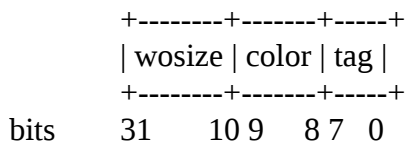


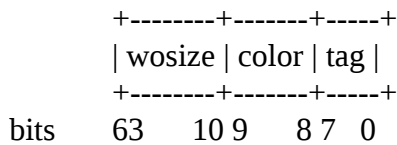
Organisation et implantation des valeurs ZAM

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 :

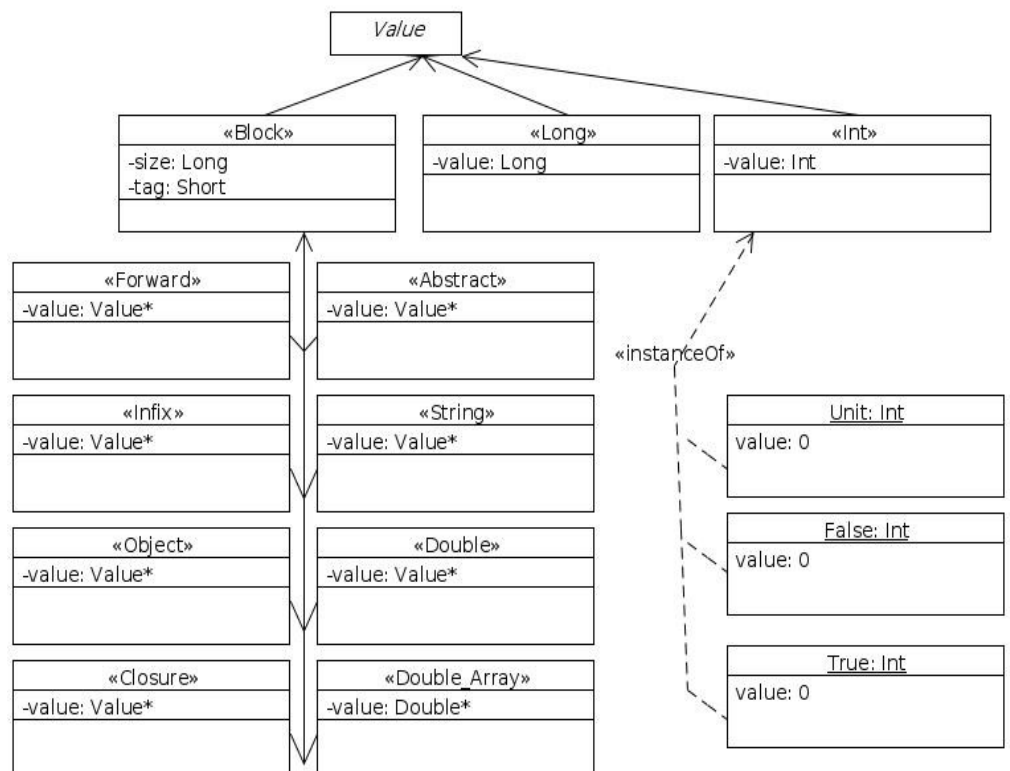
252 : String_tag :

253 : Double_tag :

254 : Double_array_tag :

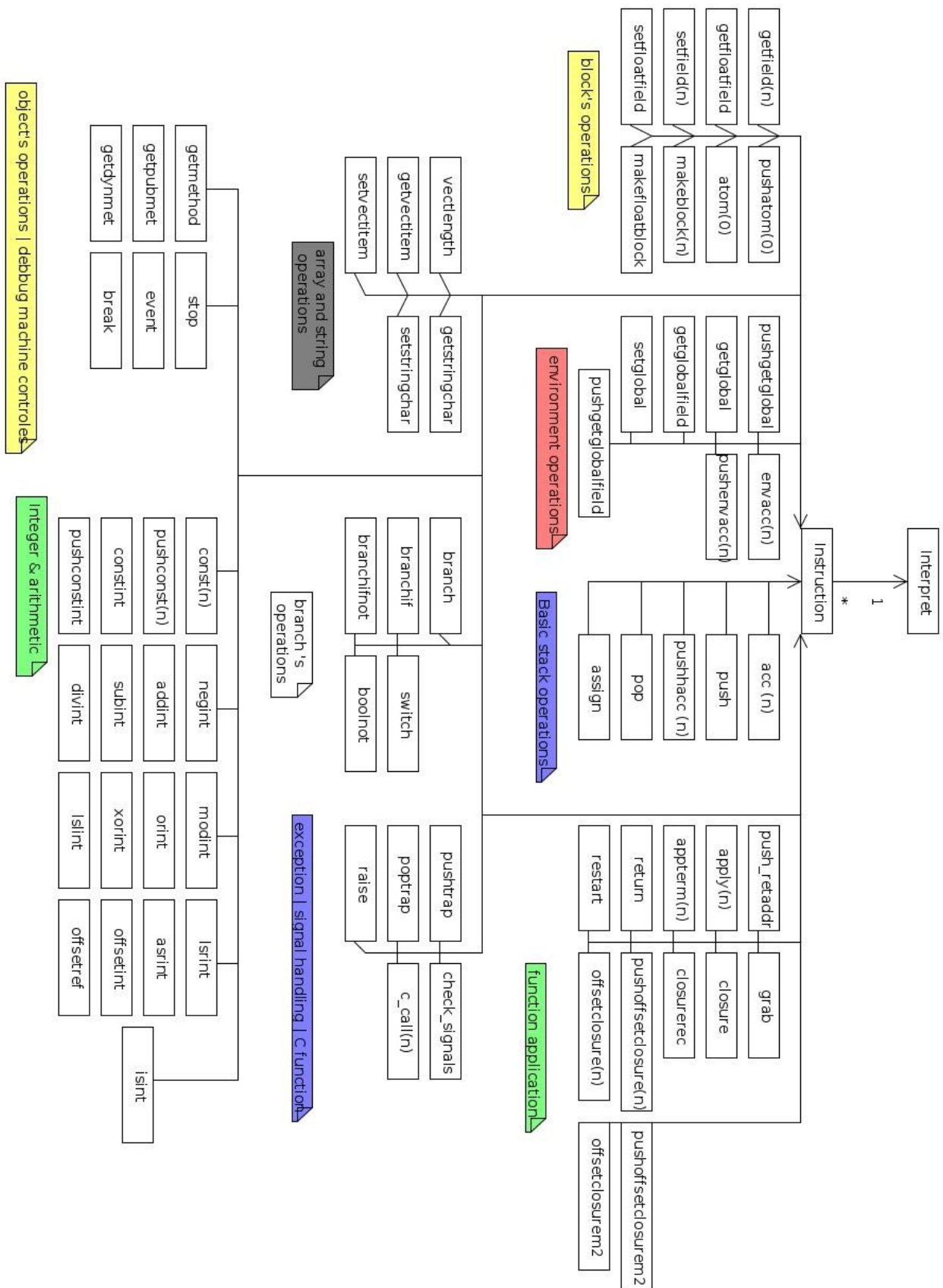
255 : Custom_tag :

Implantation :



Interpreteur d'instruction ZAM

De la même manière une énumération liera une instruction avec un entier. (instrT)



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 ##

Etat Global

Value *glob ##environnement##

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

int extra_args

Etat thread

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

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

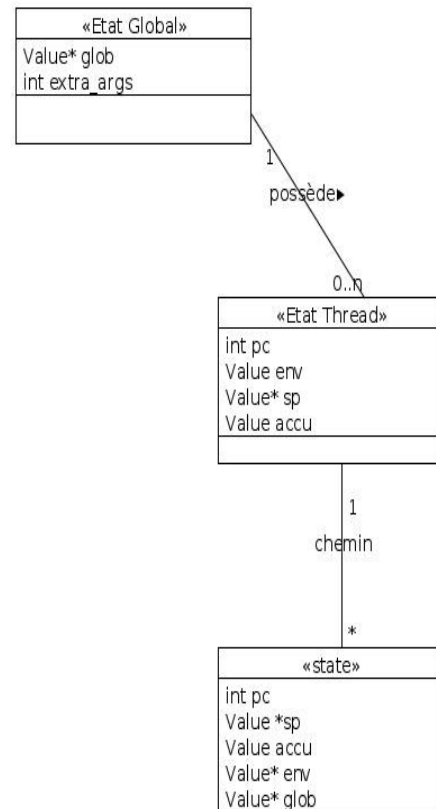
value accu ##accumulateur : valeur##

value* env ##environnement##

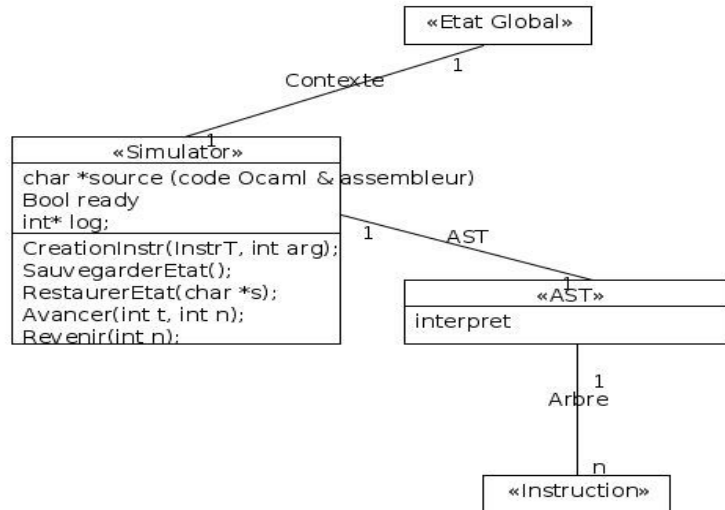
Chaque Thread possède son tas son accumulateur et son pointeur de code. Elles sont indépendante, seul l'environnement global les relie.

L'historique de la thread (chemin)

Un état, dit state est l'ensemble des positions de la thread et les anciennes valeurs du contexte général. C'est une image de l'étape précédente.



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.