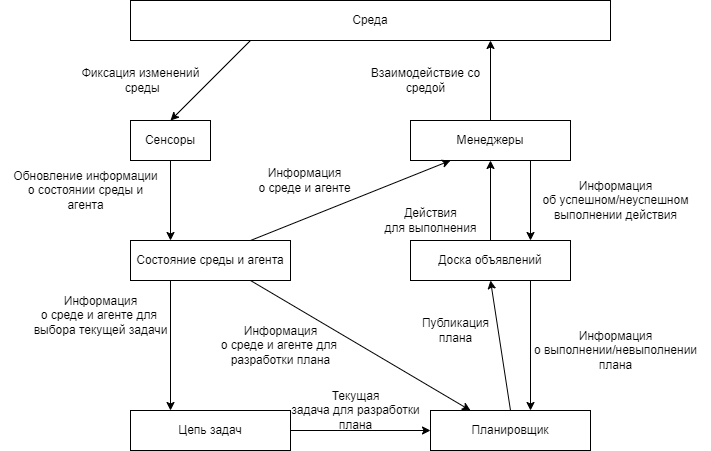
**Архитектура**



1. Сенсоры (S). Считывают информацию об изменении среды и обновляют её.

2. Состояние среды и агента (q Q). Данные вида «параметр: значение». Разделяемая память с одним «писателем» (сенсоры) и тремя «читателями» (менеджеры, цепь задач, планировщик).

3. Цепь задач (TC). На основе данных о среде и агенте q Q выбирает текущую цель g и передаёт планировщику Planner.

4. Планировщик (Planner). На основе полученной цели g и информации о состоянии среды и агента q Q формирует план P и отправляет на доску объявлений B, прослушивая информацию об успешности выполнения плана на основе меток {ti}.

5. Доска объявлений (B). Хранит текущий план P, передаёт его менеджерам M, прослушивая и передавая планировщику информацию об успешности выполнения плана на основе меток {ti}.

6. Менеджеры (M). На основе плана и состояния среды и агента воздействуют на среду. Каждому действию ai из плана P на доске B ставят метку ti.

Плюсы:

1. Поддержка принятия решений на стратегическом уровне. Цепь задач в будущем можно будет расширить до иерархической сети задач, цели из которой будут выбираться исходя из большего количества факторов. По сути, для принятия стратегических решений можно будет реализовать отдельный планировщик.

2. Данная модель обучаема как на тактическом уровне (на уровне менеджеров), так и на стратегическом (на уровне планирования).

3. Расширяемость. Для расширения функционала достаточно добавить новые действия или цели, не описывая новые условия и функции для переходов в каждом состоянии (как при использовании конечных автоматов, например).

Минусы:

1. Так как модель использует метод поиска в пространстве состояний, на поздних этапах игры количество доступных действий в каждом состоянии может возрасти настолько, что планировщик не будет успевать формировать актуальный план. Поэтому система вряд ли сможет использовать весь спектр возможностей StarCraft, или будет требоваться ограничение глубины поиска.