|  |
| --- |
| Drexel University Graduate Thesis |
| Intent Recognition Engine (IRE) |
| Software Design Document |

|  |
| --- |
| Michael Kozak  3-5-2018 |

Contents

[Common Terms 1](#_Toc508059550)

[Overview 2](#_Toc508059551)

[Purpose of this Document 2](#_Toc508059552)

[Software Requirements 2](#_Toc508059553)

[Top-Level Design 2](#_Toc508059554)

[Second Level Design 2](#_Toc508059555)

[Components 3](#_Toc508059556)

[SCTechTreeManager 3](#_Toc508059557)

[Strategy Space 3](#_Toc508059558)

[Graph Utils 3](#_Toc508059559)

[Terran Tree Manager 3](#_Toc508059560)

[Strategy Reader 3](#_Toc508059561)

# Common Terms

|  |  |
| --- | --- |
| Term | Definition |
| Blueforce | Player or Allied army |
| Redforce | Hostile player |
| Capability | Unit, building, or upgrade |
| Research | Upgrades of unit or building capabilities purchased through an already built building |
| Strategy | A discrete collection of Capabilities in particular ratios designed to gain dominance over an opposing player with a bias towards land, hybrid, or air combat |

# Overview

The Intent Recognition Engine (IRE) is a new component in the NOVA StarCraft AI. IRE enhances NOVA’s current capabilities by reasoning on partial observability of enemy capabilities in order to infer likely strategies. These inferences are then passed to the NOVA strategy manager for preparing effective counter-strategies prior to a full-scale attack.

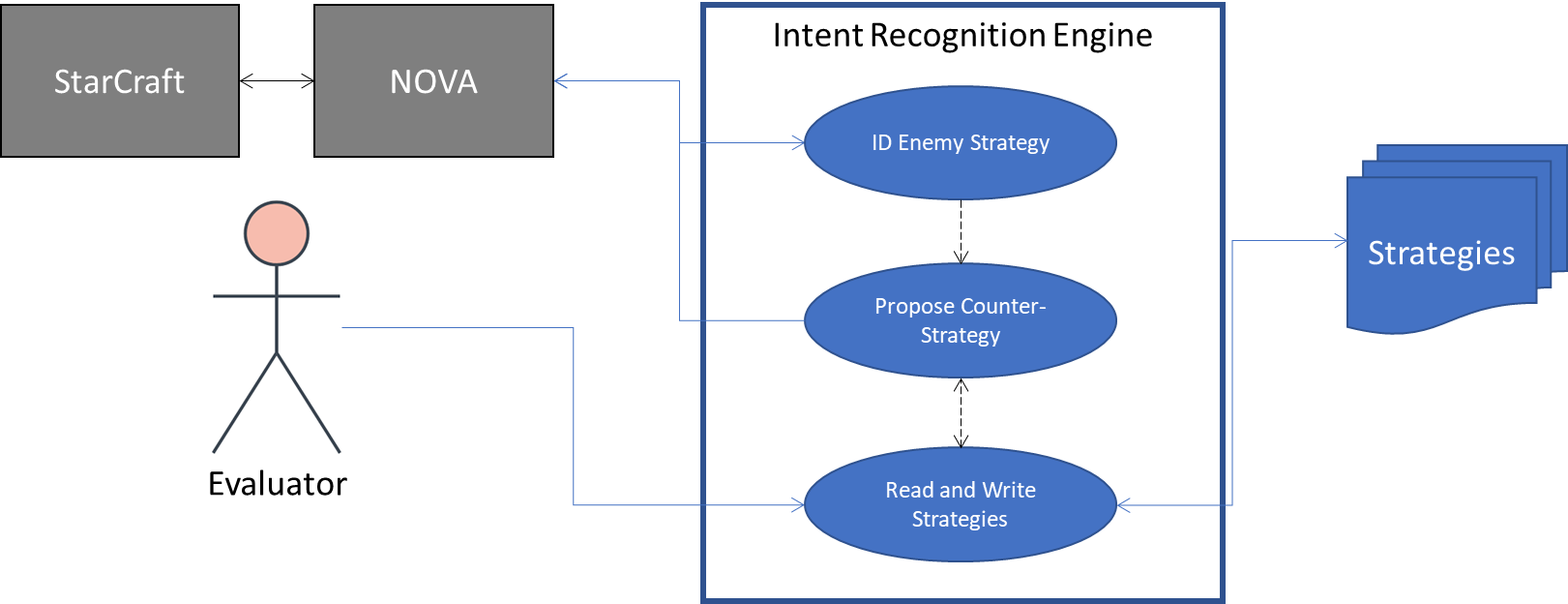
# Purpose of this Document

The purpose of this document is to provide background information as to the design of the system. Specifically, design decisions made within IRE in laying out the architecture, including constraints, assumptions, and reasoning behind choices made. Each component in the system is then outlined to detail its purpose, functional ins and outs, and its relation to the other components.

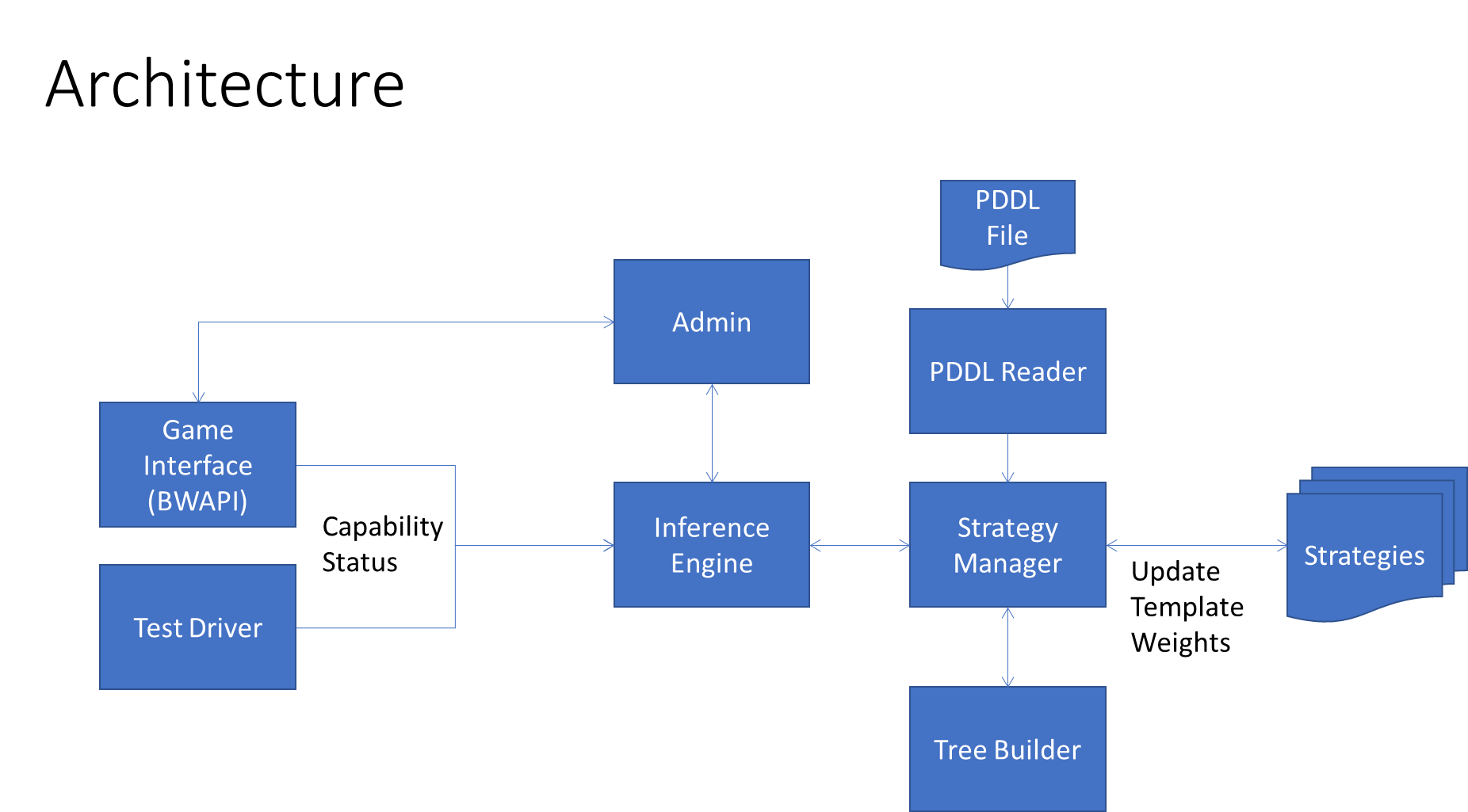
# Software Requirements

* StarCraft Brood War v1161
* Nova Master GIT pull
* BWAPI v4.20
* BWTA v2.2
* Visual Studio 2017 (v141)
* Boost v1.66.0
* Apr
* Apr-util
* Geometry-develop
* Log4cxx for Visual Studios 2010 (v100)

# Use Case Design



# Component Design



# Classes

## SCTechTreeManager

**Purpose:** External Main for testing and evaluating IRE outside of NOVA

**Inputs:** N/A

**Outputs:** DOT files for each races’ tech tree as well as DOT files for the observed enemy race tech tree once all observations have been made

**Interacts With:** TerranTreeManager, GraphUtils

## Strategy Space

**Purpose:** Represents an N-dimensional graph in which all enemy strategies are stored. The axes of this graph each represent a single feature of a strategy such as air vs anti-air or agressiveness versus defensiveness. When an enemy unit or building is detected, the "strategy space" is informed and it increases the "strength" of strategies that rely on this unit. Over time, the strategies most likely to be engaged by the enemy will be the "strongest" and IRE can use that information to build a counter-strategy.

**Inputs:** Strategies, Observations (UnitType)

**Outputs:** DOT files for each races’ strategy space, identified strategy of enemy race, recommendations for countering ID’d strategy

**Interacts With:** TerranTreeManager

## Graph Utils

**Purpose:** Collection of structs, typedefs, and static utility functions for representing the Starcraft Tech Trees and the Strategy Graphs

**Inputs:** N/A

**Outputs:** N/A

**Interacts With:** All

## Terran Tree Manager

**Purpose:** Represent the full tech tree for the Terran Race.

**Inputs:** Observations (UnitType)

**Outputs:** SCGraph where each node has a strength relative to the number of times observed

**Interacts With:** StrategySpace, StrategyReader, GraphUtils, SCTechTreeManager

## Strategy Reader

**Purpose:** Handles reading the strategies from file and storing them for future access during runtime.

**Inputs:** N/A

**Outputs:** List of strategies for each race

**Interacts With:** GraphUtils, SCTechTreeManager

# Strategies

Although there are near infinite numbers of potential combinations of units, buildings, and research upgrades, strategies can be generally identified as specific paths taken down the full tech tree. These strategies vary from race to race based on their unique tech trees. As part of NOVA research, the following strategies have been identified based on past competitions. For each table, the strategies are enumerated by name. Where the high-level purpose of the strategy is unclear, it is detailed in brackets.

For each strategy, the intensity among each axis in the Strategy Space is listed. These intensities are based on IRE developer input using past experience with StarCraft as a basis for weights. The intent is that for future development these intensities can be learned, but for initial proof-of-concept these fixed numbers are considered sufficient for testing and evaluating IRE.

## Terran

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **A vs AA** | **G vs AG** | **Atk vs Def** |
| Bio (marines/medics) | -0.25 | 0.75 | 1 |
| Rax\_fe (second base) | 0 | 0 | -1 |
| Two\_facto(tanks) | 0 | 1 | 0.5 |
| Vultures | 0 | 1 | 0.75 |
| Air(wraiths) | 0.75 | -0.5 | 0.25 |
| drop | 0.25 | 0.25 | -0.25 |

## Protoss

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **A vs AA** | **G vs AG** | **Atk vs Def** |
| Two\_gates (zealots) | 0 | 1 | 1 |
| fast\_dt (dark templars) | -0.25 | 0 | 0 |
| Templar | -0.5 | 0 | 0.5 |
| Speedzeal (zealots+upgrades) | 0 | 1 | 0.75 |
| Corsair | -1 | 0 | -0.75 |
| Nony (dragoons) | 0 | 0 | 0.5 |
| Reaver\_drop | 0.25 | 0 | -0.25 |
| Two\_gates (zealots) | 0 | 1 | 1 |

## Zerg

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **A vs AA** | **G vs AG** | **Atk vs Def** |
| Speedlings (zerlings) | 0 | 1 | 1 |
| Fast\_mutas (mutalisks) | 1 | -1 | 0.25 |
| Mutas (expand+mutas) | 0.75 | -0.75 | -0.25 |
| Lurkers | 0 | 0 | -0.25 |
| Hydras | -0.25 | 1 | 0.75 |