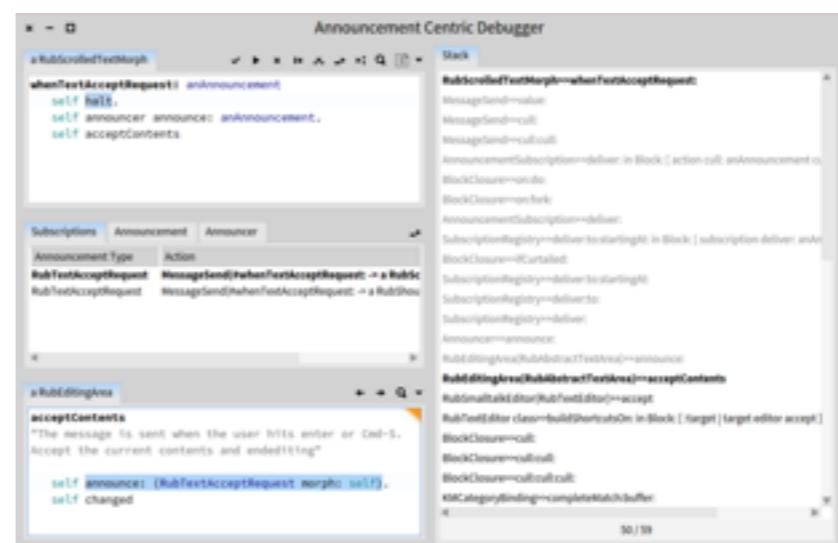
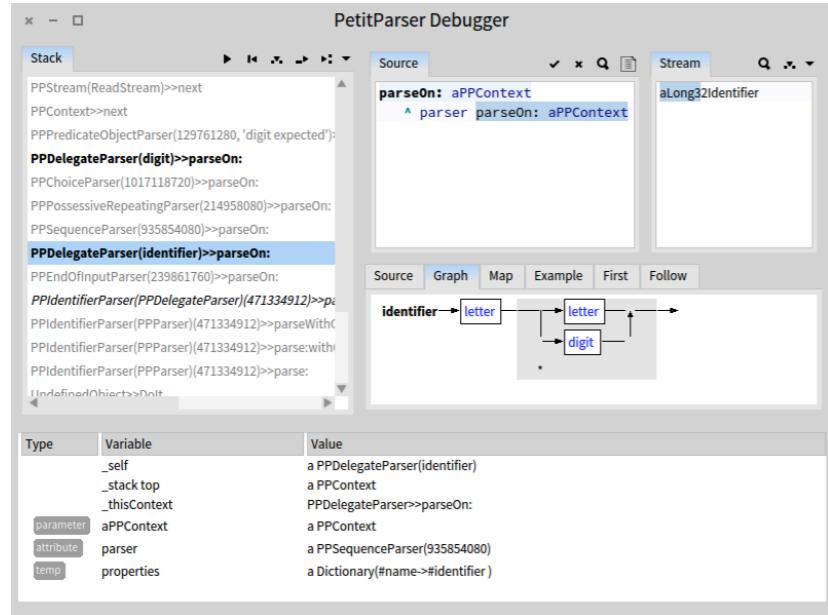


# Glamour

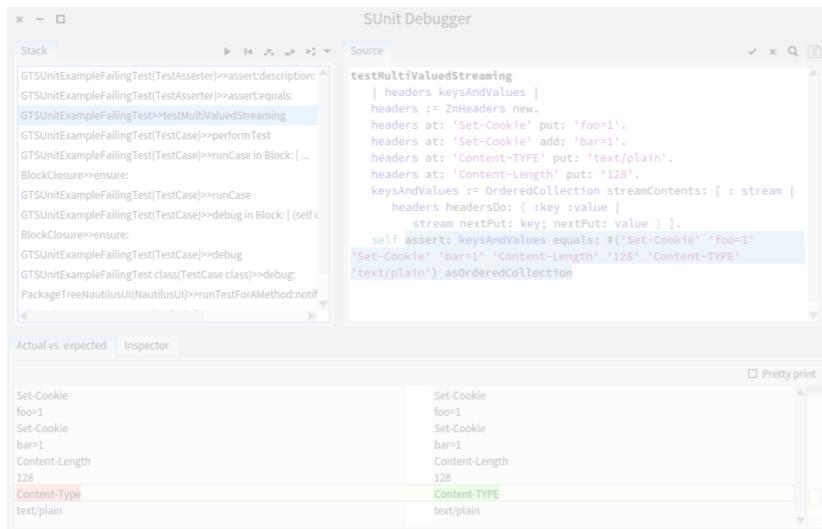


# Petit Parser

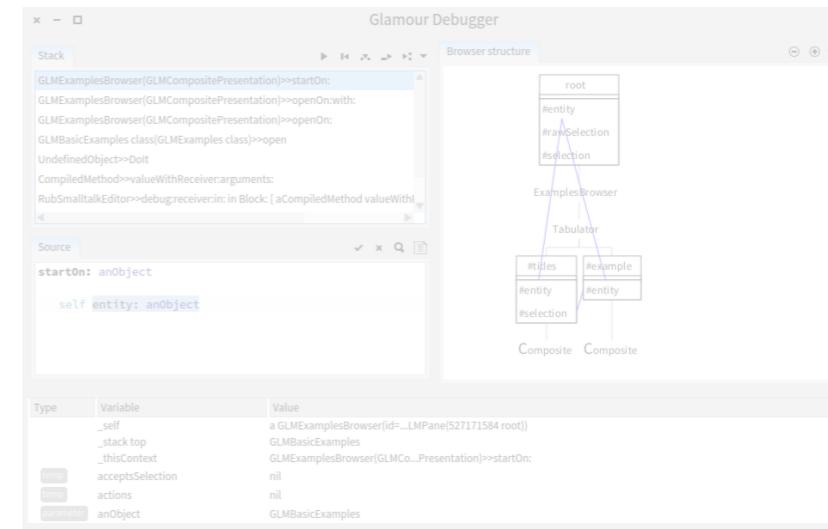
# Events



# SUnit

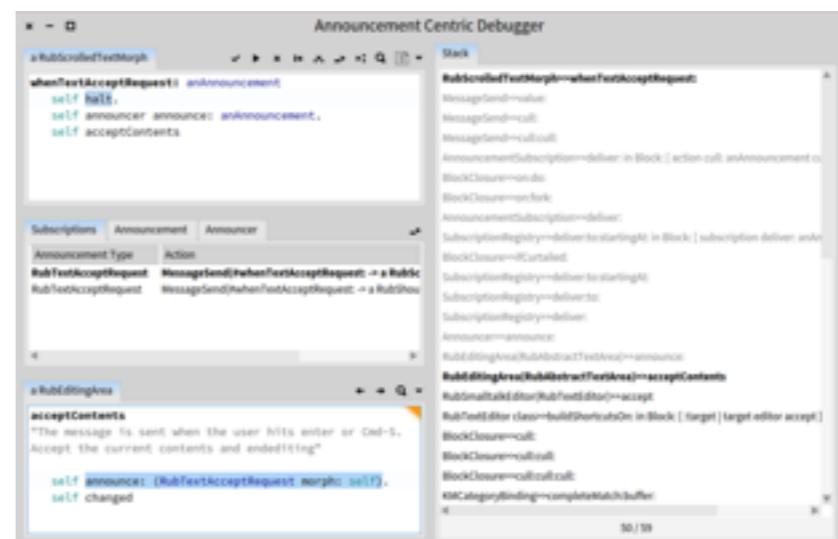
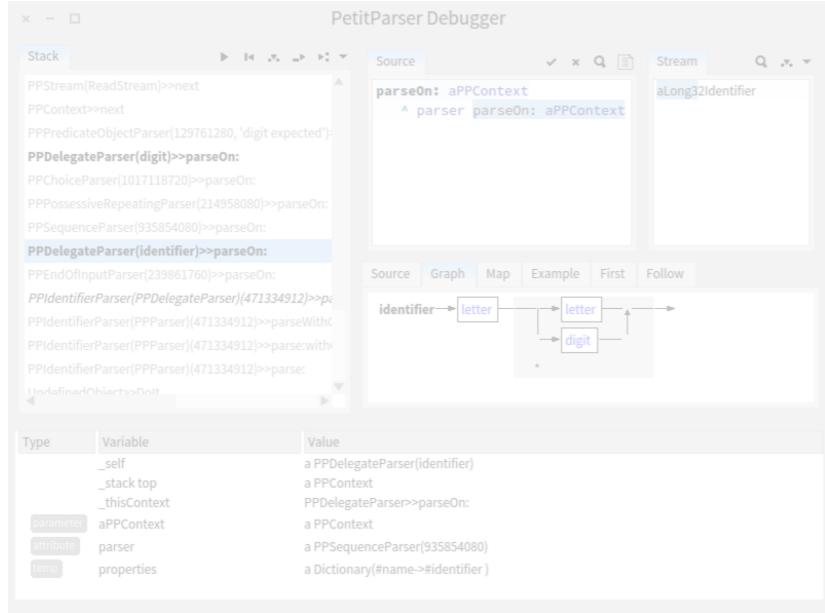


# Glamour

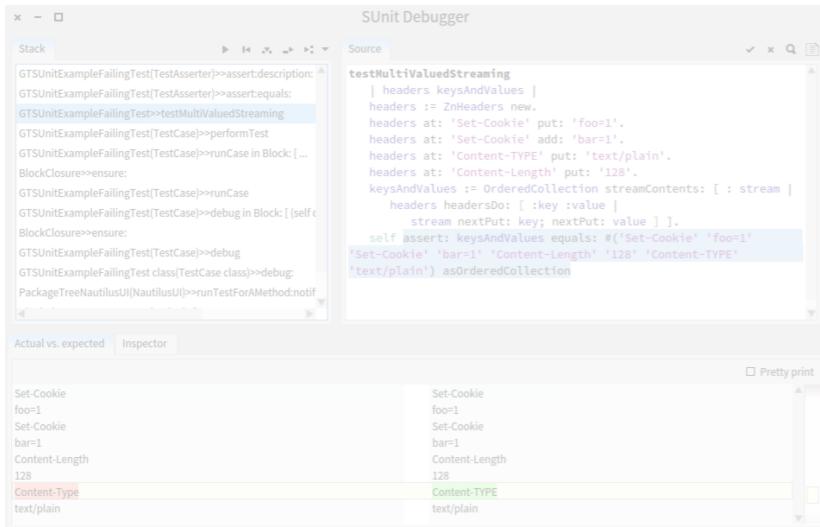


# Petit Parser

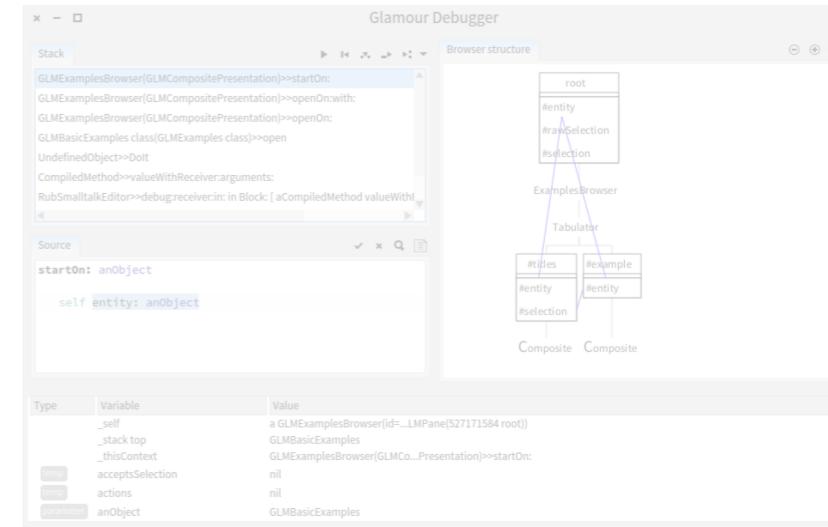
# Events



# SUnit



# Glamour

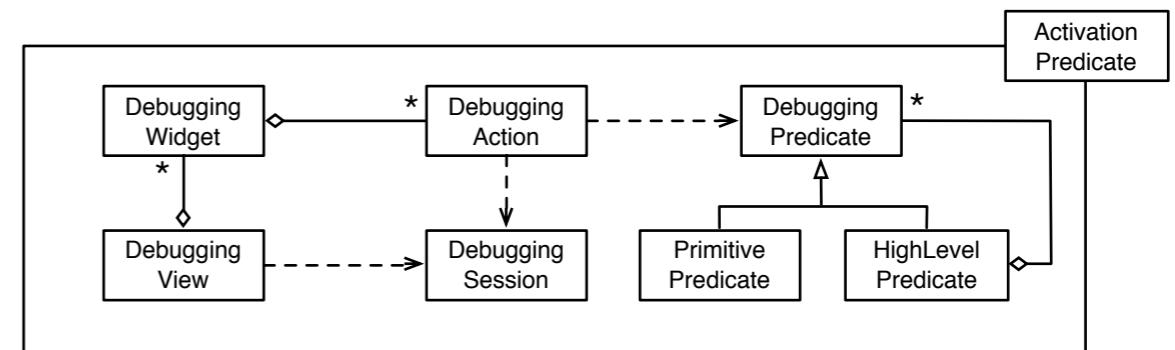


# Conclusions

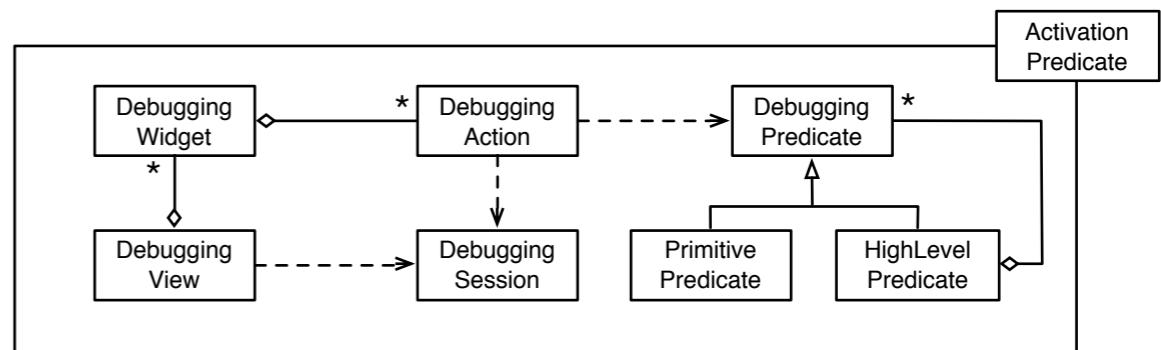
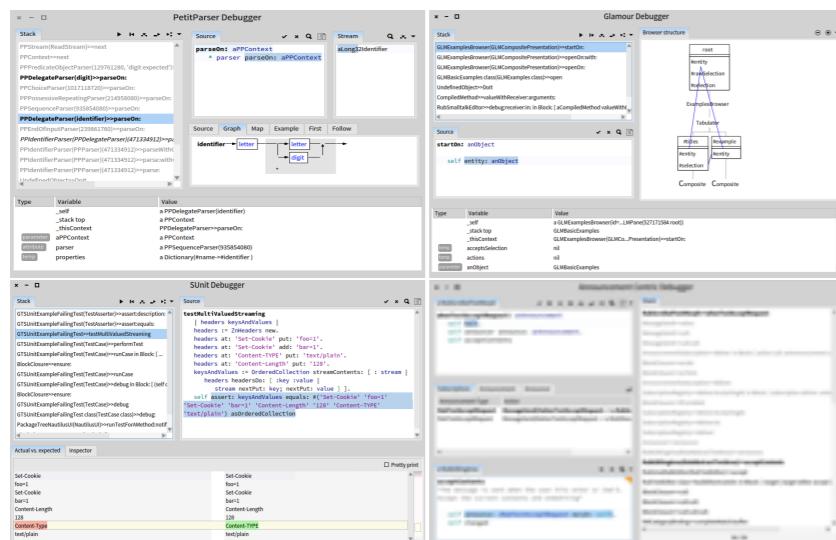
# Conclusions



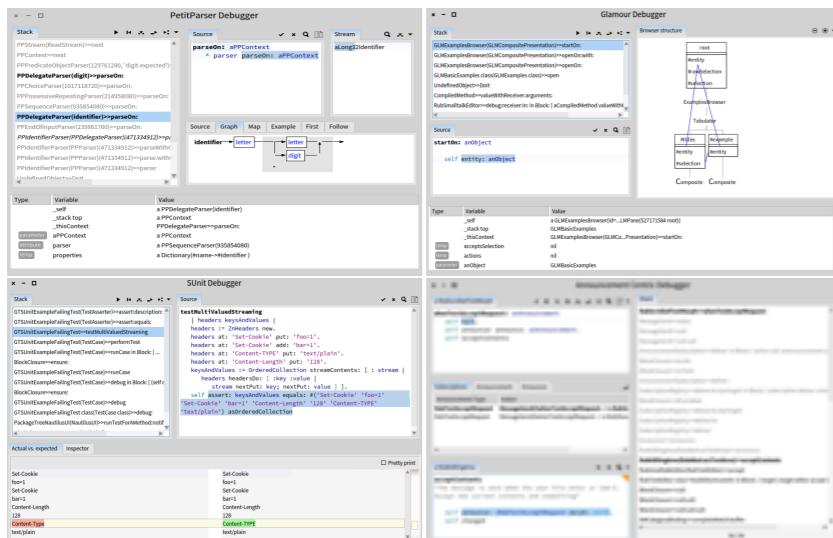
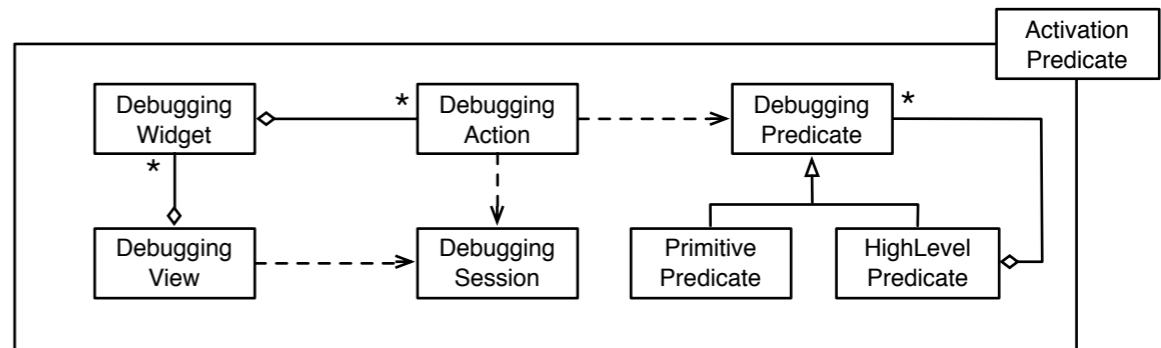
# Conclusions



# Conclusions



# Conclusions



	Session	Operations	View	Total
Base model	800	700	-	1500
Default Debugger	-	100	400	500
Announcements	200	50	200	450
Petit Parser	100	300	200	600
Glamour	150	100	50	300
SUnit	100	-	50	150

gt.moosetechnology.org

