A method to track the root cause of the Ammolite bug, using object-centric breakpoints

The description of the Ammolite bug can be found here: ../Description/Ammolite.md

Bug characteristics

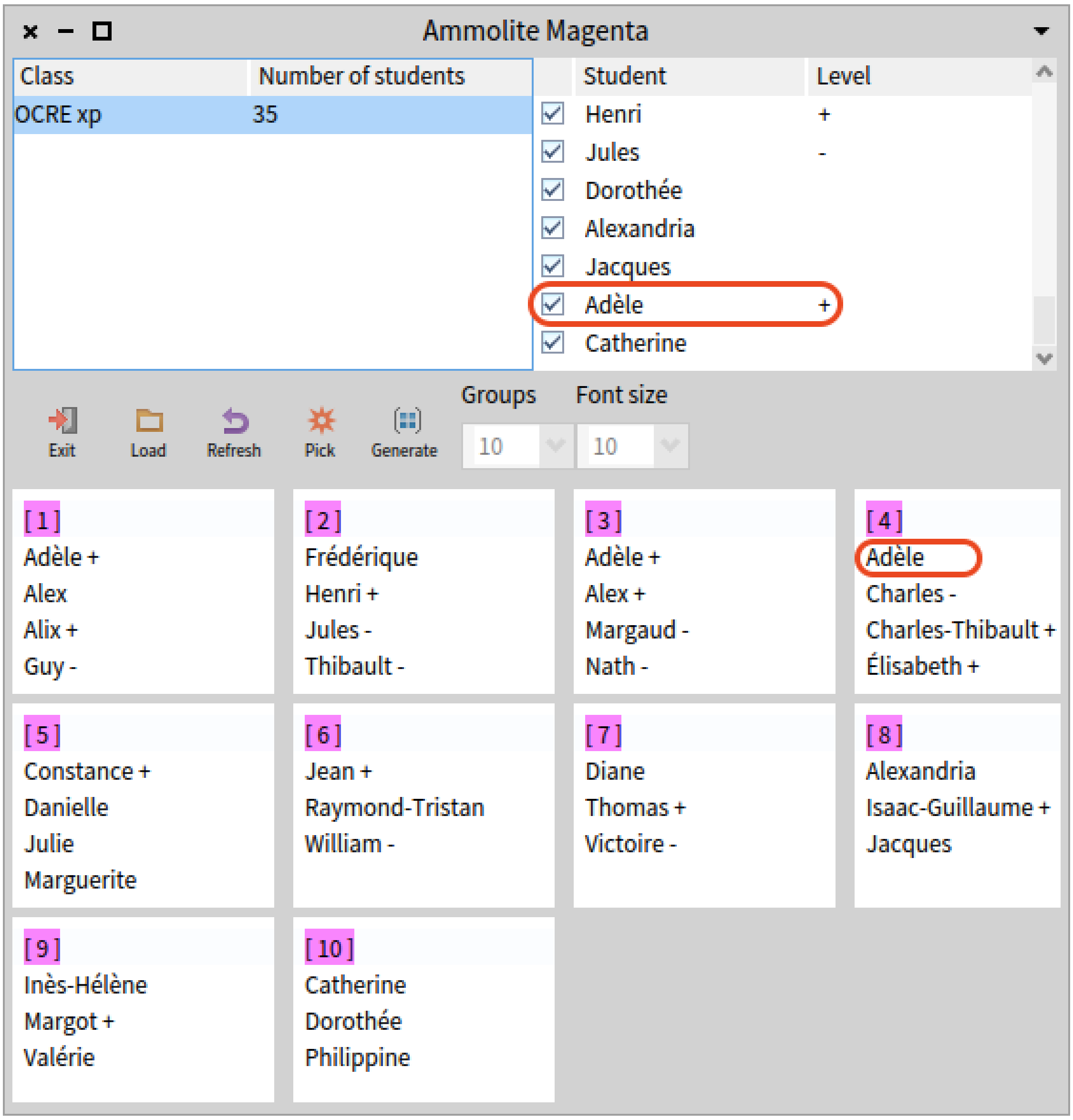
The bug has the following characteristics:

- It is a real bug of Ammolite which itself is a real application (used by teachers to create well-balanced groups of students),
- although not the same scenario, it conforms to Ressia'12 example `Live Object Interaction` where an object is incorrectly displayed on the application (here, with a missing marker).

Debugging method using object-centric debugging

Observing the bug

The bug can be observed when trying to generate groups of students using the `Generate` button.
One of the student, the last one named "Adèle" has no marker displayed in the group table while it is correctly displayed in the list:



Ammolite Magenta

Class	Number of students
OCRE xp	35

	Student	Level
☑	Henri	+
☑	Jules	-
☑	Dorothée	
☑	Alexandria	
☑	Jacques	
☑	Adèle	+
☑	Catherine	

Groups 10

Font size 10

[1]
Adèle +
Alex
Alix +
Guy -

[2]
Frédérique
Henri +
Jules -
Thibault -

[3]
Adèle +
Alex +
Margaud -
Nath -

[4]
Adèle
Charles -
Charles-Thibault +
Élisabeth +

[5]
Constance +
Danielle
Julie
Marguerite

[6]
Jean +
Raymond-Tristan
William -

[7]
Diane
Thomas +
Victoire -

[8]
Alexandria
Isaac-Guillaume +
Jacques

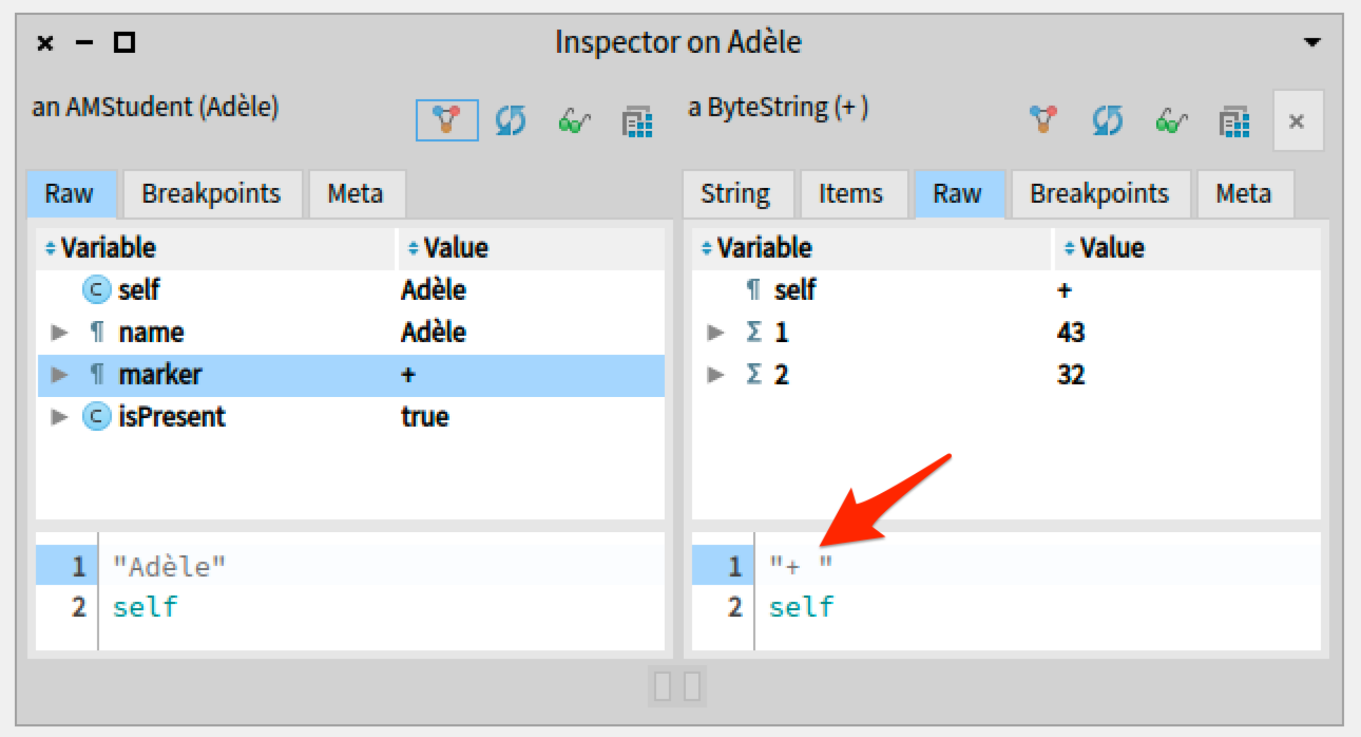
[9]
Inès-Hélène
Margot +
Valérie

[10]
Catherine
Dorothée
Philippine

Investigating with standard tools

We can inspect the object presenting the problem by right-clicking on the last student named "Adèle" in the list of students.

From there, one could already have an idea of the problem when looking at the "marker" attribute.



Variable	Value
self	Adèle
name	Adèle
marker	+
isPresent	true

```
"Adèle"
self
```

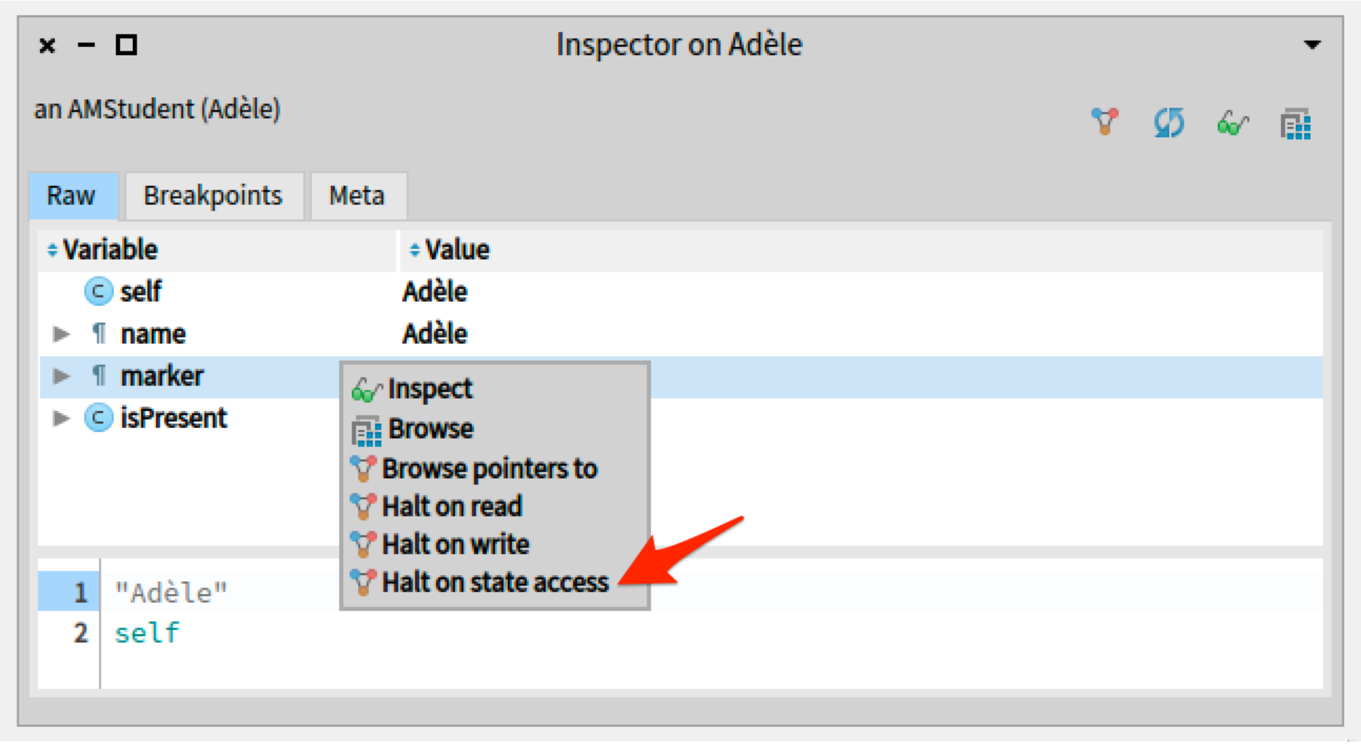
Variable	Value
self	+
1	43
2	32

```
"+ "
self
```

As pointed by the red arrow, there is a trailing space behind the "+" representing the marker.

Tracking the root cause with the object-centric debugger

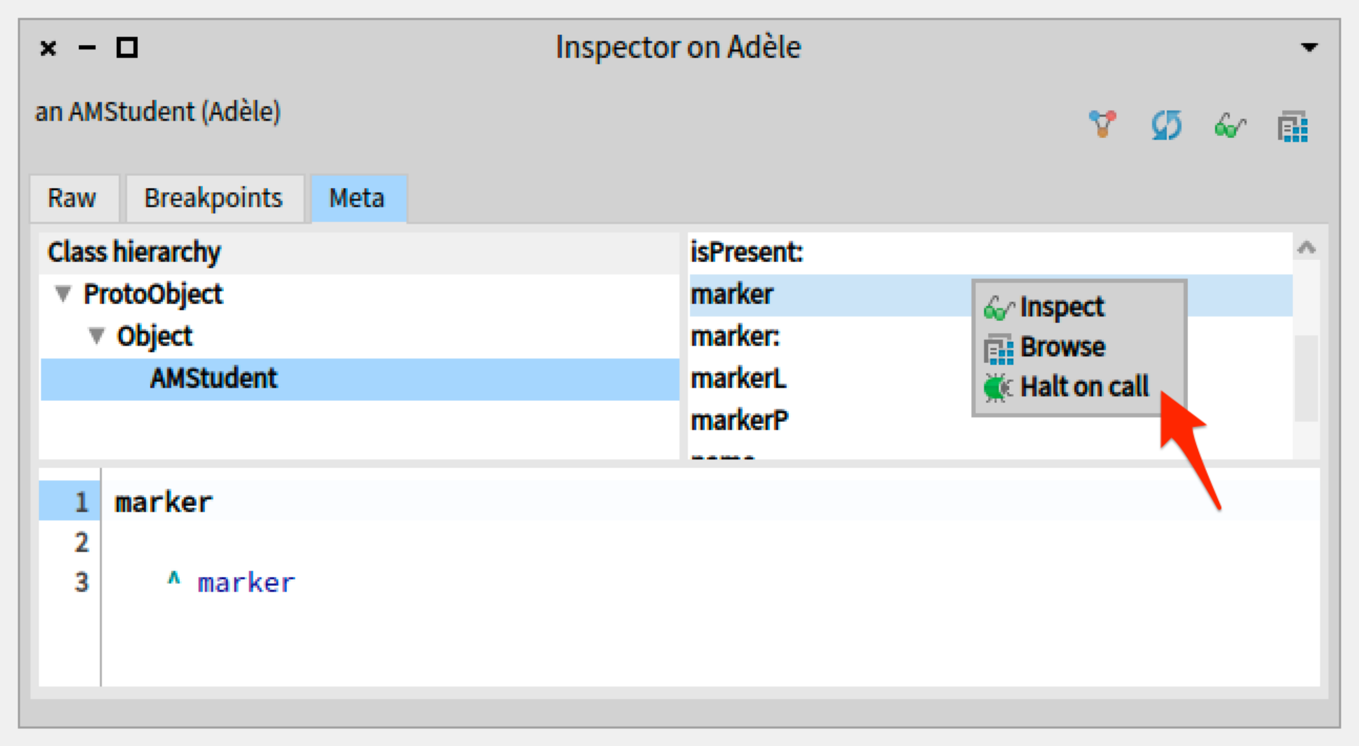
The following logical question would be "When/where is the marker accessed or modified in the code?". To answer it, one can place a `Halt on state access` breakpoint on the attribute "marker" of the problematic student.



Variable	Value
self	Adèle
name	Adèle
marker	
isPresent	

```
"Adèle"
self
```

Another possibility if the developer catches the trailing space detail would be to compare the marker with any other student's marker and see if the latter also have a trailing space.
If not, there is high chances for the trailing space to be the problem.
To confirm that, one would need to search for the access to the marker, and that can be done with a `Halt on call` placed on the accessor of the marker.



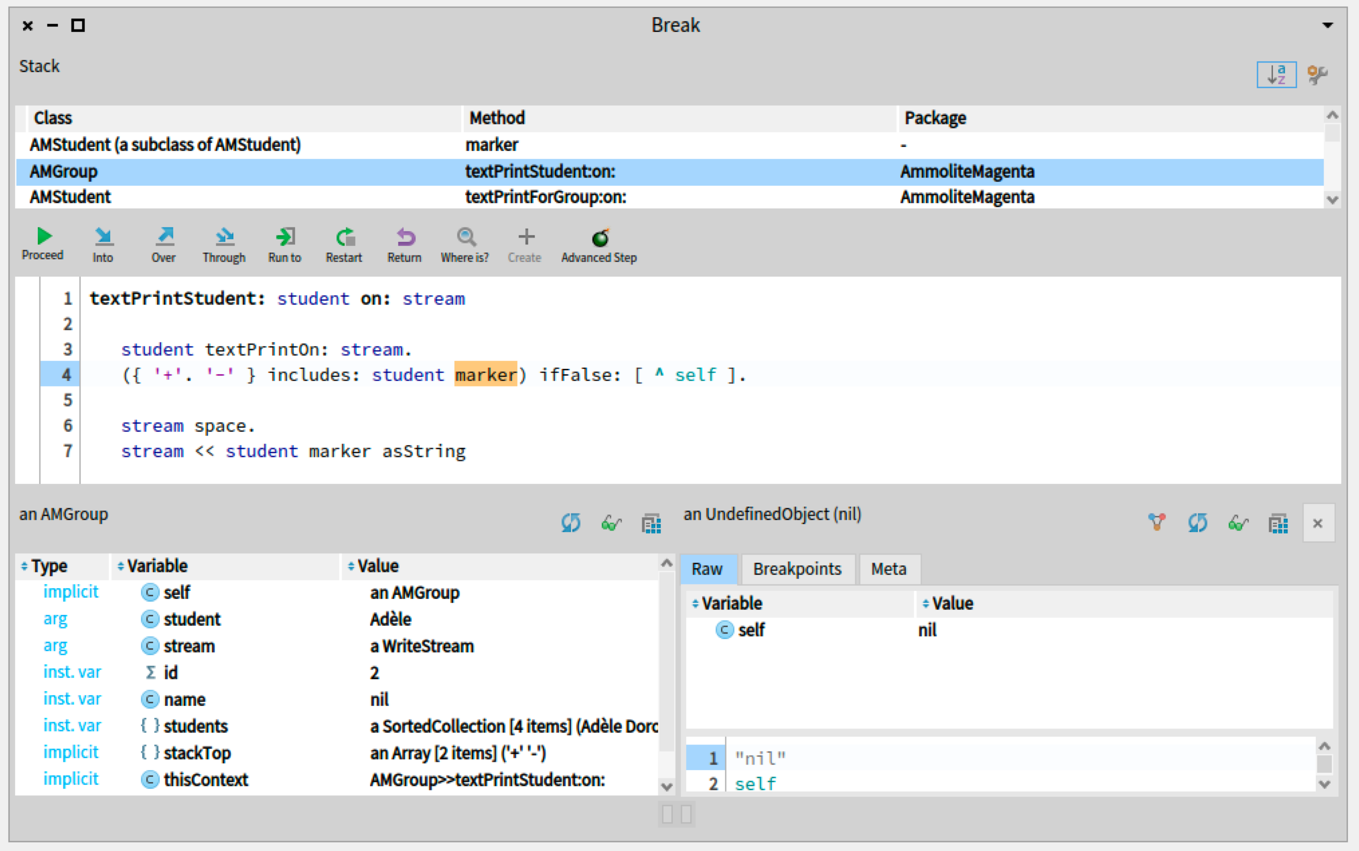
Class hierarchy
ProtoObject
Object
AMStudent

isPresent:
marker
marker:
markerL
markerP

```
marker

	^ marker
```

Ultimately with both these possibilities for debugging Ammolite, the developer should end up in the following method:



Class	Method	Package
AMStudent (a subclass of AMStudent)	marker	-
AMGroup	textPrintStudent:on:	AmmoliteMagenta
AMStudent	textPrintForGroup:on:	AmmoliteMagenta

```
textPrintStudent: student on: stream

	student textPrintOn: stream.
	({ '+'. '-' } includes: student marker) ifFalse: [ ^ self ].

	stream space.
	stream << student marker asString
```

Type	Variable	Value
implicit	self	an AMGroup
arg	student	Adèle
arg	stream	a WriteStream
inst. var	id	2
inst. var	name	nil
inst. var	students	a SortedCollection [4 items] (Adèle Dorc
implicit	stackTop	an Array [2 items] ('+' '-')
implicit	thisContext	AMGroup>>textPrintStudent:on:

Variable	Value
self	nil

```
"nil"
self
```

The method reads the marker to print it on the application interface only if it is recognized (either "+" or "-"), since the marker of our student contains a trailing space it is not recognized and the method execution ends here.

To fix this problem, one would need to correct the data from which the marker of our problematic student is derived or to trim the marker before checking its validity to handle this corner case.