

INTRODUCTION	2
SPÉCIFICITÉ FONCTIONNEL	2
PRINCIPES DU CODAGE DE HUFFMAN	2
AVANTAGES DU CODAGE DE HUFFMAN	2
INCONVÉNIENTS DU CODAGE DE HUFFMAN	3
SPÉCIFICITÉS TECHNIQUES	3
ANALYSE	3
DIAGRAMME DE CAS UTILISATION	3
DIAGRAMME DE SÉQUENCE	4
DIAGRAMMES ACTIVITÉS	7
CONCEPTION	7
RÉSULTAT & Difficultés rencontrées	8
TEST	8
CONCLUSION	8

INTRODUCTION

Le stockage ou la transmission de données volumineuses pose problème soit vu l'espace mémoire nécessaire pour le stockage soit vu le débit et le temps demandés pour la transmission, La compression est la solution pour ce problème. Elle consiste à transformer les données vers une forme moins gourmande en espace de ce fait nous avons plusieurs méthodes de compression d' informations (compression statique , de type dictionnaire, dynamique, etc) dès lors nous allons étudier une méthode de compression Le codage de Huffman permet de représenter les données avec le moins de bits possible.

ETUDE DE BESOINS ET SPÉCIFICITÉS

L'analyse des besoins vise à identifier les exigences du projet et à faire le point sur les éléments attendus.

SPÉCIFICITÉ FONCTIONNEL

PRINCIPES DU CODAGE DE HUFFMAN

le codage de huffman permet de réduire la taille d' un fichier sans le modifier : elles code les symboles ayant une plus grande probabilité en utilisant un principe basé sur la construction d un arbre binaire.

AVANTAGES DU CODAGE DE HUFFMAN

le principal avantage de la compression en utilisant ce codage est absence de perte de données c'est à dire que le fichier compressé est identique au fichier initiale

INCONVÉNIENTS DU CODAGE DE HUFFMAN

La taille de l'arbre binaire généré qui s'ajoute au code compressé influe sur la taille du fichier compressé et peut réellement avoir un impact considérable lorsqu'il s'agit d'un fichier de petite taille. On note également l'analyse statique du message qui peut prendre du temps et qui cause une légère diminution de la vitesse de compression du fichier.

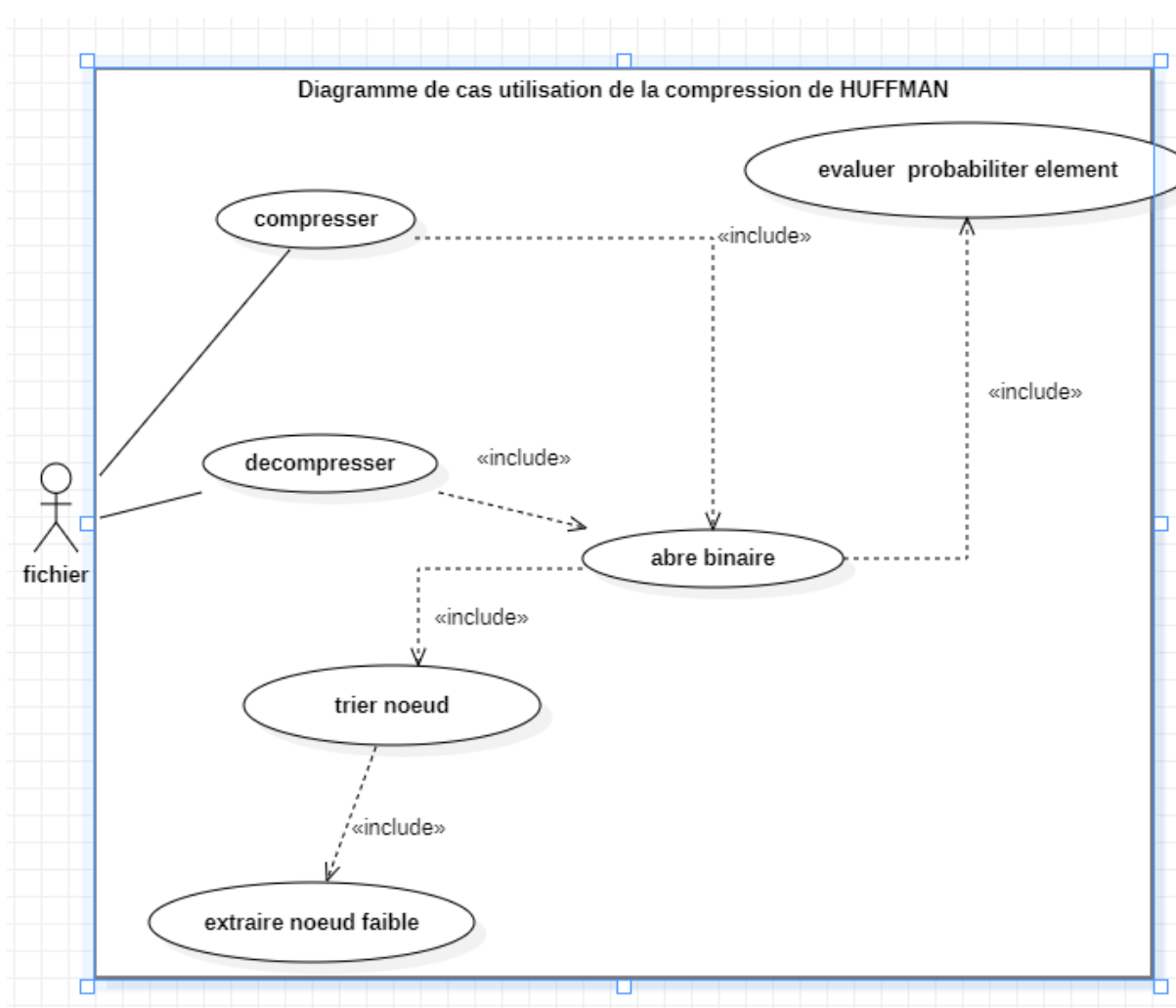
.

SPÉCIFICITÉS TECHNIQUES

ANALYSE

La compréhension de la méthode de compression de HUFFMAN se fera en utilisant une modélisation UML à travers ses différents diagrammes.

- DIAGRAMME DE CAS UTILISATION

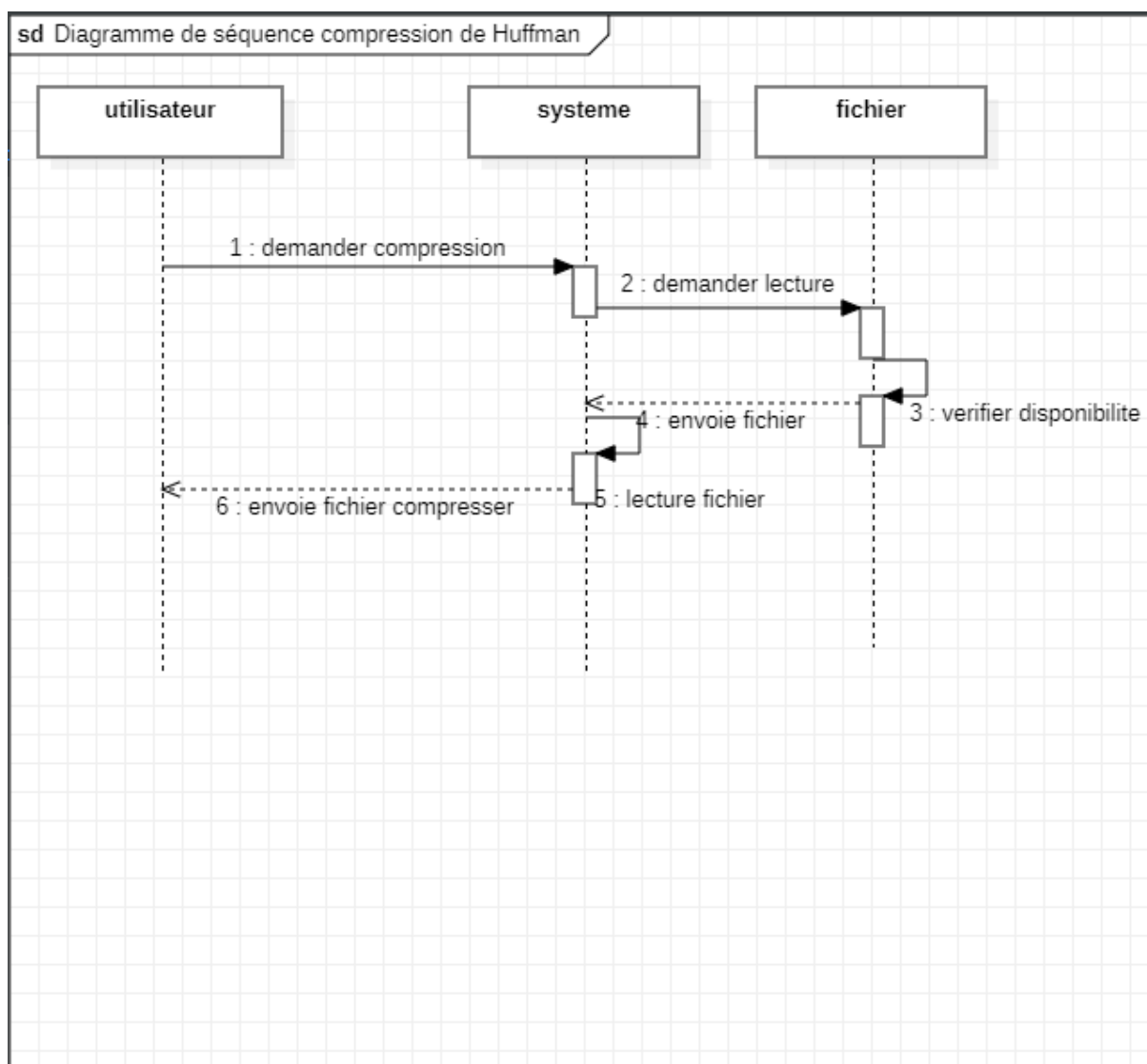


Lorsqu'un utilisateur effectue une opération de compression, il doit automatiquement ouvrir le fichier .

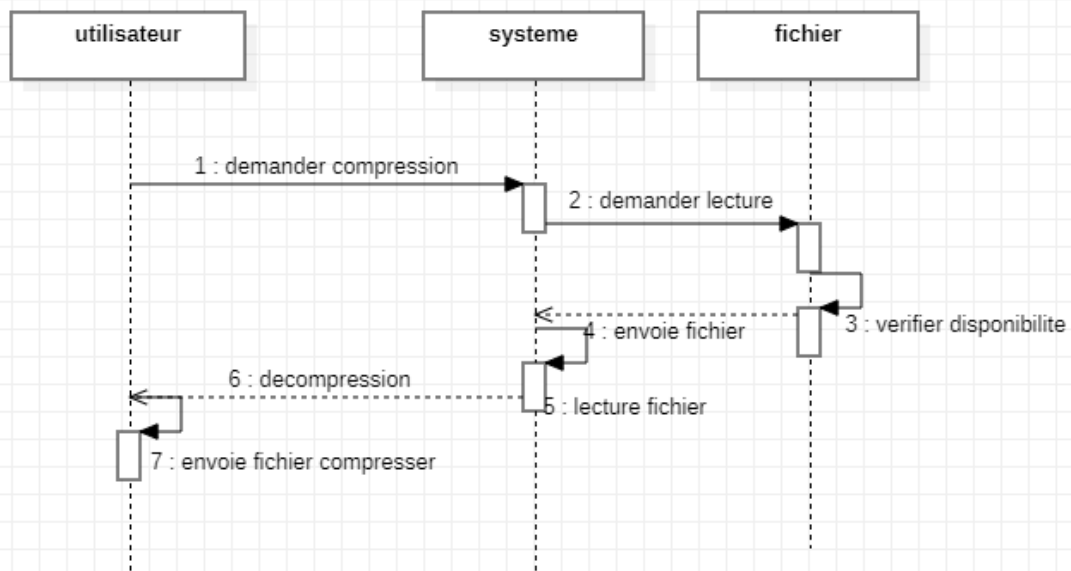
le programme doit automatiquement construire arbre binaire associé au fichier dès lors la réalisation de la construction de cet arbre nécessite des opérations de recensement des occurrences des différents caractères et de trie.

De même la décompression s'effectue en utilisant l'arbre binaire généré lors d'une opération de compression

DIAGRAMME DE SÉQUENCE

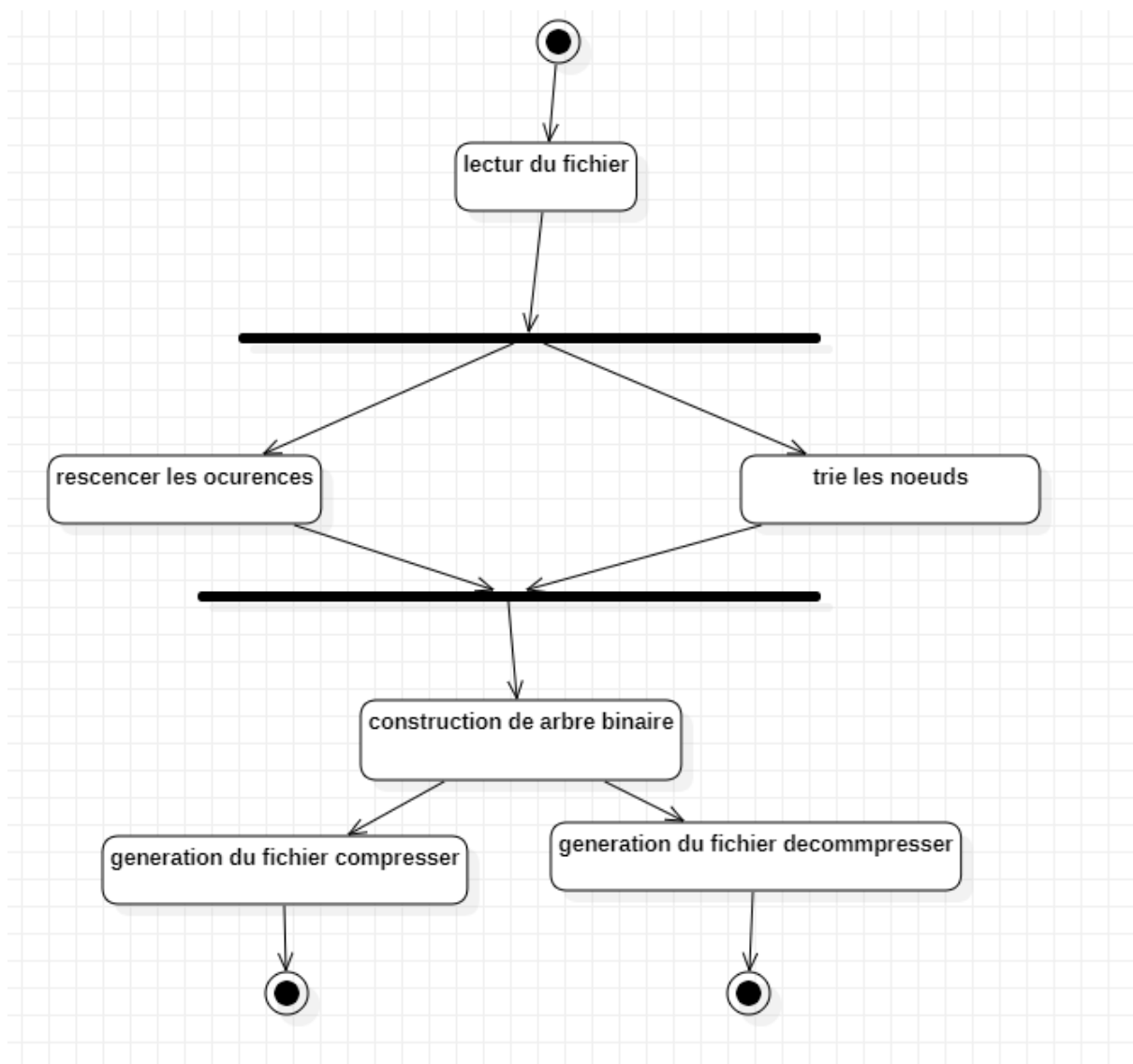


Après la lecture d' un fichier on doit pouvoir l'ouvrir ce qui passe par une lecture au préalable.
puis on génère un fichier compresser .



La décompression s'effectue en utilisant le fichier compresser .

- DIAGRAMMES ACTIVITÉS



Pour compresser un fichier le programme doit lire le fichier et extraire simultanément les caractères et stocker dans un tableau , puis ce tableau est trié en associant a chaque caractère sont occurrence
Ceci constitue les premier nœuds qui nous serviront à construire l'arbre binaire final.
une fois arbre établie on peut lire les différentes valeurs binaires associé à chaque caractères et le stocker dans un fichier qui représente le fichier compresser .
la décompression s' effectue en utilisant l'arbre binaire généré pendant la compression de ce fichier

CONCEPTION

RESULTAT

TEST

CONCLUSION

le codage de HUFFMAN est un algorithme de compression sans perte de données , il est simple et optimal.