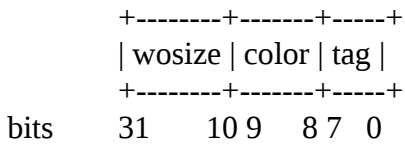


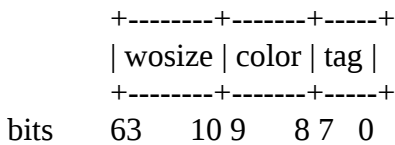
## Organisation et implantation des valeurs ZAM

### Structure de l'entete d'un champ

For 16-bit and 32-bit architectures:



For 64-bit architectures:



\*/

## Reperage d'un bloc d'un entier ##

1° bit == 1 ? long : bloc

Enumération des Tags : (blocT)

### Signification des Tags : < 251

250 : Forward\_tag :

249 : Infix\_tag :

248 : Object\_tag :

247 : Closure\_tag :

246 : Lazy\_tag :

### Signification des Tags : >= 251

251 : Abstract\_tag :

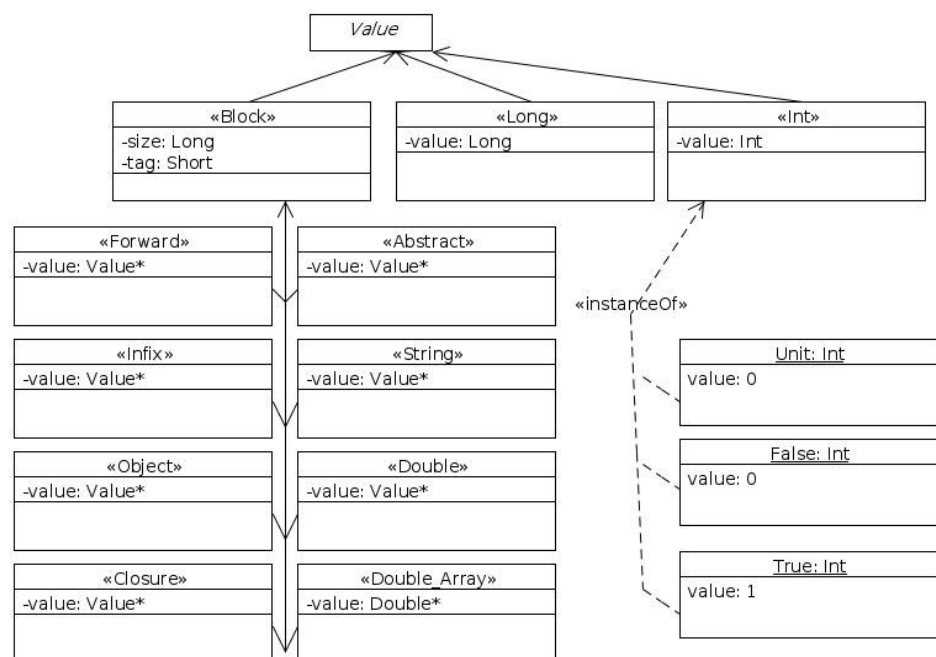
252 : String\_tag :

253 : Double\_tag :

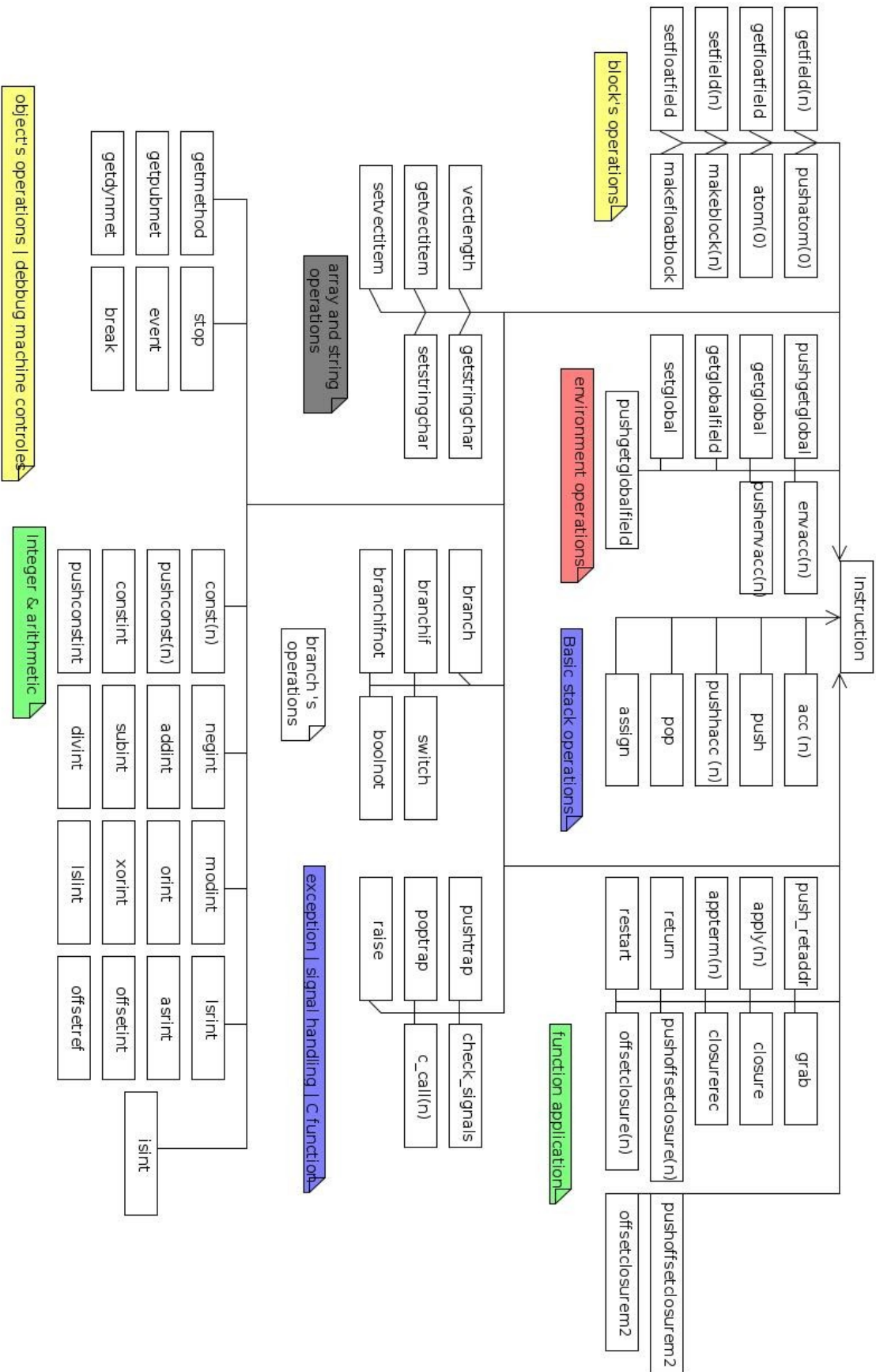
254 : Double\_array\_tag :

255 : Custom\_tag :

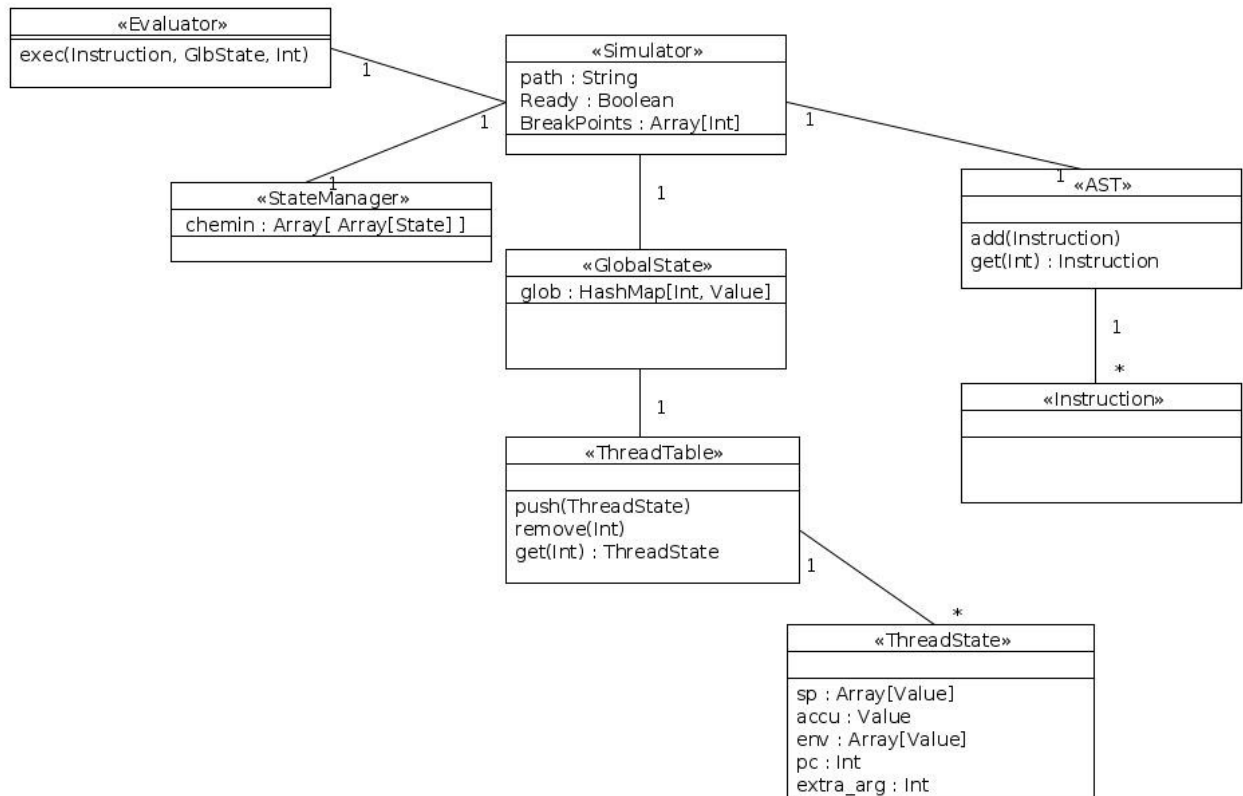
Implantation :



Les instructions ZAM

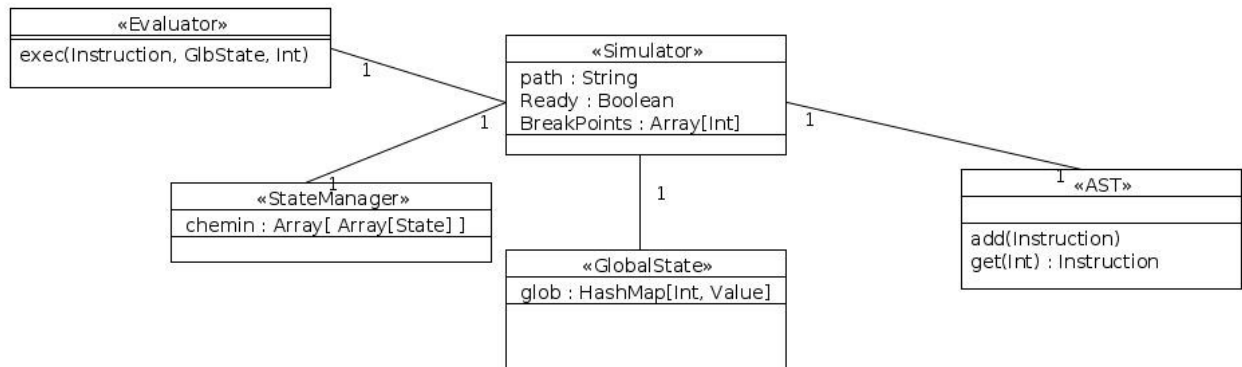


## General :



Voici la partie physique du simulateur, il reproduit ce que fait une ZAM et gère toutes les données autour.

## Simulateur :



Le simulator est le point d'entrée du programme.

Les sources correspondent à la version Ocaml et assembleur du programme.

Ready est à vrai lorsque que tout est initialiser pour pouvoir exécuter le programme.

Avancer fait avancer le thread t de n instructions.

Revenir(n) Restaure à la position n du thread concerné.

Log enregistre le thread faisant le pas courant, ainsi les liens des avancements entre thread peut être retrouvé.  $\text{Log}(t, n)$  = Le thread t fait n pas.

### Contexte :

**## On y trouve un état global et des états dynamiques**

**L'état global correspondant à l'état du programme (tas...)**

**Les états dynamiques sont les données d'un thread ##**

### GlobalState

Value \*glob ##environnement##

S'aidant de Scala, l'environnement global est implanté par une Hasmap reliant l'itération (utilisée dans les instructions) à la valeur.

ThreadTable

C'est la table des threads existant dans le code. Cette classe permet d'accéder et de gérer les threads.

### ThreadState

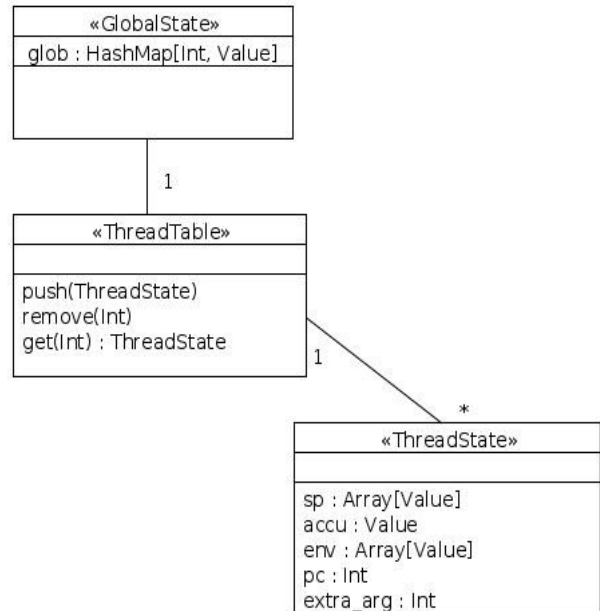
Entier pc ##Position courante dans l'AST##

value \* sp ##pile, pointeur de valeur##

value accu ##accumulateur : valeur##

value\* env ##environnement##

extra\_arg int ##extra arg##



*Chaque Thread possède son tas son accumulateur et son pointeur de code. Elles sont indépendantes, seul l'environnement global les relie.*