



```

virtual void TimeStep&#40; const TIoDataContainer& receptors,
TIoDataContainer& effectors &#41;;

private:
    const TIoLimitsContainer receptorsLimits;
    const TIoLimitsContainer effectorsLimits;

    typedef std::vector<double> TDistributionType;           // Тип
для статистических распределений
    typedef std::vector<TDistributionType> TDistributionsContainer;
//    Контейнер из статистических распределений
//    Кластер статистической модели мира
    struct CCluster &#123;
        CCluster&#40; int receptorsCount, int effectorsCount &#41;;

        TDistributionsContainer ReceptorsStatistics;
        TDistributionsContainer EffectorsStatistics;
        TDistributionType ContextStatistics;
        TDistributionType::value_type Weight;
        TDistributionType::value_type Expectation;
    &#125;;

    typedef std::vector<CCluster> TClustersContainer;
    TClustersContainer worldModel;
    TDistributionType context;
    TIoDataContainer prevStepEffectors;

    //    Описание кандидата нового состояния эффектора
    struct CEffectorVariant &#123;
        CEffectorVariant&#40; TDistributionType::value_type
likelihood, int testedStateIndex,
        TIoDataContainer::value_type variant &#41;;

        const TDistributionType::value_type Likelihood;
        const int TestedStateIndex;
        const TIoDataContainer::value_type Variant;
    &#125;;

    //    Компаратор кандидатов
    struct CEffectorVariantComparer : public
std::binary_function<CEffectorVariant, CEffectorVariant, bool> &#123;
        bool operator&#40;&#41;&#40; const CEffectorVariant& left,
const CEffectorVariant& right &#41; const;
    &#125;;

    //    Массив приоритетных очередей
    typedef std::vector< std::priority_queue<CEffectorVariant,
std::vector<CEffectorVariant>,
        CEffectorVariantComparer> > TVariantQueuesContainer;

    //    Описание протестированного состояния эффектора

```

```
typedef std::vector< std::vector<bool> > TVariantsContainer;
class CEffectorsTestedState &#123;
public:
    CEffectorsTestedState&#40; const TIoDataContainer& effectors
&#41;;
    //          Породить новую комбинацию
    CEffectorsTestedState&#40; CEffectorsTestedState& baseState,
        int effectorIndex, TIoDataContainer::value_type
newVariant &#41;;
    const TIoDataContainer& Effectors&#40;&#41; const &#123;
return effectors; &#125;
    const TVariantsContainer& TestedVariants&#40;&#41; const
&#123; return testedVariants; &#125;

private:
    TVariantsContainer testedVariants;          //
протестированные комбинации
    TIoDataContainer effectors;          //          состояние
эффекторов
&#125;;
typedef std::vector<CEffectorsTestedState> TTestedStatesContainer;

void calculateNextStepContext&#40; const TIoDataContainer& receptors,
TDistributionType& adArchive &#41; const;
void updateStatistics&#40; const TIoDataContainer& receptors, const
TDistributionType& adArchive &#41;;
void findBestReaction&#40; const TDistributionType& adArchive,
TIoDataContainer& effectors &#41; const;
&#125;;
```

From:  
<http://ailab.ru/wiki/> - AI Lab Wiki

Permanent link:  
[http://ailab.ru/wiki/doku.php?id=%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8:mumla:%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5\\_%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5\\_%D0%B4%D0%BB%D1%8F\\_%D0%B0%D0%B3%D0%B5%D0%BD%D1%82%D0%B0\\_v0.1](http://ailab.ru/wiki/doku.php?id=%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8:mumla:%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5_%D0%B4%D0%BB%D1%8F_%D0%B0%D0%B3%D0%B5%D0%BD%D1%82%D0%B0_v0.1)

Last update: 2021/04/07 21:33