UQÀM | Faculté des sciences

Université du Québec à Montréal

LES JEUX SERIEUX

Présenté par:

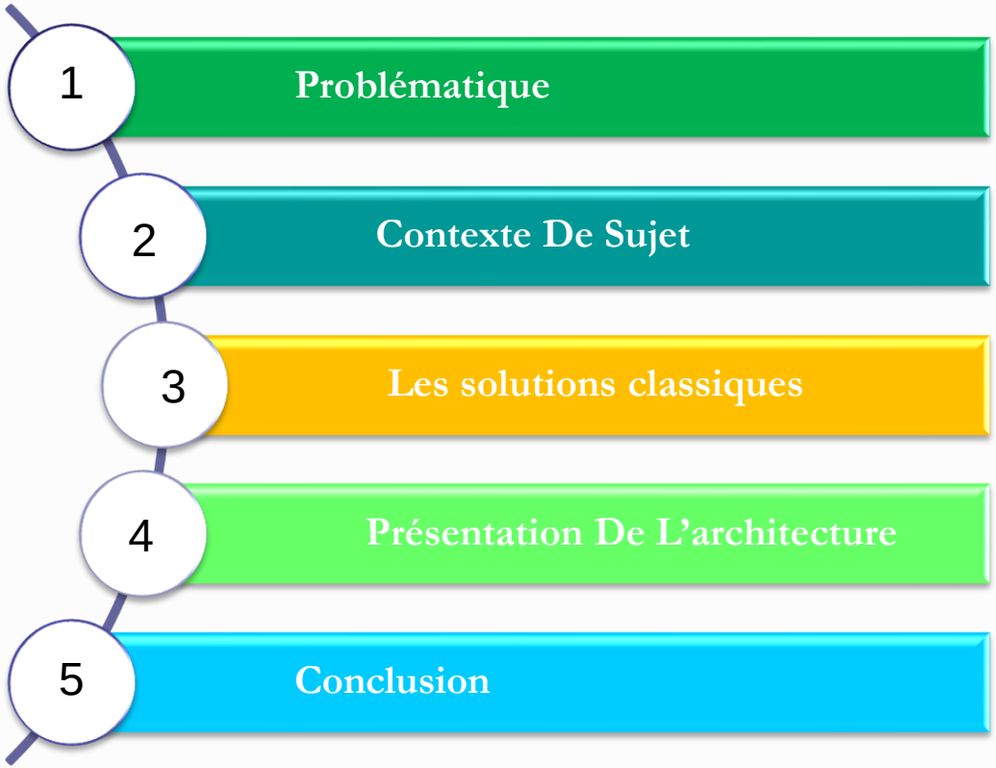
Wafa FENNANI

Architecture de planification et contrôle d'exécution pour infanterie pour les jeux sérieux

Intelligence Artificielle

Session Automne 2017

JEUX SERIEUX :
ARCHITECTURE DE PLANIFICATION ET DE CONTROLE D'EXECUSION D'INFANTERIE



Problématique

La simulation tactique dans l'infanterie : moyen indispensable pour: entrainement , évolution comportementale des soldats.

But: un environnement simulateur efficace* adapté aux exigences.

*Efficace= réaliste, intelligent, a moins coût

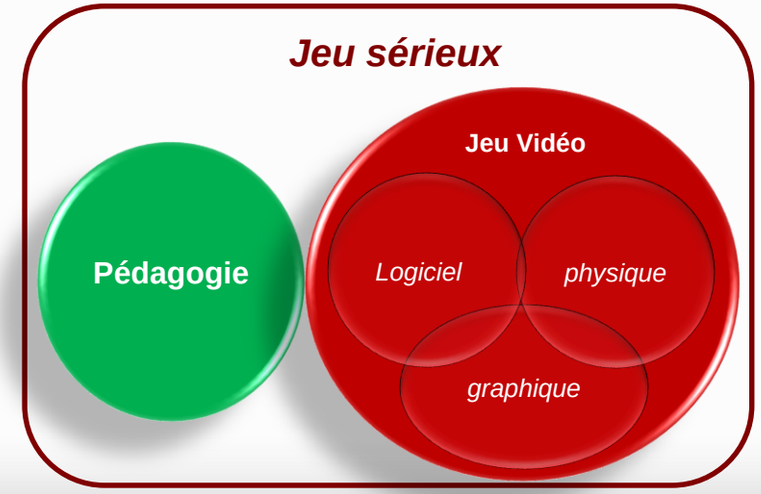
l’armée et les jeux sérieux

une approche : enseigner, former et développer de nouveaux concepts d'engagement chez les armées
réutilise la conception de **jeux vidéo**
offre un environnement **immersif** et **interactif** aux apprenants

C’est Quoi un jeu sérieux?

"un Jeu sérieux est un jeu vidéo (avec un environnement réaliste ou artificiel) auquel les auteurs rattachent une composante pédagogique. «

Louise SAUVE (chercheuse au Canada)

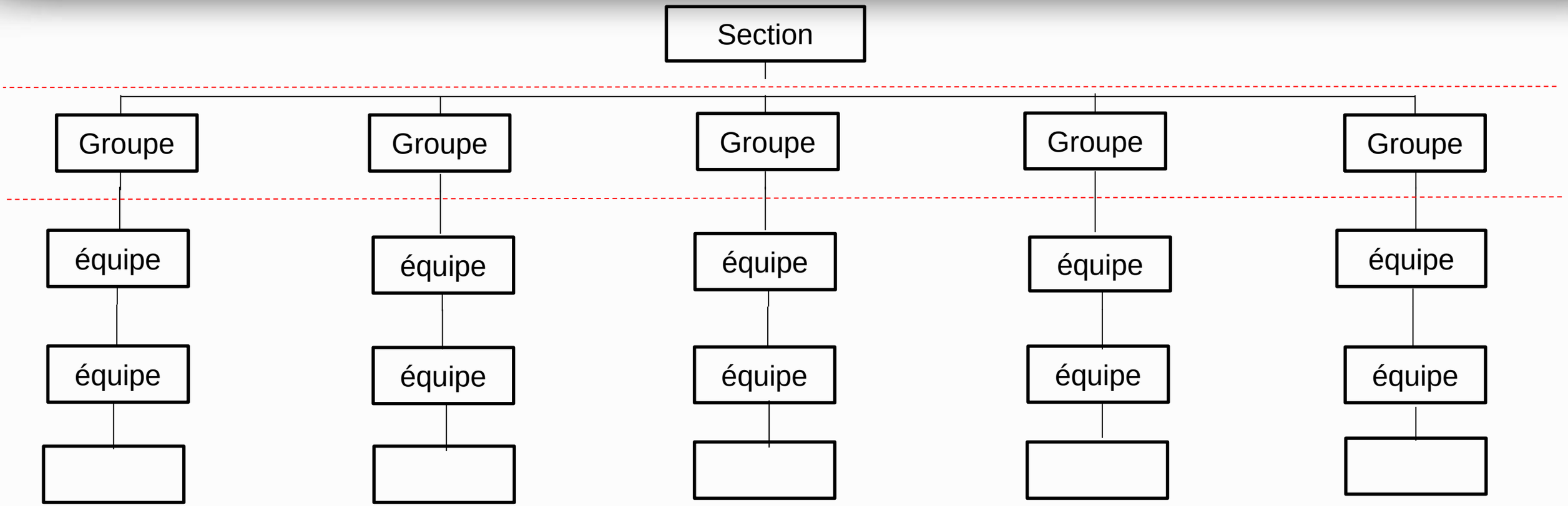


Combat Medic

Apach

ASCEND

America's Army[2002]



Organisation d'une section d'infanterie d'armée

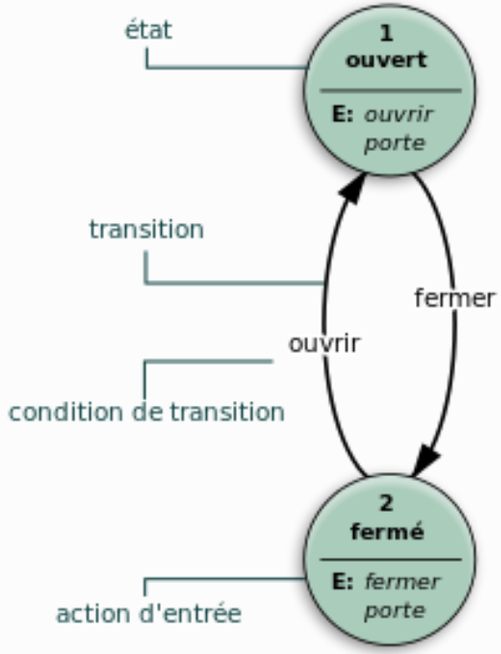
Différentes exigences soulignent le comportement des agents ainsi que la conception architecturale d'un tel jeu sérieux:

Apprentissage	Entrainement	Développement de Conception
Conceptuelle: Scenarios de base de l'infanterie facile a programmer	**Gérer un grand nombre de soldats, construire une situation complexe.**	comprendre l'impact de nouvelles tactiques, techniques et procédures opérationnelles
Comportementales: Comportement réaliste du soldat simulé	**Insiste sur la coordination, les ordres et les capacités de rapporter de l'utilisateur final.**	
Des séquences d'action basiques analysables lors de revus après l'action	les performances des joueurs sont analysables lors d'une revue après l'action	**évaluer les nouvelles performances du système et les interactions homme-machine**

comportement = réaliste + intelligent → une solution efficace

machine à états finis:

- état :
  associé à une action
  exécuter la même action si l'état n'est pas modifié
- Transition = passage d'un état a état
  associée à une condition
  exécuter la transition si la condition est satisfaite



Comportements plus complexes
Nombreuses actions
Situations différentes

→ La modélisation par machine a états: tâche complexe

La planification d'actions:

- Introduite dans l'industrie du jeu vidéo en 2005
  - ❖Jeu F.E.A.R [2005]

Planification d'action: Jeu F.E.A.R

- F.E.A.R (ou FPS) est un jeu simulateur d'infanterie de tir en vue subjective
- (l'acronyme FPS anglais) First-Person Shooter)
- Implémente l'algorithme GOAP (Goal Oriented Action Planning)

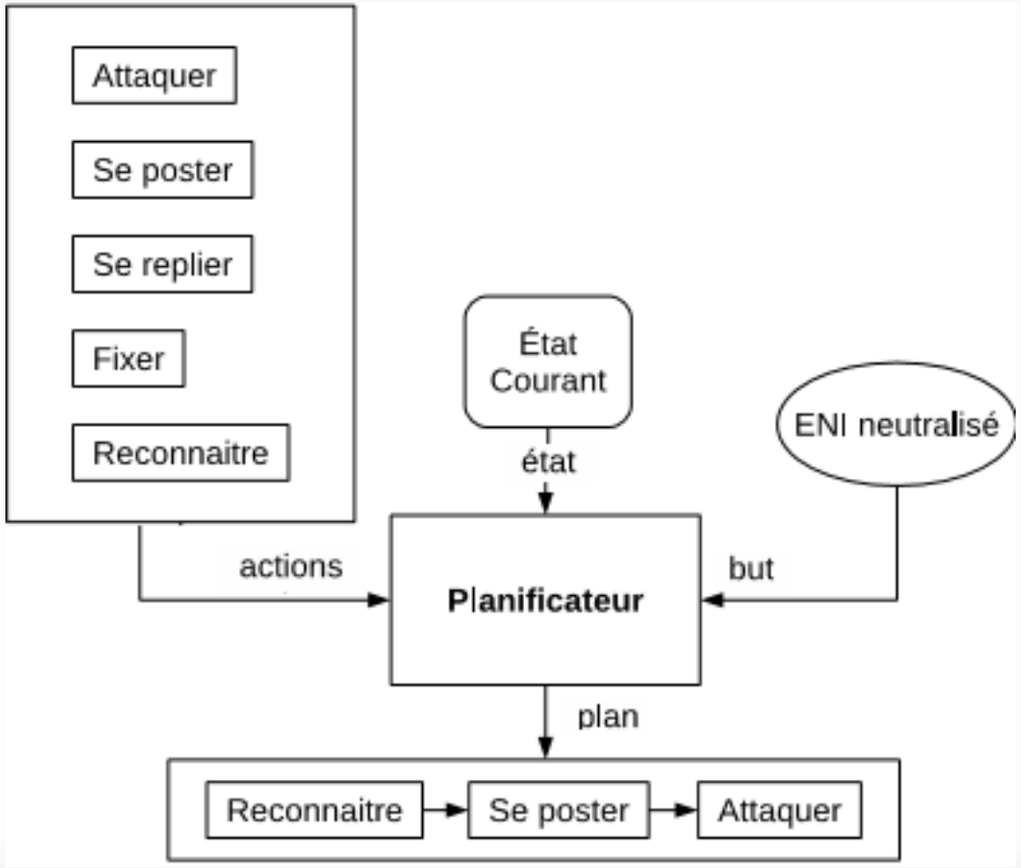
GOAP:

- Automatise la recherche des plans optimaux
  - Des comportements dynamiques
  - Comportement simplifié
- Tableaux statiques en C++
- Actions en C++ (non un langage PDDL)
- A*
- H= somme formules satisfaites / a l'état courant et au but

GOAP:

Le comportement des agents et la coordination:

- Agent: son planificateur, sa représentation : Approche mono-agent



- L'enchainement des actions n'est pas modélisé explicitement
  - ➢ calculé automatiquement à partir d'un modèle logique des actions spécifiées dans un « domaine de planification ».

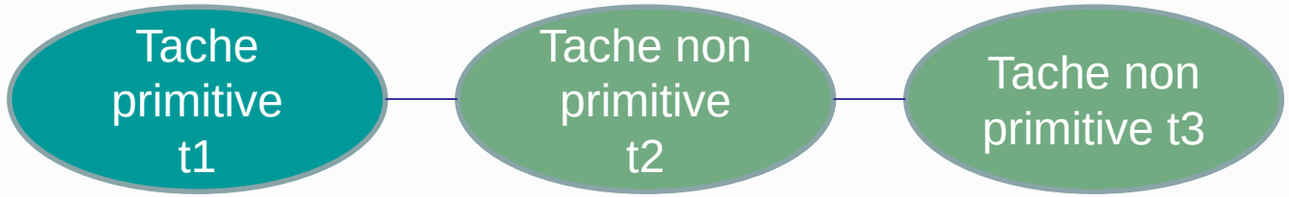
- Modélisation simple
- Temps de calcul énorme.

Les algorithmes de planification

- le nombre d'états parcourus exponentiel par rapport à la taille du plan retourné

Les FPS récents implémentent la planification « réseau de tâches hiérarchique » HTN (Hierarchical Task Network).

En planification HTN:
actions associées à des taches primitives*,
on les ajoute des taches non primitives**,
sous forme d'un réseau de sous-taches,
ordonnées entre elles.



Des règles (= méthodes) spécifient la décomposition des tâches.

- modélisées dans le «domaine de planification»

*tache primitive = tache non décomposable
**tache non primitive = tache décomposable

Comment la planification se fait?

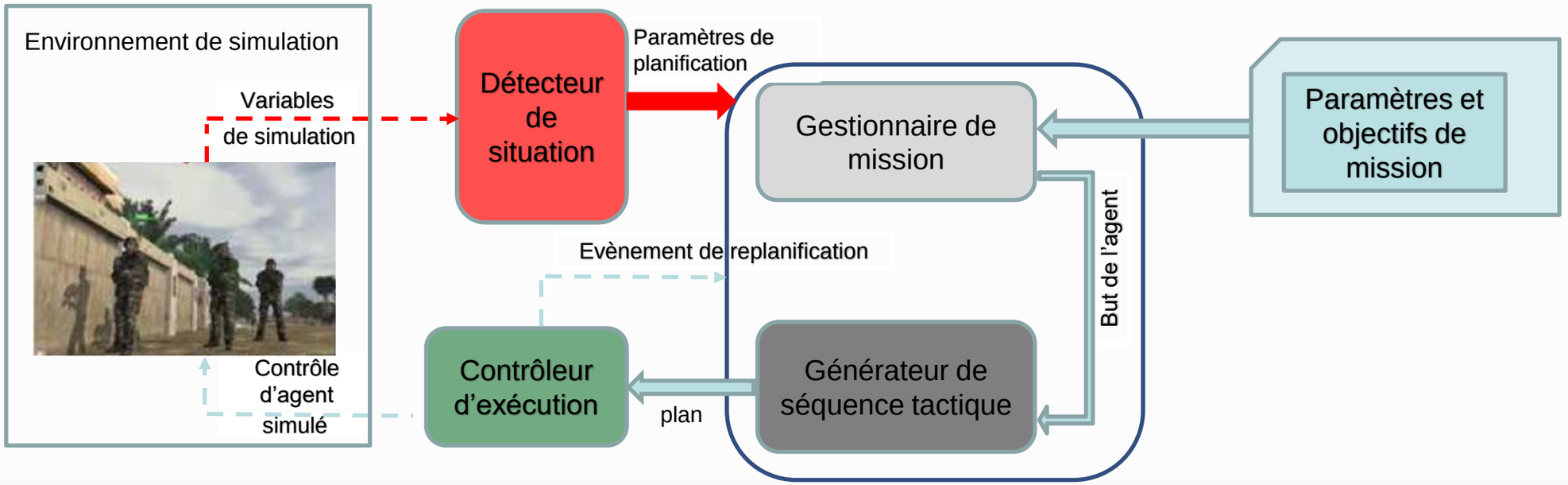
Planificateur HTN: Définition

1- Choisir une méthode pour une tache composée
2- l'instancie
3- décompose la tache en sous-tache
4- choisi et instancie des méthodes
5- décompose les sous-taches

Jusqu'à tâche primitive

Si plan irrealisable ⇨ marche-arriere + autre methode

Architecture de planification et d'exécution pour un jeu vidéo



Le gestionnaire de mission

1. Résolu les plan a long et a moyenne terme pour les unités des niveaux supérieurs.

   mission de planification et ordonnancement

   - définie une course d'actions pour trouver les objectifs de mission

   mission de décomposition de plan pour les unités inferieurs

- Prend en considération:
  - ✓ la structures du terrain
  - ✓ les capacités des unités
  - ✓ la coordination entre les unités

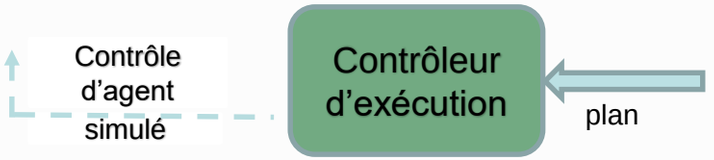
2. Gère les pannes des plans tout au long de la chaine de commande

Chaque niveau hiérarchique → un ordre opérationnel (OPORD)

- Organise les unités inferieurs
- Alloue les Ressources nécessaires

Contrôleur d'exécution

- Exécute les actions (plan) provenant du générateur de séquences
- agit avec l'environnement de simulation
  - pour chaque action: défini la mobilité, l'orientation, la position du soldat
  - Demande des requêtes au soldat:
    - ✓ Un point de passage a atteindre
    - ✓ évaluer l'inter-visibilité entre deux zones
    - ✓ trouver des objets menaçant dans le champs de vision



Préconditions vérifiées
(les actions précédentes achevées)

requête émise avec succès
(un point de passage a été trouvé)

coordination spatio-temporelle satisfaite

Logique d'évaluation de succès

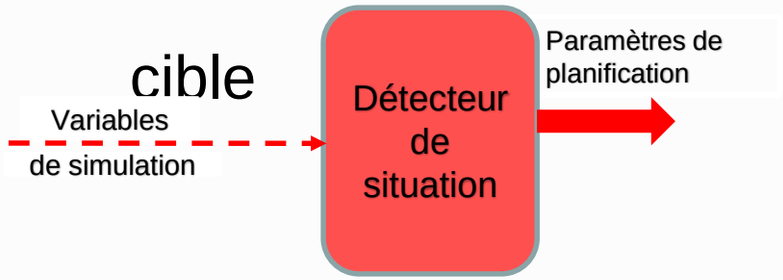
	équipe	section	groupe	Bataillon
Tps limite	<1s	<5mn	5 à 15mn	>1h
unité	10	4	16	70

❖ Action non exécutée ➡ Evènement de replanification déclenché

Perception de l'environnement et détection de menace:

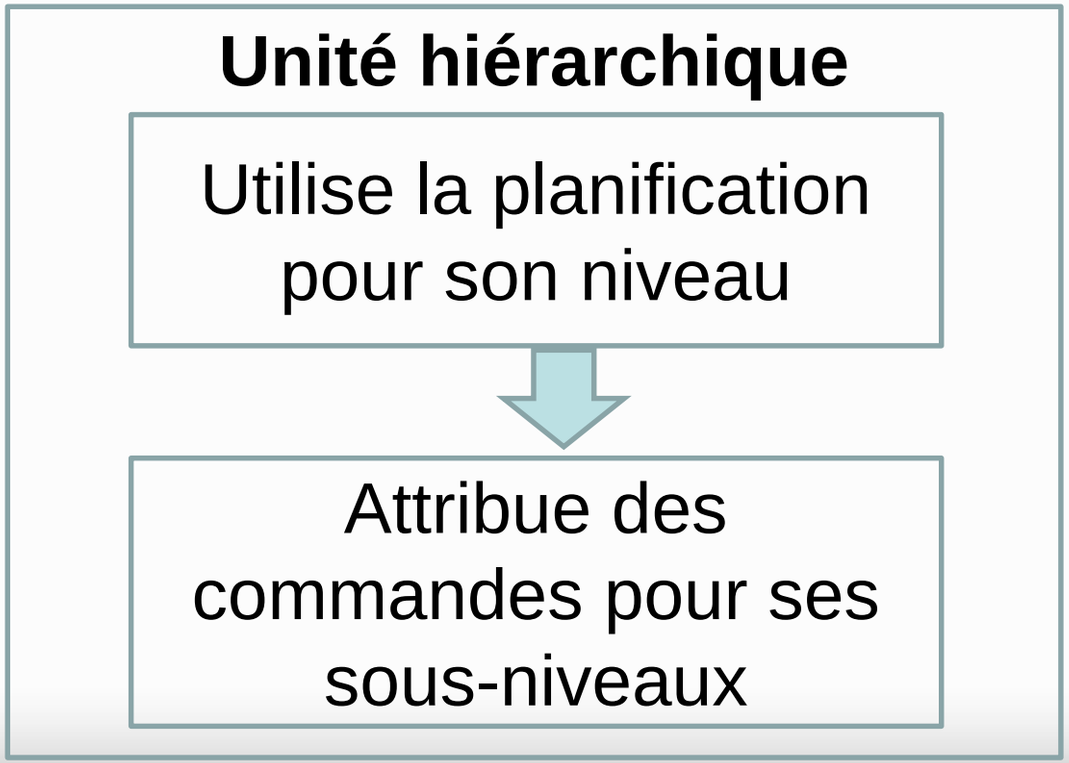
Situation tactique est maintenu

* un système d' acquisition et de poursuite de cible



- Gestion des rapports d'observations
- Désignation des cibles
- L'estimation de la position de l' ennemie a court terme
- la prédiction de la trajectoire de l' ennemie a long terme

Générateur de séquence tactique



- Niveau supérieur: vérifie la coordination entre les unités.
- Une unité est autonome dans son domaine d'action.
- Une exigence spatio-temporelle est exigée par le contrôleur d'exécution.

Générateur de séquence tactique

L'environnement de simulation en évolution → Plan fréquemment invalidé → Replanification d'une partie du plan global

⚠ Planification classique: replanification globale : tous les actions des niveaux → plan global =tâche complexe

↓

invalidé par un évènement déclenché

Générateur de séquence tactique

➡ planification hiérarchique HTN

- complexité des recherche réduite
- un évènement affecte une partie du plan global

Domaine de planification de l'infanterie

- Les unités de plus bas niveau: évènement de replanification fréquent

Comportement extrêmement simple a planifier:

Modélisation: formalise SHOP (Simpel hierarchical ordered planning)

➡ domaine de planification:

tâches de surveillance={d'actions simple}

SHOP : planificateur qui génère des étapes de chaque plan
dans l'ordre d'exécution en sachant l'état courant
a chaque étape

Exemple

Requête:

(defproblem problem monitor
((down soldier1) (detected soldier1 target sector))
((monitor soldier1 sector)))

Plan retourné:

(:task !report soldier1 target)
(:task !stand-up soldier1)
(:task !use-weapon soldier1 target

Domaine de planification:

```
(defdomain monitor (
(:operator (!use-weapon ?soldier ?target) ((up ?soldier)) () ())
   ))

(:method (use-weapon ?soldier ?target)
; soldier cannot use weapon if down
((down ?soldier))
((!stand-up ?soldier) (!use-weapon ?soldier ?target))
()
((!use-weapon ?soldier ?target)))
```

- La programmation en C++ pour la premiere implementation de SHOP2
- Le planificateur implémente les principales fonctionnalités
- Composant analyseur capable de lire les domaines de planification
- Les plans d'actions pour un bas niveau sont calculés a moins d'une milliseconde avec l'implementation de SHOP2

Conclusion

Bien que les différents composants de cette architecture permette de réagir a l'évolution dynamique de l'environnement de simulation [cela est assurer par l'implication du module de contrôle d'exécution , ainsi que le module de perception de l'environnement].

Mais les planificateurs SHOP (base sur HTN) qui sont implémentés ne retournent que des plans totalement ordonnés , qui ne sont pas adaptés pour la coordination des soldats devant agir en parallèle les uns aux autres.