Математическая модель:

Q - множество состояний

A – множество действий

I ( Q) - начальные условия.

q = {<pi, vi>} Q – состояние, описываемое через множество параметров (конечное) pi и соответствующих им значений vi, где vi может принимать числовые значения, булевы и массивы/кортежи из игровых элементов.

a = <pr, ef> A – действие, описываемое через предусловия pr = {<pi, vi>} ( q), которым должно удовлетворять текущее состояние q для выполнения действия, и эффект ef = {<pi, ki>}( q), производимый на состояние при выполнении, где ki - показатель того, как изменится параметр pi состояния.

F: QxA => Е = {ei} ( A) – функция поиска набора доступных действий Е для заданного состояния q.

С: QxAхt=>c – функция определения стоимости действия в заданном состоянии, где t – время, пройденное с начала игры, с – вычисленная стоимость.

h: QxG=>c – эвристическая функция определения стоимости достижения цели g G из состояния q Q.

ГМ: QxA=>Q функция перехода: для каждого состояния q, принадлежащего Q, и действия a, принадлежащего A, определяет следующее состояние q'= ГМ(q,a).

g = <cond, priority> (priorityi ≠ priorityj при i ≠ j) - цель, описываемая через cond – условия, которые должны быть достигнуты, и priority – приоритет цели.

Требуется найти план для цели gi – упорядоченное множество действий P={a1, ..., an}, такое что суперпозиция функций перехода ГМ(ГМ (…ГМ (ГМ (q0, a1) a2)…, an-1), an) удовлетворяет условиям condi цели при q0, принадлежащего I.

TC (Task Chain) = {gi} – набор целей с приоритетом.

Planner = <A, F, C, h, Гм, QxG=>P> - планировщик, способный находить все возможные действия для заданного состояния, оценивать их стоимость и создавать план P, исходя из текущего и целевого состояний.

M – набор менеджеров для выполнения плана P.

S – набор сенсоров для обновления информации о состоянии.

B = <P, {ti}> (|{ti}| = |P|) – доска объявлений, где P – текущий план, {ti} – множество меток к каждому действию ai из P. ti = (empty/success/fail/inProcess), где empty означает, что i-е действие ещё не рассматривалось менеджерами, success – успешно выполнено, fail – не может быть выполнено, inProcess – в процессе выполнения.

Модель функционирует следующим образом:

1. Текущее состояние q хранит набор параметров и их значений об агенте и среде и обновляется с помощью сенсоров S, которые считывают информацию из среды.

2. Каждый сенсор реагирует на определённое изменение среды и отвечает за обновление определённых параметров состояния (сенсор, отвечающий за постройки, обновляет информацию при строительстве или разрушении зданий; отвечающий за ресурсы – об их количествах и темпах добычи и т.д.).

3. Цепь задач (Task chain – TC) хранит множество целей {gi} с разным приоритетом priorityi, выбирая в качестве текущей цели ту, условия condi которой не удовлетворяют текущему состоянию q, и которая имеет наивысший приоритет. Выбранная задача gi передаётся планировщику Planner.

4. Планировщик, получив цель gi, начинает строить план P:

1) С помощью функции F из множества действий А находятся все действия {ei}, которые могут быть выполнены в текущем состоянии.

2) С помощью функций С и h для каждого действия вычисляется его стоимость, а с помощью функции Гм – переход из одного состояния в другое.

3) Строится дерево состояний с переходами в виде действий с разной стоимостью, пока с помощью алгоритма А\* не будет найден такой план Р ={a1, ..., an} с минимальной суммарной стоимостью действий, что суперпозиция функций перехода ГМ(ГМ (…ГМ (ГМ (q0, a1) a2)…, an-1), an) равна состоянию qn, удовлетворяющему условиям condi цели gi, а q0 – текущее состояние.

4) Полученный план Р передаётся доске объявлений B.

5. Доска объявлений В хранит план Р и множество меток {ti}, каждая из которых содержит информацию о действии аi из Р. Действие может быть выполнено, провалено, не рассмотрено или находиться в процессе выполнения. Доска объявлений сообщает планировщику об изменении каждой из меток.

6. Менеджеры М считывают план Р с доски объявлений В и выполняют его, воздействуя на среду, оставляя метки {ti} на доске объявлений.