

Modifications effectuées après l'écriture du rapport

1. Corrections apportées au prototype animal-végétation

1.1. Paramètres de la classe *Ingestion* non initialisés

Lors du lancement de simulations, j'obtenais souvent un arrêt de l'exécution sans aucun message d'erreur. L'erreur provenait, dans le module végétal, de l'écriture de la hauteur de l'herbe et de la quantité de matière disponible des quinze cellules du champ de vision de l'animal dans le fichier de choix *choix.csv*. Si l'animal se trouvait en bord de parcelle, certaines de ces cellules pouvaient se trouver en dehors de la parcelle. Comme les valeurs de ces cellules n'étaient pas initialisées, elles contenaient le contenu de la mémoire, c'est-à-dire une valeur qui n'était pas correcte. Il suffisait donc d'initialiser la biomasse et la quantité de matière à 0 dans le constructeur de la classe *Ingestion*, cette classe permettant de récupérer les caractéristiques d'une cellule. Ainsi pour les cellules à l'intérieur de la parcelle les valeurs sont calculées et mises à jour à chaque opération, mais restent à 0 pour les cellules en dehors de la parcelle. Cette solution évite donc d'avoir des variables non initialisées et a résolu le problème d'arrêt de l'application.

1.2. Modification des équations de calcul de vitesse d'ingestion

Ces différentes modifications ont été effectuées suite au travail de validation de Magali Jouven. Dans un premier temps, dans les classes *Ingestion* et *Preference*, la valeur de MLV a été fixée à $0.05 * \text{biomasse_VV}$ au lieu d'une valeur modulée selon le degré d'épaisseur. Ensuite, le seuil pour la transition entre les équations de calcul de la vitesse d'ingestion potentielle a été modifié à 0.25 au lieu de 0.1. Enfin, la vitesse d'ingestion potentielle a été corrigée par un facteur dépendant du poids de l'animal : $0.4596 * (\text{pow}(\text{Poids}, 0.71) - \text{pow}(69, 0.71))$.

2. Modifications apportées au module spatial et social

2.1. Arrêt de l'activité courante d'un animal

Un animal peut voir son activité interrompue dans deux cas :

- Lorsqu'il doit se rapprocher ou s'éloigner du troupeau. Lorsque son déplacement est terminé, il reprend son activité s'il était en train de manger ou sinon il effectue un nouveau choix d'activité (cf. Figure 1) ;
- Lorsqu'un animal le prévient qu'il est devenu leader. Il choisit alors une nouvelle activité quand il a fini de se rapprocher du leader et que le leader a fini son déplacement long (cf. Figure 2).

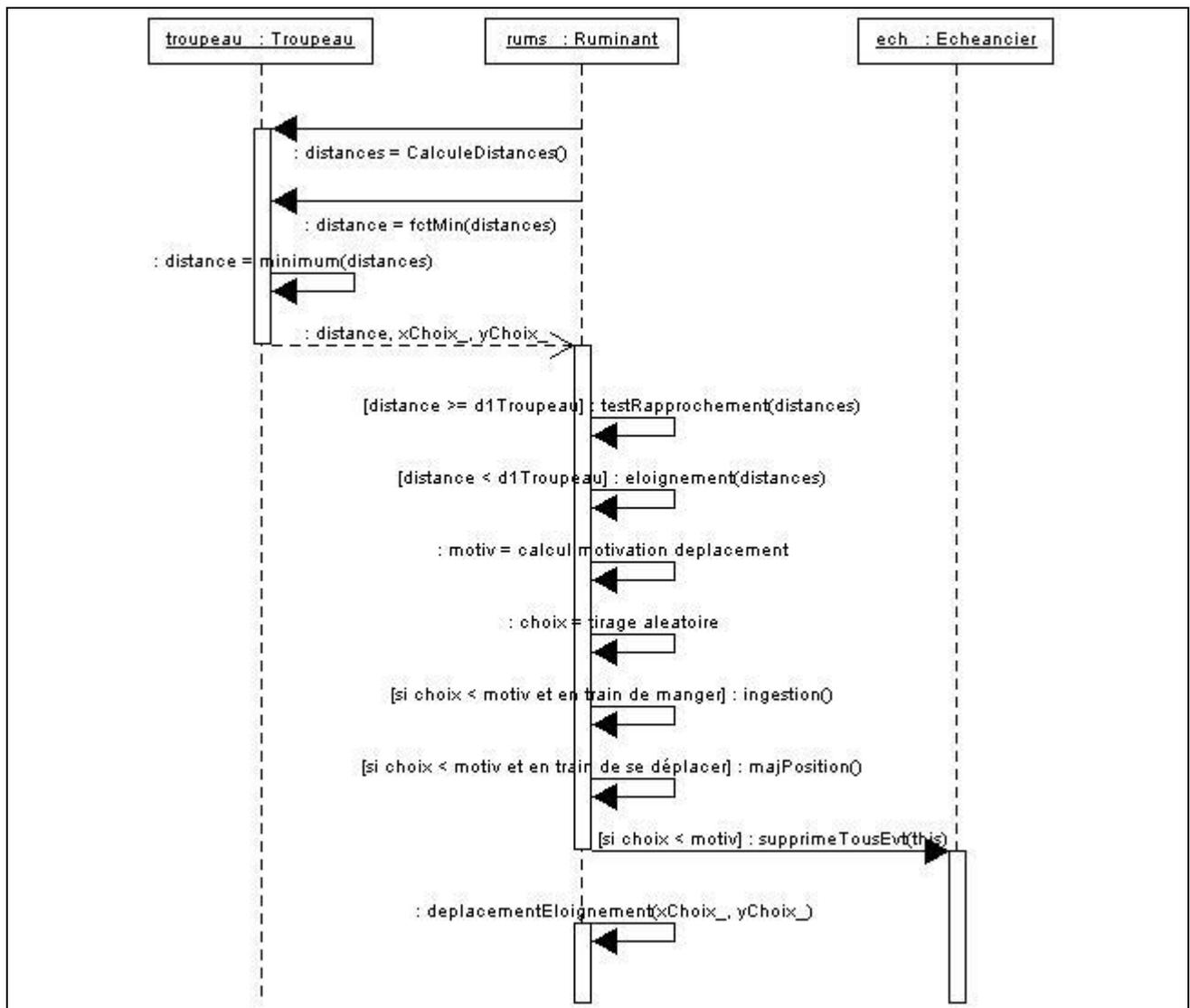


Figure 1 : Diagramme de séquences de la cohésion sociale

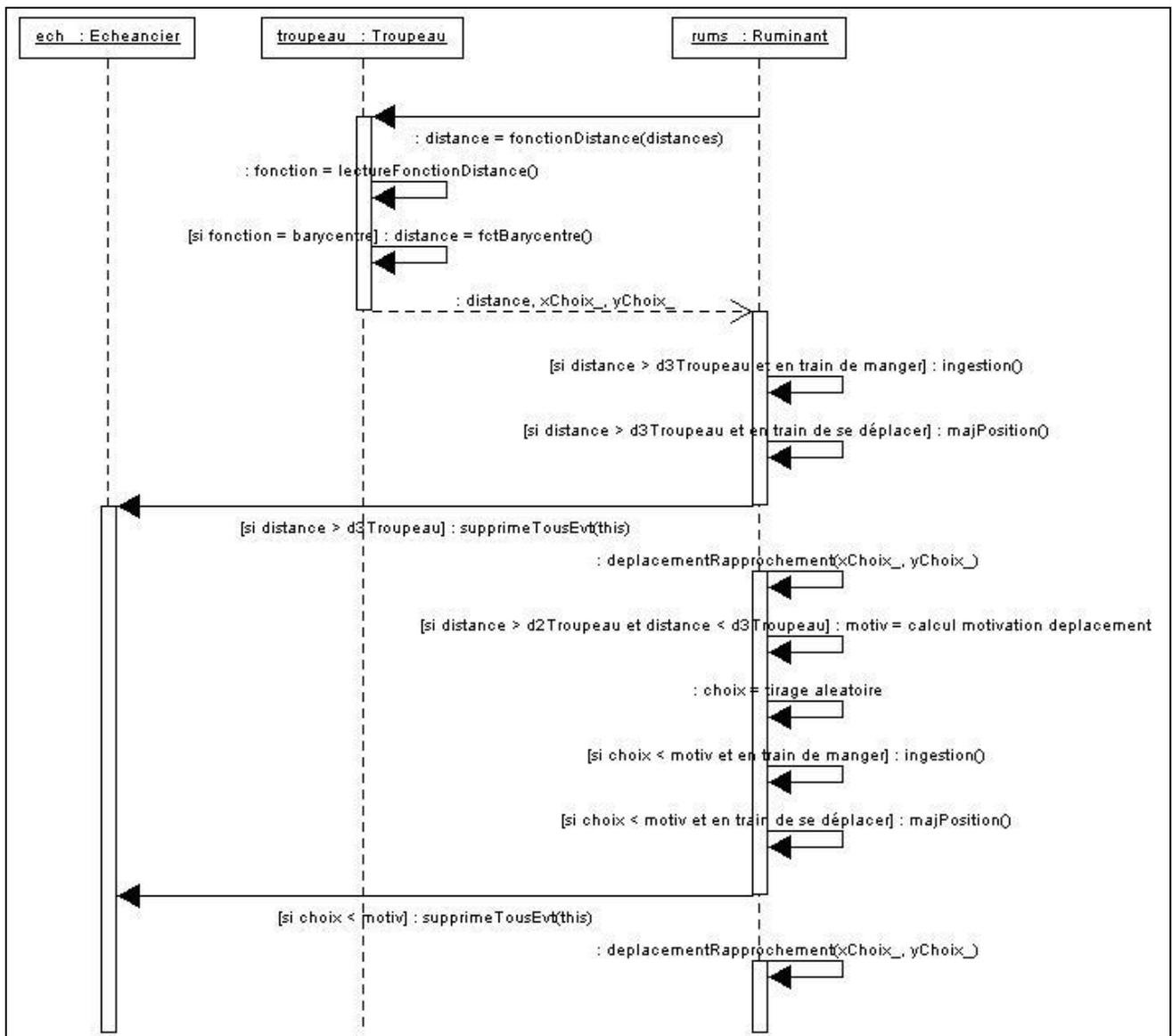


Figure 2 : Diagramme de séquences du rapprochement par rapport au leader

Si l'animal était en train de se déplacer, il faut mettre à jour sa position. Par contre, s'il était en train de manger, il faut qu'il ne puisse manger que la quantité de matière qu'il a réellement eu le temps de défolier. Pour cela, on a décomposé la méthode *ingereUneCellule* de la classe *Ingestion* en deux parties :

- Une première nommée *calculTempsCellule* qui calcule la vitesse d'ingestion que l'animal pourra avoir sur la cellule et le temps qu'il pourra y rester ;
- Une deuxième nommée *ingereUneCellule* qui permet à l'animal d'ingérer ce qu'il a eu le temps de défolier.

Un événement, l'événement *FinActivite* est inséré pour permettre à l'animal d'ingérer la quantité défoliée sur la cellule courante. Cet événement appelle la méthode *finActivite* qui a aussi pour but de mettre à jour les paramètres de rumen à la fin de la séquence.

La méthode *forceFinActivite* a un fonctionnement assez proche de la méthode *finActivite* mais elle est appelée lorsqu'un animal prévient le troupeau qu'il est devenu leader. Elle met donc à jour les paramètres de rumen si l'animal avait commencé une activité et elle supprime en plus tous les événements concernant l'animal.

2.2. Le générateur de nombres pseudo-aléatoires

Auparavant, le générateur de nombres pseudo-aléatoires du gnu était utilisé mais il n'était pas vraiment adapté puisqu'il ne permettait pas d'initialiser la graine. Le générateur de nombre pseudo-aléatoires a donc été modifié, on utilise désormais le générateur de L'Ecuyer avec mélange de Bays-Durham, des Numerical Recipes. Il est compris dans une classe *Random* qui fournit les lois uniformes, exponentielles et gaussiennes. Pour plus de souplesse, la graine est choisie dans l'interface et la graine est initialisée au début de la simulation, sa valeur reste la même pour toutes les réplifications.

2.3. Initialisation des paramètres globaux de la simulation

Pour initialiser les paramètres globaux de la simulation saisis par l'interface, comme on n'utilise plus le système de librairie, il fallait se servir d'un fichier pour que le simulateur puisse connaître les paramètres saisis. On se sert donc du fichier *archive.txt* qui recense les différents paramètres saisis. Le simulateur lit ce fichier dont le nom est passé en paramètre. Pour cela, il utilise une nouvelle classe *FichierConfigGestion* qui se base sur la classe *FichierConfigTroupeau*. Les valeurs lues par cette classe permettent ensuite d'initialiser les attributs de la classe *Parametres*. Enfin, le simulateur a également besoin du répertoire de sortie de la réplification courante puisque c'est l'interface qui gère les réplifications. L'exécutable du simulateur est donc appelé avec deux arguments, le fichier *archive.txt* et le répertoire de sortie de la réplification courante.

2.4. Fichier de configuration du ruminant

Le fichier de configuration du ruminant a été modifié pour pouvoir être utilisé dans l'interface. Ce fichier, qui était auparavant au format *cfg*, comprenait les caractéristiques de tous les animaux, les paramètres de chaque animal étant séparés par une ligne de caractères *. Comme ce caractère n'était pas reconnu dans le format *cfg*, le fichier a été transformé au format *csv* pour éviter de créer un nouveau type particulier de fichier.

Le fichier *ruminant.csv* se base donc sur le modèle du fichier *environnement.csv*, avec la liste des paramètres sur une première ligne et plusieurs jeux de valeurs ensuite, un jeu sur chaque ligne pour chaque animal. Il a donc fallu modifier la classe *FichierConfigRuminant* en conséquence. Elle n'hérite plus de la classe *Fichier* et la lecture du fichier se fait grâce aux fonctions de manipulation des fichiers en C.