UNCLASSIFIED

Australian Government
Department of Defence

Machine-Learning & Recommender Systems for C2 of Autonomous Vehicles

Glennn Moy

on behalf of

Don Gossink, Glennn Moy, Darren Williams, Kate Noack
Josh Broadway, Jan Richter, Steve Wark
Planning and Logistics, Decision Sciences, DST Group, Australia

Human-Autonomy Teaming

- Overall project – Balances two components:

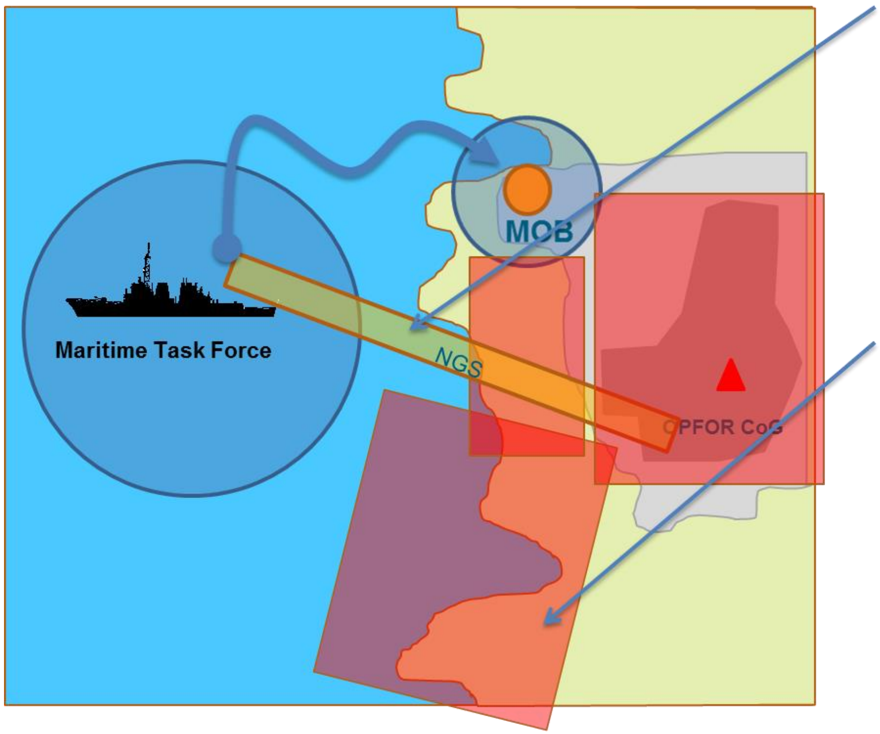
Autonomy
Lead: Don Gossink

HAT - Autonomy Team:

- Don Gossink
- Glennn Moy
- Darren Williams
- Katherine Noack
- Josh Broadway
- Jan Richter
- Steve Wark

HAT Context

Recommendations for
Command & Control of
Multiple UxVs



IMPACT

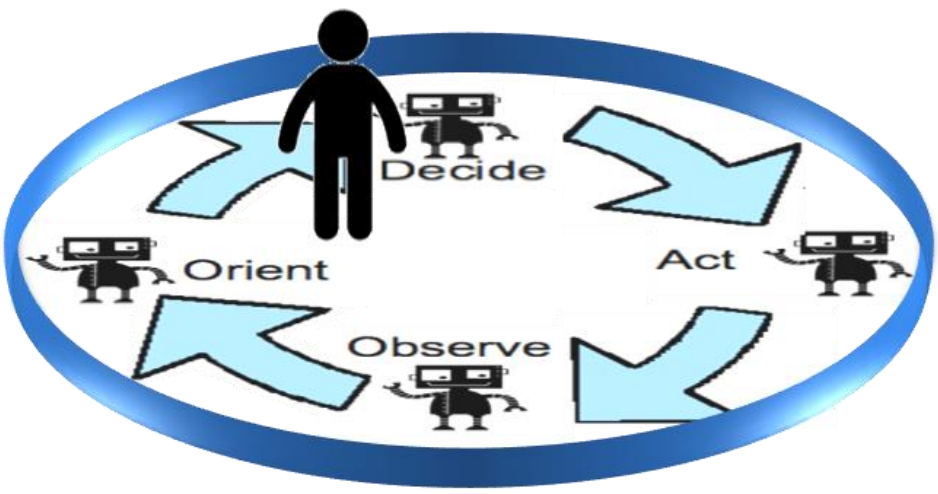
- U.S. Project aimed at:
  - Developing Intelligent Multi-UxV Planner with Adaptive Collaborative/Control Technologies
  - **Key Concept:** High-level, goal-oriented plays

Recommender System & Play-Monitor:

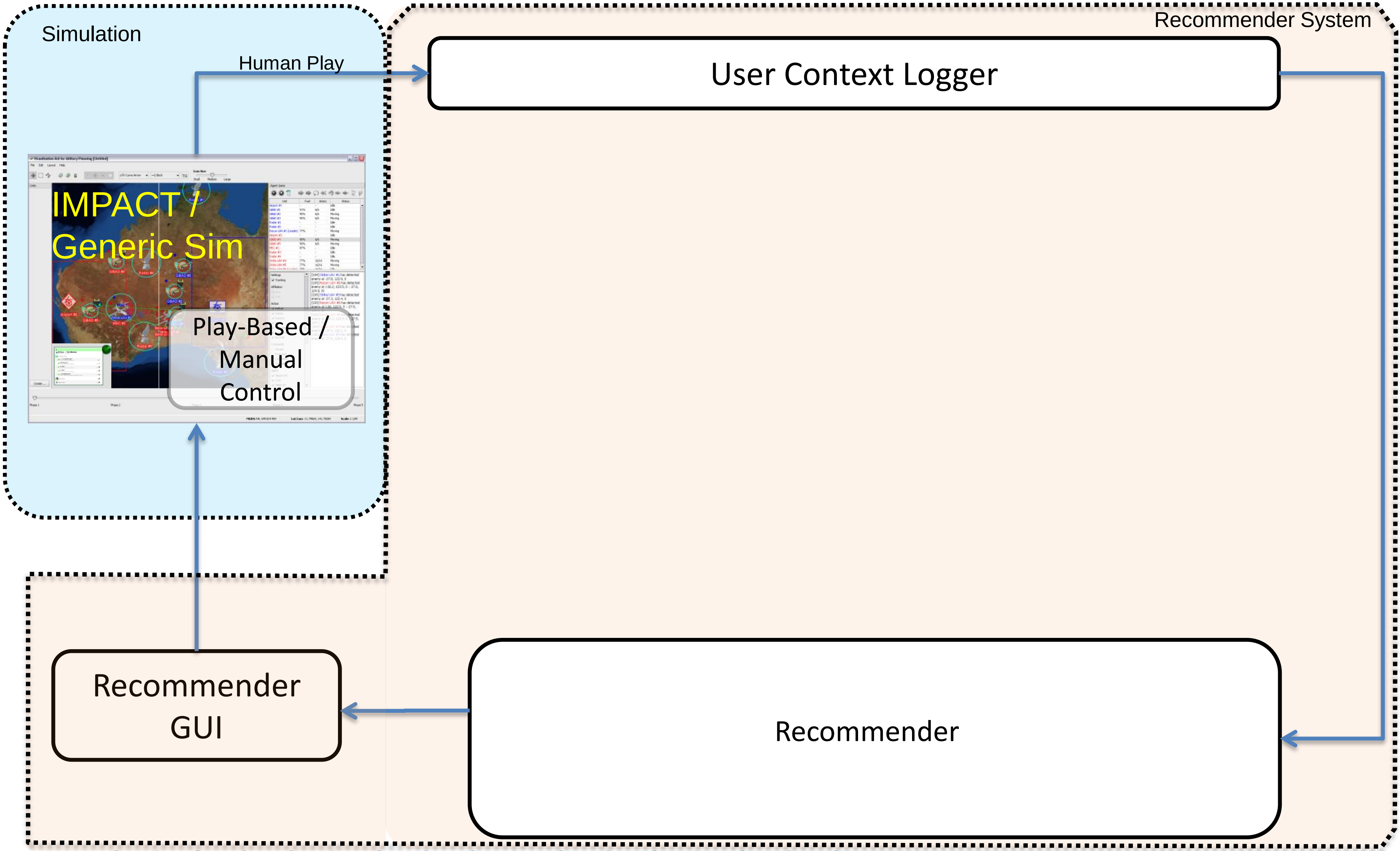
- Goal:
  - Develop advanced Recommender System(s) to reduce the cognitive burden on operators through:
    - Recommendations, alerts and constraints.

      over the top of
    - Lower-Level Autonomy

"Human on the loop"



Top-Level Architecture:



HAT Challenge

- Limited access to IMPACT or other multi-UxV control system
- Need to integrate recommendations into various (unknown) system components
  - **Solution:**
    - Recommender *loosely coupled* to the underlying system/simulation.
    - Recommender that can *learn* recommendations at a range of C2 levels.
    - Recommender techniques that work:
      - When heuristics are not known
      - In new contexts (not previously prepared)

Inspiration & Approach

- *Build something that can learn…* *(like we learn???)*

Eg: Humans try out new moves and seeing what works

Eg: Humans think about what works & what others might do in response to my moves.

	Self Learning	Learning from Others
Experience	Trying	Watching
Logic, Rules & Heuristics	Reasoning	Explaining

Eg: Humans watch a good player and learn to recognise good board positions

Eg: Humans learn by having a new strategy explained

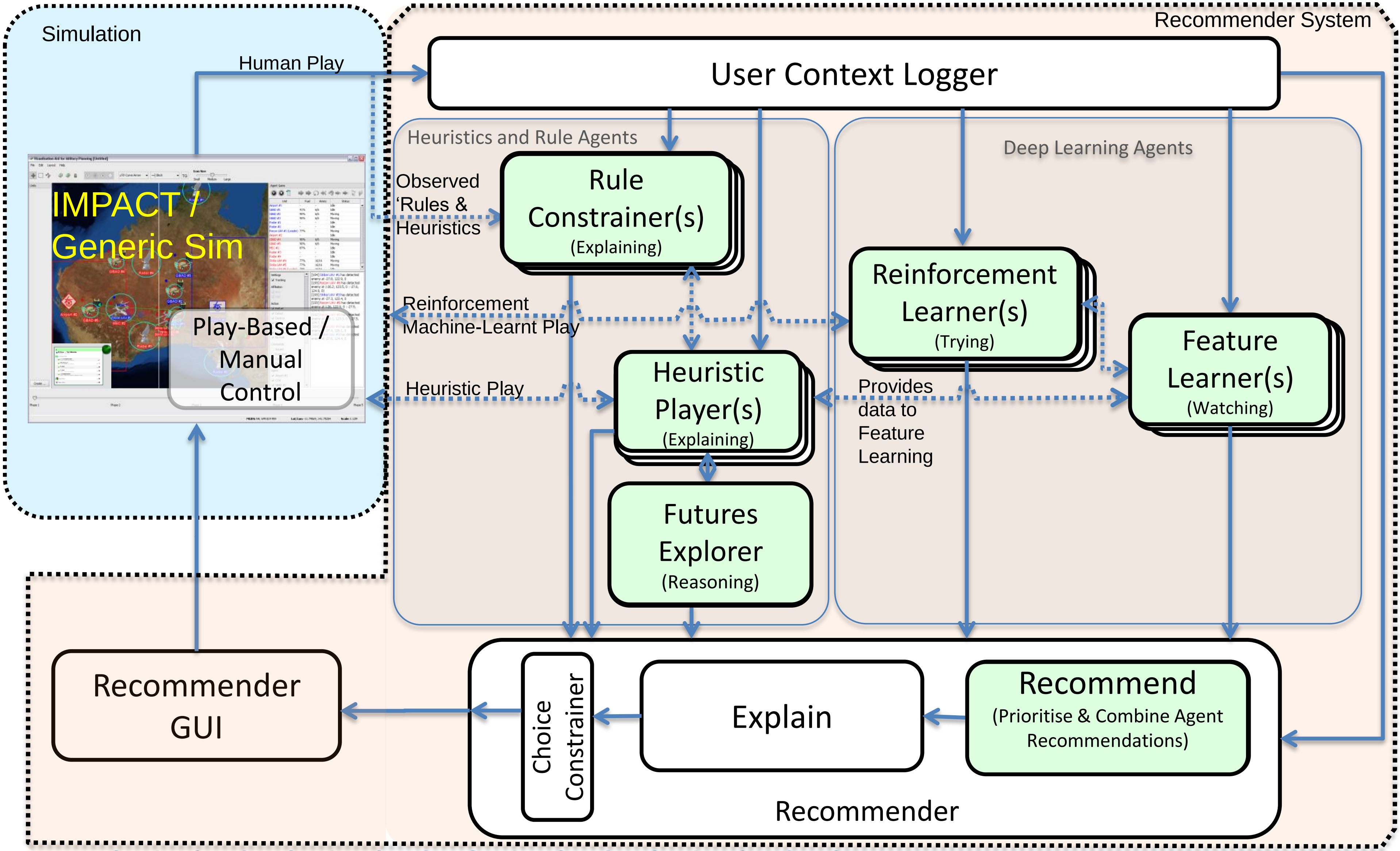
*(NB: about gathering and structuring data to learn…not **how** we learn)*

Techniques & Requirements:

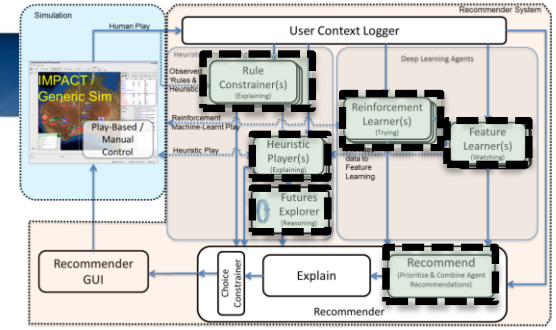
	Self Learning	Learning from Others
Experience	**Trying:** • **Techniques:** Reinforcement Learning • **Requirements:** A simple model that machine can control that it can undertake re-enforcement learning on.	**Watching:** • **Techniques:** Supervised Deep Learning • **Requirements:** **Data** from expert playing a simulation that it can watch. • Human Expert or Heuristic player that mimics an expert

Logic, Rules & Heuristics	**Reasoning:**
	• **Techniques:** Monte-Carlo State Exploration/Search, Logic, Planning.
	• **Requirements:** Implement efficient heuristic/search algorithms for exploring large state-space.
		Explaining:
		• **Techniques:** Heuristics, Agent (BDI), Algorithms, Math/Planning, Abductive logic
		• **Requirements:**
		    • A language for expressing heuristic rules.
		    • Logic for constraining solutions.

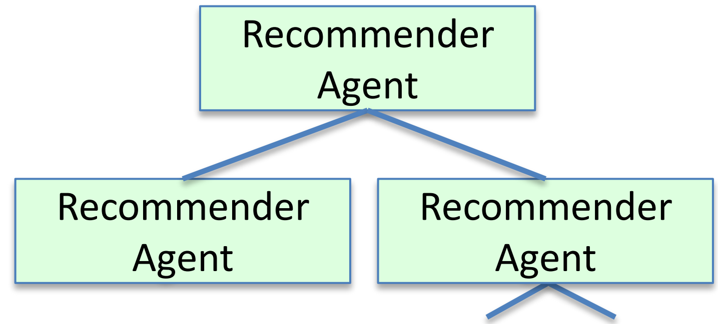
Architecture:



Recommendation Hierarchies



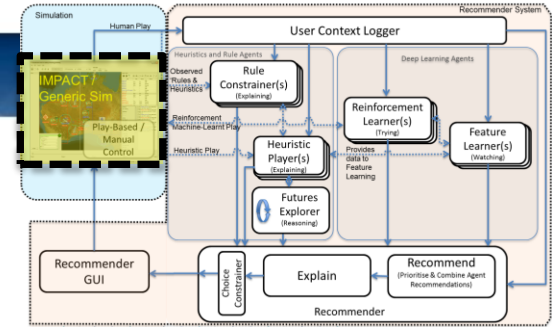
- Recommender Agents:
  - Implemented Recursively:
    - Hierarchy of recommendation agents.



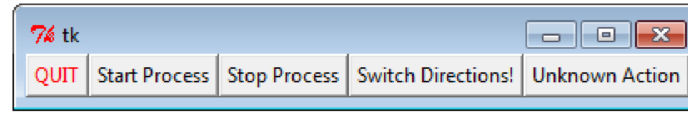
  - Key Concepts:
    - Decomposing / Triggered Recommender Agents - *Elastic* autonomy
    - *Executable* vs *Non-Executable* Recommendations

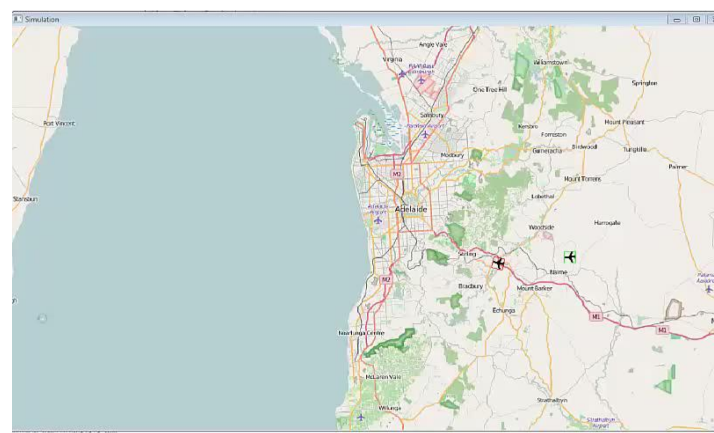
- Simulation:
  - Publishes its own capabilities for accepting/executing recommendations.

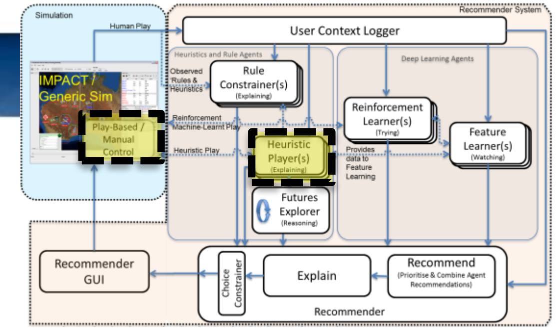
1. Simulation



- Initial UAV Control Simulations
  - Modular/Plug-and-play
    - Ultimately to be replaced by IMPACT
  - Fast
    - In a training mode for rapid learning
  - Machine or Human controllable

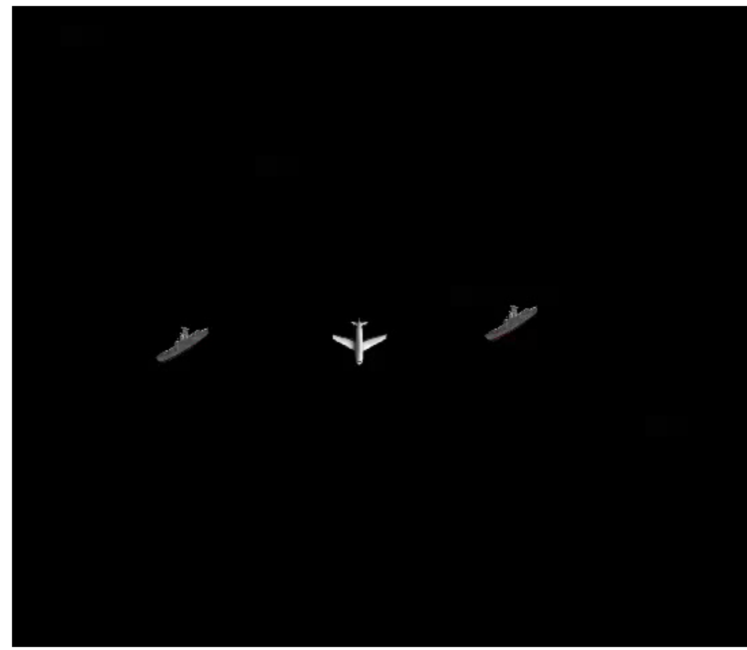




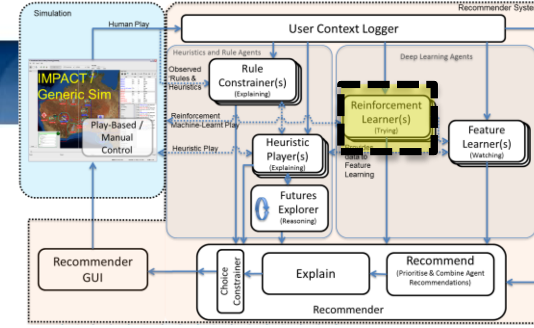


2. First Heuristic Recommenders:

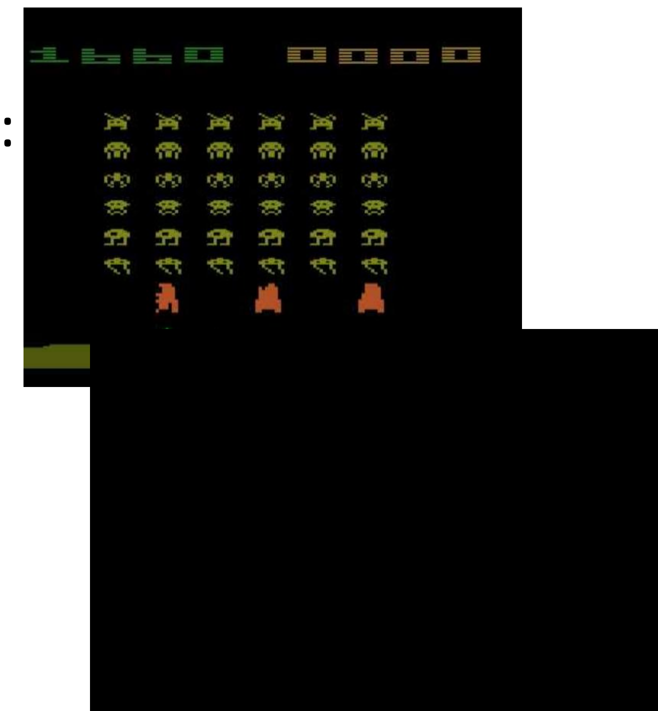
- Initial low-level *Executable* Recommenders
- Heuristics & Play Implementations
  - (a) Search for detection
    - Air-Expanding Square at/on a Point
    - Air Sector Search
    - Air Inspect Point
  - (b) Track Detections



3. First Reinforcement Recommender



- Initial Reinforcement Recommender
  - Deep Q Reinforcement Learning :
    - Combines reinforcement learning with deep neural network.
    - Originally used to play Atari Games – DeepMind/Google
  - Reinforcement Learning Challenges:
    - Credit-assignment problem
    - Explore-exploit dilemma



Deep Q Learning

Credit-Assignment Problem:

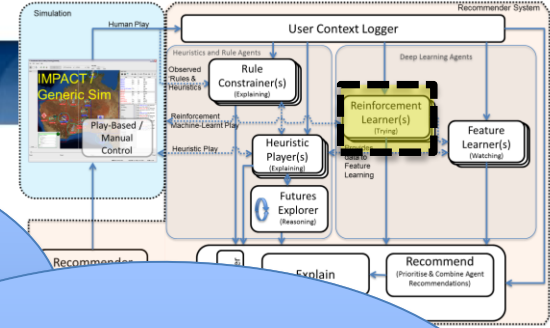
- Q(s, a) represents the maximum discounted future reward when we perform action a in state s (and get to state s' with reward r.)
  - $\mathrm{Q(s, a)} = \mathrm{R(s, a)} + \gamma \sum_{s'} T(s, a, s') \max_{a'} Q(s', a')$ - Bellman Eqn
  - $\mathrm{Q(s, a)} = \mathrm{r} + \gamma \max_{a'} Q(s', a')$
- Iterative Updates (Training):
  - **Prediction**: $Q(s, a)$
  - **Target**: $\mathrm{r} + \gamma \max_{a'} Q(s', a')$
  - $\boldsymbol{L} = \frac{1}{2}\left[\mathrm{r} + \gamma \max_{a'} Q(s', a') - Q(s, a)\right]^2$
- Experience Replay:
  - Store all experiences <s, a, r, s'> in memory.
  - Train on random mini-batches

– *Propagates rewards back in time*.

Deep Q Learning

Explore/Exploit Dilemma

- ε – greedy exploration
  - With probability ε choose a random action, otherwise go with highest Q value action.
  - ε starts at 1.0 (always random) and slowly decreases (eg to 0.1 – mostly policy-based choices)



Our Implementation

Eg: Move Location

Up, Down, Left, Right

...Eg: Change Velocity

Rotate Left, Right
Speed Up, Slo
Down, Do Not

...Eg: Switch Play

Play 1, Play 6, Play 7

wit

Keras i

- Input S e Dat : $(x, y, v$ ... $y, sensor_x$, s
- Output: Actions

– Initial Task:
  - Executable recommendation to track suspect vess
  - Feedback from the environment – Reward/P

Eg:
Reward = -(# Successful Incursions)

Eg:
Correct direction = +ve Reward
Incorrect direction = -ve Reward

Eg:
Reward = -(Distance from Goal)

the ... e given a state

Example 1:

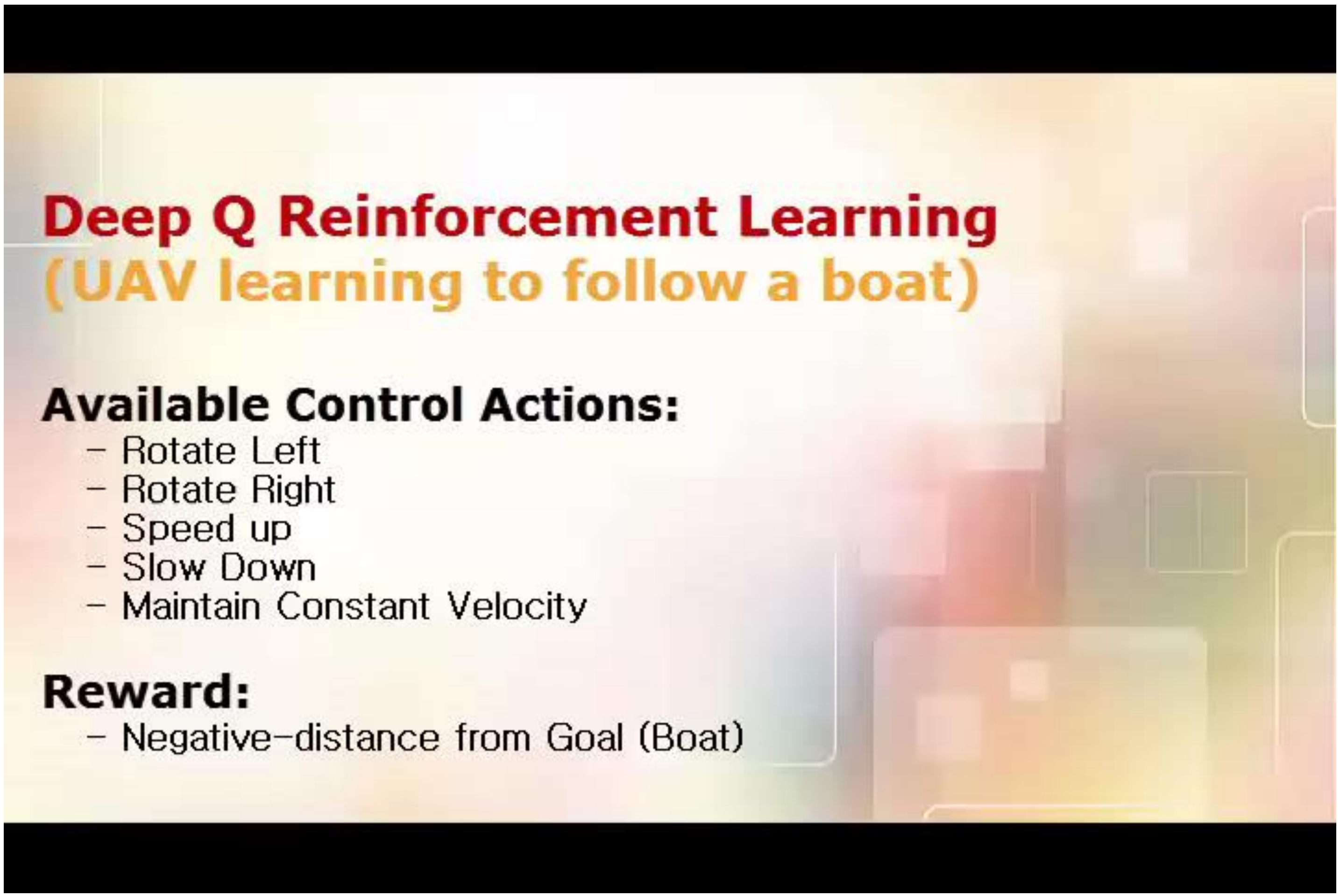
Actions: Up, Down, Left, Right (Constant Velocity, No Pause)

Reward: -(Distance from Goal)

Example 2:

Actions: Rotate Left, Right, Speed up, Slow Down, Do Nothing

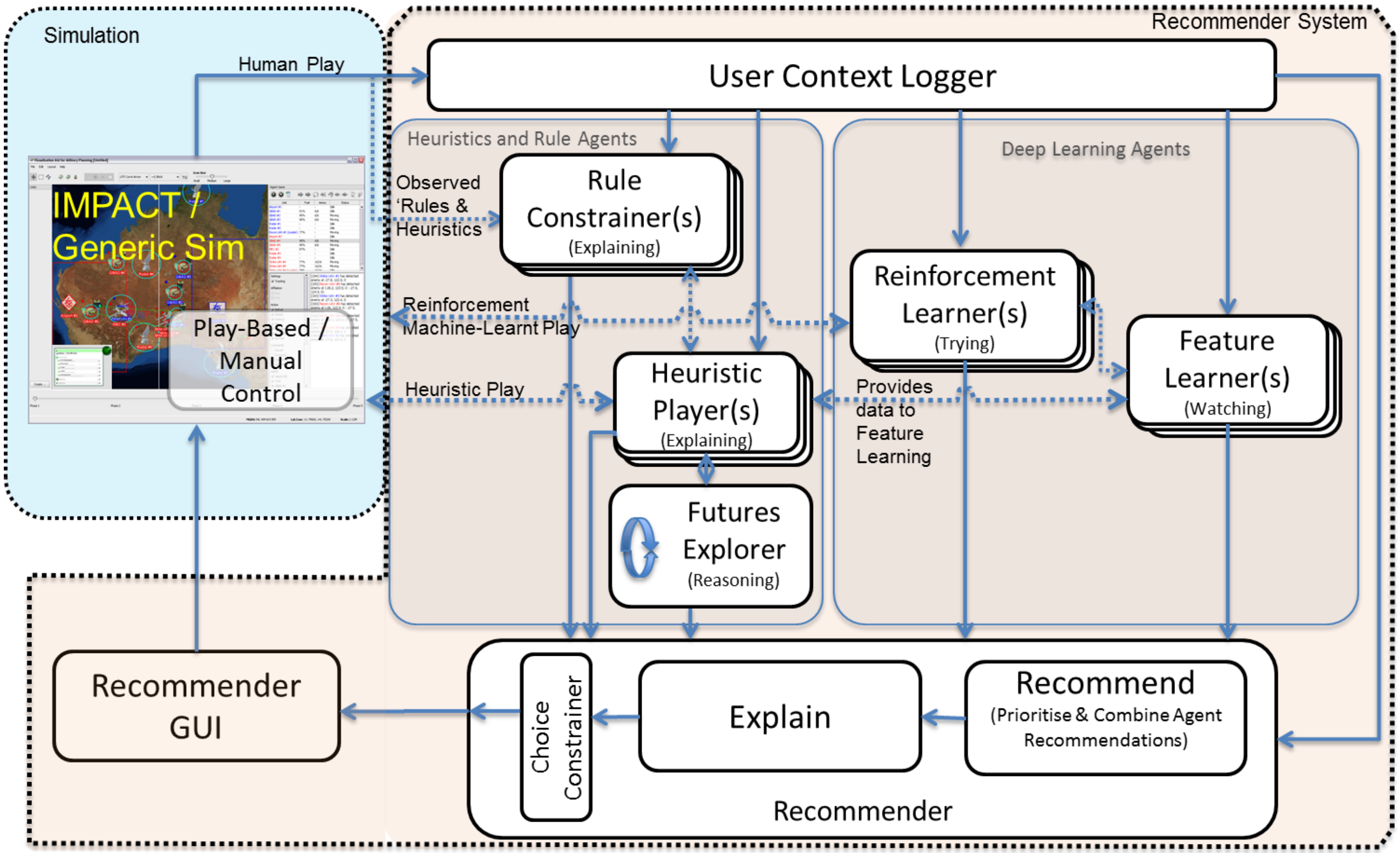
Reward: -(Distance from Goal)



Current/Ongoing R&D:

- **General:**
  - Implement Red "avoid detection" behaviour & blue agent vs red agent play
  - Learn/iterate to higher-level strategies / C2.
- **Deep Q R&D:**
  - Alternative Neural Architectures:
    - Impact on learning rate
    - Scalability with action-complexity
  - Multi-Agent Learning:
    - Single control agent vs multiple learning agents vs hybrid
  - Human-Guided Learning
    - Learning from human interactions as well as self-generated.
- **Other Recommender Agents:**
  - Agent Hierarchies
    - More complex (hierarchical) reward functions and interactions between recommender agents
  - Bayesian Inference for threat heat-maps

Questions


Simulation
Human Play
IMPACT / Generic Sim
Play-Based / Manual Control
Recommender System
User Context Logger
Heuristics and Rule Agents
Deep Learning Agents
Observed 'Rules & Heuristics
Rule Constrainer(s)
(Explaining)
Reinforcement Machine-Learnt Play
Reinforcement Learner(s)
(Trying)
Feature Learner(s)
(Watching)
Heuristic Play
Heuristic Player(s)
(Explaining)
Provides data to Feature Learning
Futures Explorer
(Reasoning)
Recommender GUI
Choice Constrainer
Explain
Recommend
(Prioritise & Combine Agent Recommendations)
Recommender