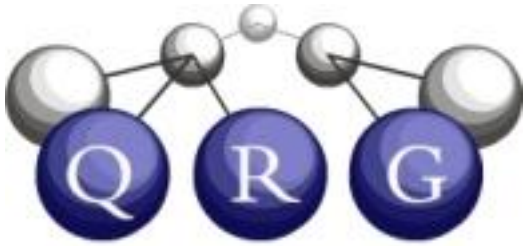
Operationalizing Tactical Representations

Thomas R. Hinrichs

Kenneth D. Forbus



Advances in Cognitive Systems - 2021

The Problem

How can we represent and use general tactical knowledge to improve performance and learning in a broad, complex domain like Freeciv?

Pursue mobile adversary?

Should defender attack first?

Tactics

Definition: *a composite, goal-directed behavior that is sensitive to adversarial actions and other external events*

Tactics are not limited to military tactics, but can include economic investments, legal tactics, social & political tactics.

We want tactics to provide large building blocks for learning and to eventually learn to make better decisions about selecting and applying tactics.

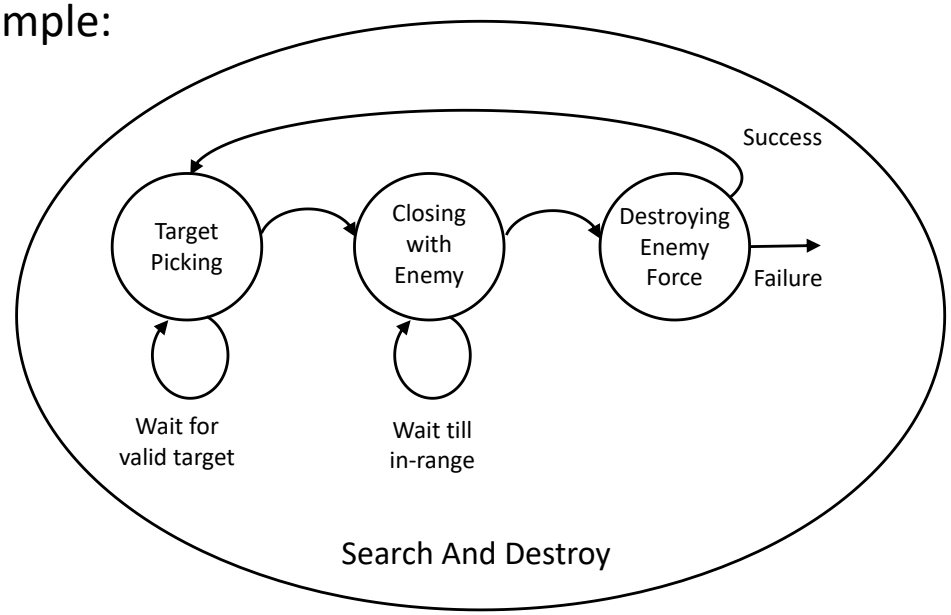
Representational Desiderata:

- Penetrability – ability to reason about reqs, resources & dependencies
- Expressiveness – capable of representing complex scheduling
- Transferability – not specific to Freeciv
- Learnability – decomposable to individually learnable decisions

Representing Tactics

We treat tactics as Neo-Davidsonian* events with named roles and internal structure. This allows incremental instantiation as resources and information become available.

Example:



*as opposed to positional notation of HTN tasks

```
(isa SearchAndDestroy TacticType)
(comment SearchAndDestroy "A SearchAndDestroy mission
is the extended event of looking for targets, maneuver
closing on, and destroying them.")
(genls SearchAndDestroy MilitaryTactic)
(genls SearchAndDestroy OffensiveTactic)
(isa SearchAndDestroy DurativeEventType)
(genls SearchAndDestroy PurposefulAction)
(participantType SearchAndDestroy performedBy MilitaryAgent)
(participantType SearchAndDestroy eventOccursAt GeographicalRegion)
(properSubEventTypes SearchAndDestroy TargetPicking)
(candidateProperSubSituationTypes SearchAndDestroy MilitaryManeuver-Offensive)
(candidateProperSubSituationTypes SearchAndDestroy Waiting)
(properSubEventTypes SearchAndDestroy ClosingWithEnemy)
(properSubEventTypes SearchAndDestroy DestroyingAnEnemyForce)
(focalProperSubsituationTypes SearchAndDestroy DestroyingAnEnemyF
(properSubEventTypes SearchAndDestroy RescheduleParentTask)
(startsAfterEndOfInSituationType SearchAndDestroy
  MilitaryManeuver-Offensive TargetPicking)
(startsAfterEndOfInSituationType
  SearchAndDestroy Waiting MilitaryManeuver-Offensive)
(startsAfterEndOfInSituationType SearchAndDestroy RescheduleParentTask
  DestroyingAnEnemyForce)
```

[…]

Connect to rest of ontology

Participants in this tactic

Types of subevents

Temporal constraints

Operationalizing Tactics

Definition: *instantiating and translating an event type into plans and executable actions supported by the architecture.*

Given a small library of game-independent tactics:

- select appropriate tactic types,
- bind participants to available game entities,
- translate to game-specific HTN tasks and primitives, and
- schedule execution such that triggering conditions and sequencing constraints are honored.

The Companion agenda supports a priority queue of tasks, looping behavior, interleaved execution and tactic instantiation, and error handling tasks.

How the tactical planner operationalizes

Because the goals, available entities, and primitive actions are expressed in the vocabulary of the game, while tactics are expressed in a domain-independent vocabulary, the tactical planner must dynamically map between them.

- It constructs and caches a discrimination tree to match game goal types to tactic types
- It climbs the isa hierarchy to match game entities to tactic participant roles
- It maps leaf event types to game action primitives or HTN tasks
- It translates scheduling constraints into Companion-native precedent relations on an agent's agenda.

Qualitative Evaluation

With the addition of 3 initial tactic types, the Freeciv player now:

- builds out road networks across its civilization,
- pursues mobile adversaries, and
- learns when and when not to attack approaching invaders.

With respect to the desiderata:

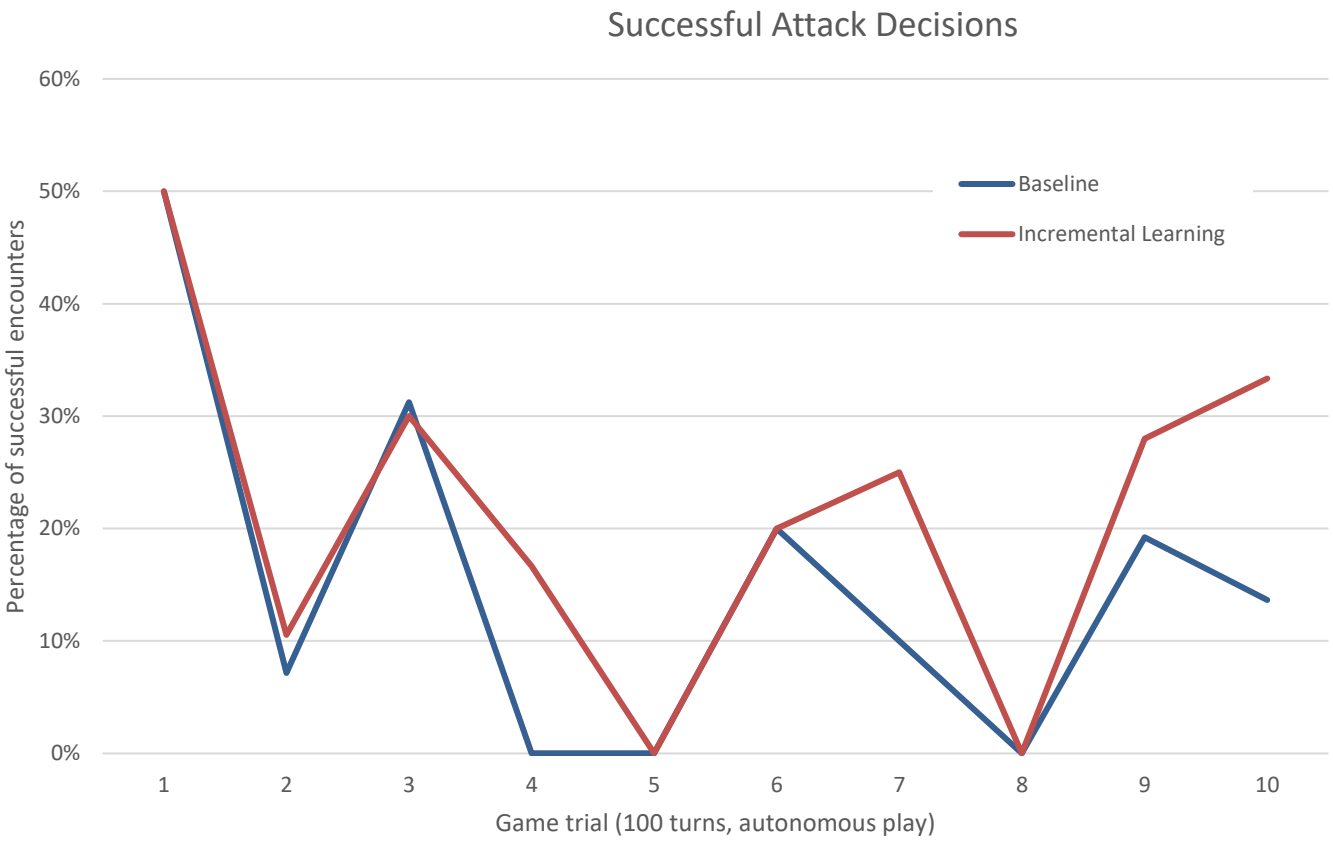
Penetrability: There is an accessible audit trail from actions to tactics to goals. Behavior is explainable.

Expressiveness: The representation supports looping, waiting, failure handling, and structured behavior. It directly supports contingency planning.

Transferability: Tactics are not specific to a game, though the game entities and actions must inherit from the general ontology.

Learnability: By clearly distinguishing decisions from the intrinsic structure of a tactic, it is possible to learn to improve performance on that tactic.

Example: Attack or Defend?



Improves success rate of attacks by 19% over 10 games

Conclusions

Background tactical knowledge can aid learning by providing “10 league boots” for exploration and by clearly delineating learnable decisions and parameters.

The purpose and structure of a tactic should not be specific to a concrete domain, but criteria for selecting tactics and binding participants should be learnable decisions.

The mechanisms for operationalizing a tactic for a particular (game) simulation and execution architecture are not conceptually difficult, but are critically important for realizing performance benefits.