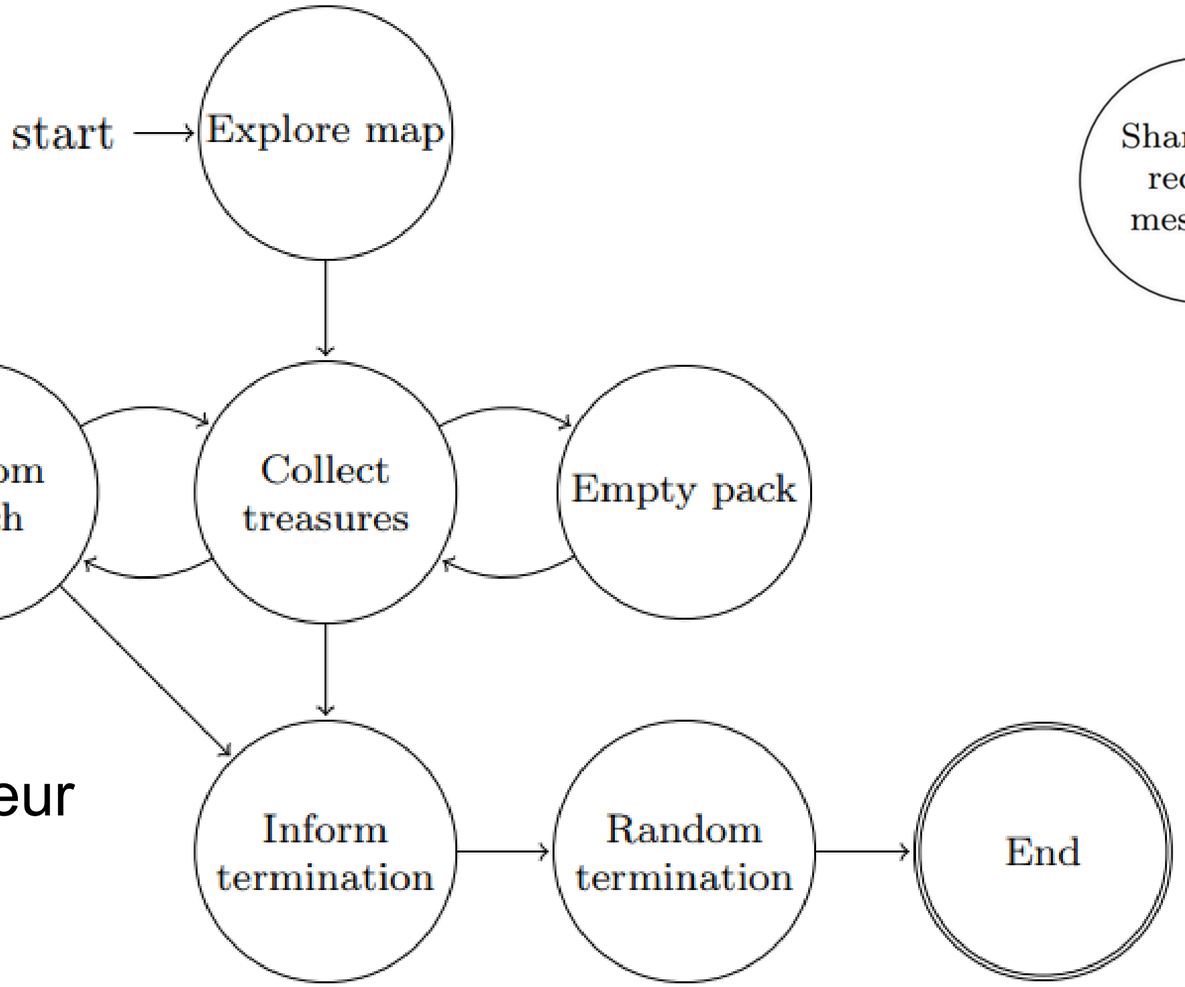
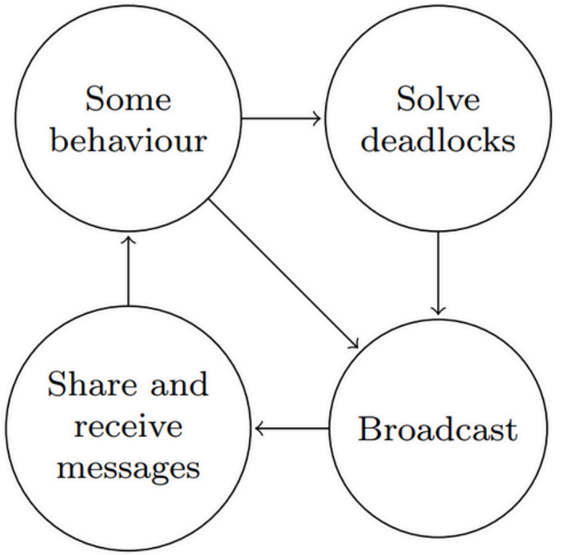
FSMBehaviour



collecteur



Composant de base

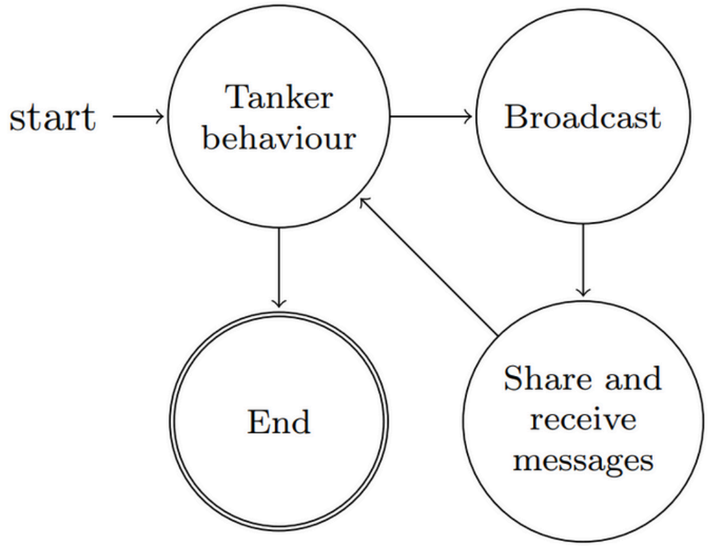
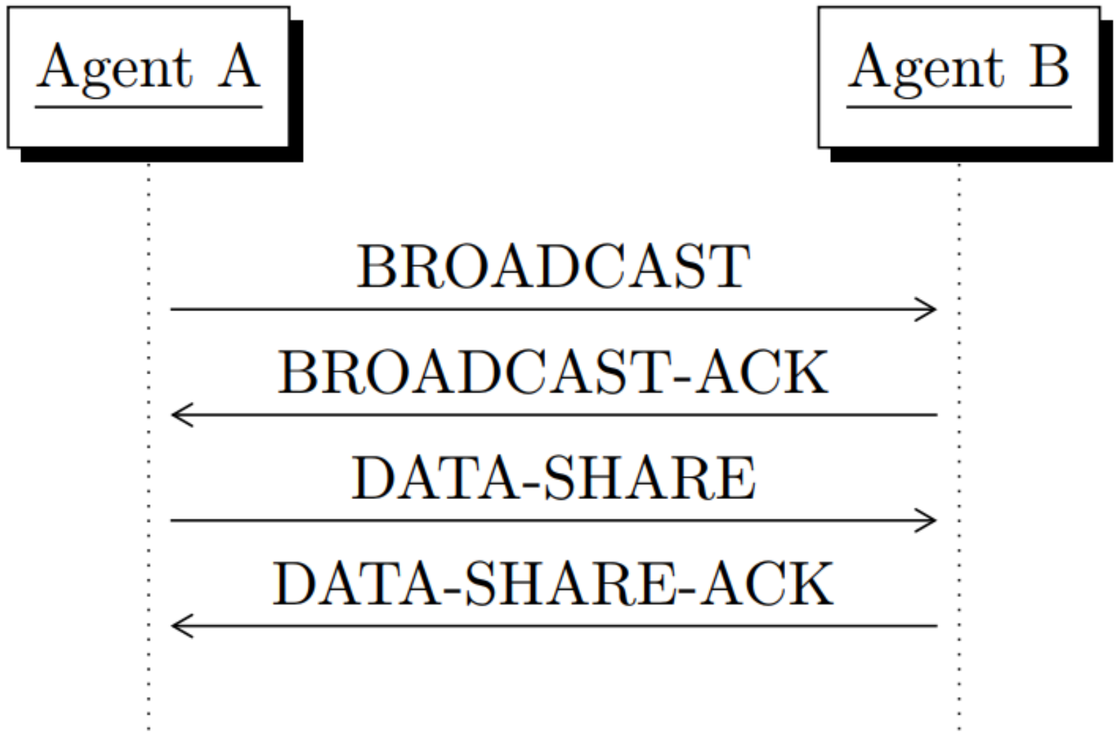


Figure 3: Automate associé aux agents silo.

Protocole de communication



Déplacement et exploration

informations partagées

Le DataContainer partagé entre agents contient :

- une représentation partielle du graphe exploré (graph),
- la liste des trésors connus (treasures),
- l’état des agents silos (tankerAgentKnowledges),
- les statistiques de collecte de chaque agent (treasureCollectionKnowledge),
- ainsi que les préférences individuelles (agentPreferences).

Déplacement et exploration

sélection de la destination

- Explore Map : les nœuds encore ouverts.
- Collect Treasure : les positions contenant des trésors.
- Empty Pack : un agent silo pour vider son sac.
- Random Search : combiner des positions aléatoires et des cases proches des silo pour rester synchronisé.
- Termination Inform : informer tous les silo non encore prévenus.
- Random Termination : choisir des positions finales distinctes aléatoires

Gestion des interblocages

stratégie

- Attente aléatoire : l’agent commence par attendre un temps $t \in [50,1000]$ ms, tiré aléatoirement. Cela permet de laisser la voie se libérer naturellement.
- Recalcul d’un chemin: l’agent recalcule un nouveau plus court chemin en enlevant l’arête bloquée ou vers un autre objectif candidat.
- Random move: Si aucune alternative valide n’est trouvée, l’agent effecte un déplacement aléatoire vers une case adjacente libre.

Gestion des interblocages

- Problème: plusieur agents bloqués en ligne peuvent mettre longtemps à se débloquer avec notre stratégie
- Amélioration:

```
if (agentKnowledge.mode().equals(AgentMode.EMPTY_PACKAGE)) {
  agentKnowledge.resetShortestPath();
  agentKnowledge.setAttemptPositionID(null);
  return;
}
```

- Limite: cela ne résout pas le problème en cas de exploration car aucun agent n’est en mode Empty Package. Nous retournons au shéma aléatoire.

Ramassage des trésors

```
public boolean canContinueToCollect() {
  return currentPackCapacity() > 0 && !pickableTreasurePositions().isEmpty();
}
```

```
if (agentKnowledge.canContinueToCollect()) {
  nextStateTransition = 1;
  return;
}
```

- Une amélioration: si le sac à dos n’est pas encore plein, l’agent continue à chercher un autre trésor pour remplir son sac.
  Cela réduit les allers-retours.

Terminaison

- Si certains trésors découverts lors de l’exploration ne sont pas totalement collectés, les agents passent à l’état Random Research pour tenter de les retrouver.
- Cependant, cette phase peut se répéter indéfiniment si un golem a déplacé un trésor pendant l’exploration. Les agents ne peuvent pas détecter qu’il s’agit d’un trésor déplacé, et ils le considèrent comme un nouveau trésor.
- Cela augmente le nombre total de trésors connus, ce qui fausse la condition d’arrêt du programme. À la fin, les agents cherchent donc des trésors qui n’existent pas, ce qui empêche le système de se terminer correctement.