Advanced Real-Time Hierarchical Task Networks
A New Approach

Tomohiro Mori
AI Engineer
SQUARE ENIX CO., LTD.
moritomo@square-enix.com

Kousuke Namiki
Senior AI Engineer
SQUARE ENIX CO., LTD.
namikous@square-enix.com
Tomohiro Mori
AI Engineer: SquareEnix

Tomohiro Mori researched about CharacterAI and Animation in Future University Hakodate. He joined SquareEnix in 2019.

He is researching AI and Animation
- Full Procedural Animation
- Character AI using Hierarchical Task Network

Kousuke Namiki
Senior AI Engineer: SquareEnix

Kousuke Namiki has been involved in the game industry since 2008, and joined SquareEnix in 2012.

He worked in the development of titles such as
- FINAL FANTASY XV
- KINGDOM HEARTS III
- FINAL FANTASY VII REMAKE
as Game AI engineer and QA automation AI engineer.

Currently, he is engaged in research and development of next-generation AI system in Advanced Technology Division.
Introduction
Choices of Game AI

- **Candidates of Game AI Systems**
  - Finite State Machine
  - Behavior Tree
  - GOAP
  - Hierarchical Task Network (HTN)

- Recent Game AI has Excelled in immediate response.
- **Long-term behavior** is a major challenge today.
HTN Introduction

• What is HTN?
  • Traditional planning method
  • Planning in a static domain
    • e.g. Delivery Optimization in Logistics

• Expectations
  • Predict future situations
  • Take deliberate action
An Example of HTN

- There are two types of task in HTN.
  - Primitive
  - Compound

- Decompose
  - Break down a Compound Task into Multiple Tasks

- Hierarchical Planning
  - Break down large problems into sub-problems
HTN in Games

- HTN excels at planning for a static environment
- Console market has seen a rise in real-time games
- HTN is not designed for dynamic environments
- Player behavior is unpredictable to HTN
The Problem of Uncertainty in HTN

• In HTN, after the planning phase is over, tasks are arranged in a chain and executed in sequence.

• Once a task chain is created, HTN can’t modify that. If the environment changes, the existing plan is discarded and new plan is recreated by re-planning.

• The more frequent the re-planning, the less able the AI will be to think long-term, and the more it will turn into a reactive AI with a narrow view.
Plans are nothing
Planning is everything
Our Motivation

- Even though various situations change in real time, we want to achieve more long-term behavior with planning.

- To take advantage of planning, we want to minimize the number of re-planning as much as possible.

- Instead of symbolic planning, we want to make planning more relevant to the game situation by using simulator.
Our Approach : ART-HTN

1. Multi Scenario Plan
   Our planner should have multiple scenarios in order to handle uncertain situation change.

2. Simulation Planner
   A game simulation would be more appropriate than symbolic planner. Better planner might reduce amount of re-planning.

3. Executor - Planner Role Separation
   By giving more authority to the executor of the plan rather than the creator of the plan, we thought we could operate the plan flexibly.
1. Multi Scenario Plan
1. Multi Scenario Plan

- Reuse temporal states
  - Traditional Planner discard temporal states in search phase and returns only a final task chain of plan.

- But temporal states can be used as Sub-Plans

- Our Multi Scenario Planner returns a plan with multiple branches. Planner leave branches in the middle state of the search whenever possible.
Situation-Task Network

- Multi Scenario Plan holds Tasks and Situations.
- Situation is a simplified model of game world, containing all simplified data in game world:
  - Character's position
  - HP
  - Internal condition such as state-id of hostile AI agent.
- Situation has links to previous and next situations forming a graph.
- We call this graph the “ST-Network”
Variation of Tasks

- **Primitive Task**
  - Simple action task which has precondition and simulate function.

- **Composite Task**
  - **FSM-Task**
    - To simulate opponent behavior, our planner can use FSM in planning phase
    - FSM can be used as a model of opponent behavior
  - **Step-Task**
    - Simple FSM, works like a Sequence Node of a Behavior Tree
    - Can have simple loops or small number of branches
    - Can describe directional actions with start and end states with
2. Simulation Planner
2. Simulation Planner

- ST-Network Generation
- Terminal Situation-Node Expansion
- Enemy-based Branch Generation
Task Assigning

- To Expand Terminal Situation Node, Decision Makers choose a task for every character.

- Decision Maker can assign new Primitive-Task or Composite Task when the precondition matches to the situation.

- Decision Maker can Expand Current Composite Task when the node allows.
Expand Composite Task

- Examples of Expansion of Composite Task
  - When FSM has transition to state a, b, c.
  - The FSM task can be expanded to a, b, c, copied instance.
  - All data in FSM-Task copied before the task execute state transition to make a, b, c versions of task instance.
Expand Composite Task

- Every agent’s decision-maker can conduct such process. So total amount of Situation Nodes tend to become huge.

- To Reduce ST-Network size, expand function receive threshold.

- The task can be expanded only when the random number exceeds this threshold.
3. Executor
Decomposition of Task

- ST-Networks choose best task to use
- Tasks are only data, cannot be executed as is
- Executor Process
  - Select High evaluation Task
  - Decompose the selected Task into Operators
  - Execute the Operators in order
Planning is nothing
Execution is everything
Example Demo

Ridge Shooting
Hill detection Algorithm

- Detect contour lines on the terrain
- Find orientation and distribution of hills by PCA
- Rough granularity and Fast algorithm for simulation
Without Ridge Shooting

View video (click)
Winning with Ridge Shooting
Final Demo
Reconnaissance-in-Force

- Eliminate weak targets while doing recon
- If a high-risk situation is encountered, attempt a retreat using a smoke screen
- Use powerful weapons to destroy enemies or make them give up the chase.
View video (click)
Summary

• In order to achieve long-term behavior in dynamic environment, HTN system should be improved.

• We present “ART-HTN” which consists of 3 features.
  • Multi Scenario Plan
  • Simulation Planner
  • Executor

• We verified that on runtime demo with dynamic environment.
Related Session

Monday, July 19 3:20pm - 3:50pm

GDC

Animation Summit
From Design:
Full Procedural Animations for Mechs

Tomohiro Mori
AI Engineer
SQUARE ENIX CO., LTD.
moritomo@square-enix.com

Kousuke Namiki
Senior AI Engineer
SQUARE ENIX CO., LTD.
namikous@square-enix.com
Advanced Real-Time Hierarchical Task Networks
A New Approach

Tomohiro Mori
AI Engineer
SQUARE ENIX CO., LTD.
moritomo@square-enix.com

Kousuke Namiki
Senior AI Engineer
SQUARE ENIX CO., LTD.
namikous@square-enix.com