SDM'Studio

ANR PLASMA

Jilles-Steeve Dibangoye, David Albert

TABLE OF CONTENTS

1 **GET STARTED**

- Installation
- Basic Usage

2 **DATA STRUCTURES**

- Worlds
- Generic States
- Generic Actions
- Value Functions

3 **ALGORITHMS**

- HSVI

4 **CONTRIBUTE**

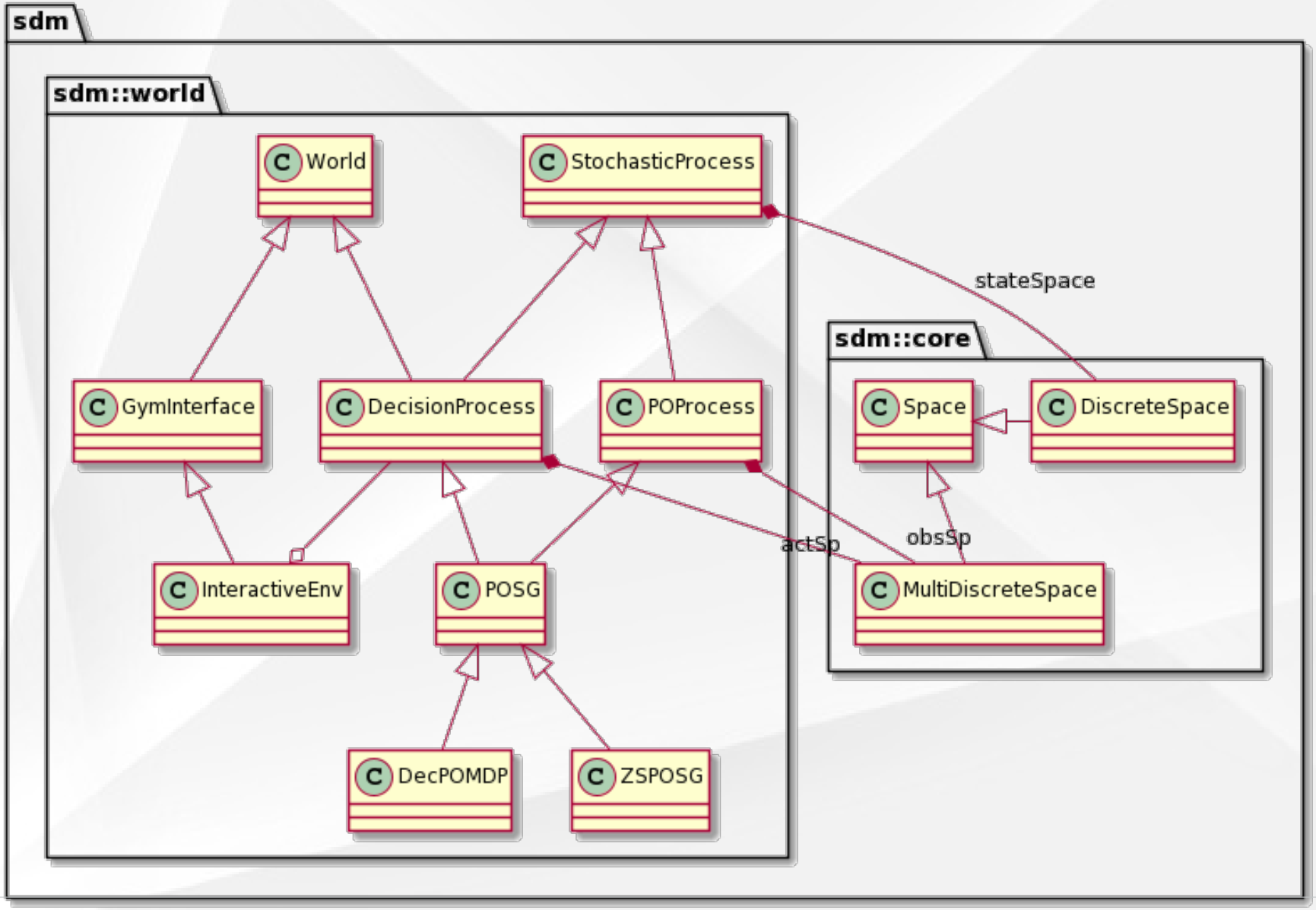
- Contribute to SDMS

© Jilles-Steeve Dibangoye, David Albert

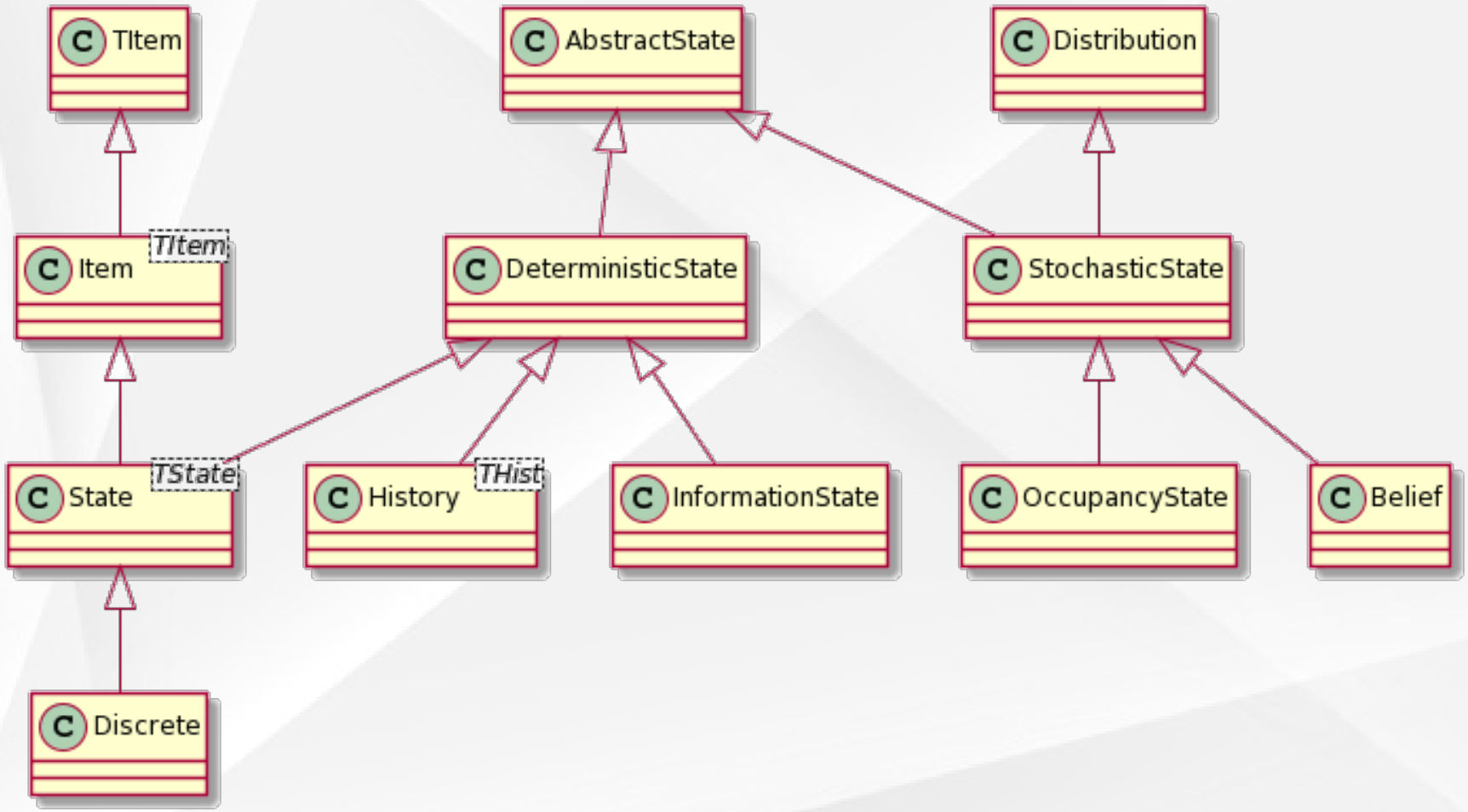
Installation

Basic Usage

Worlds - Class Diagram



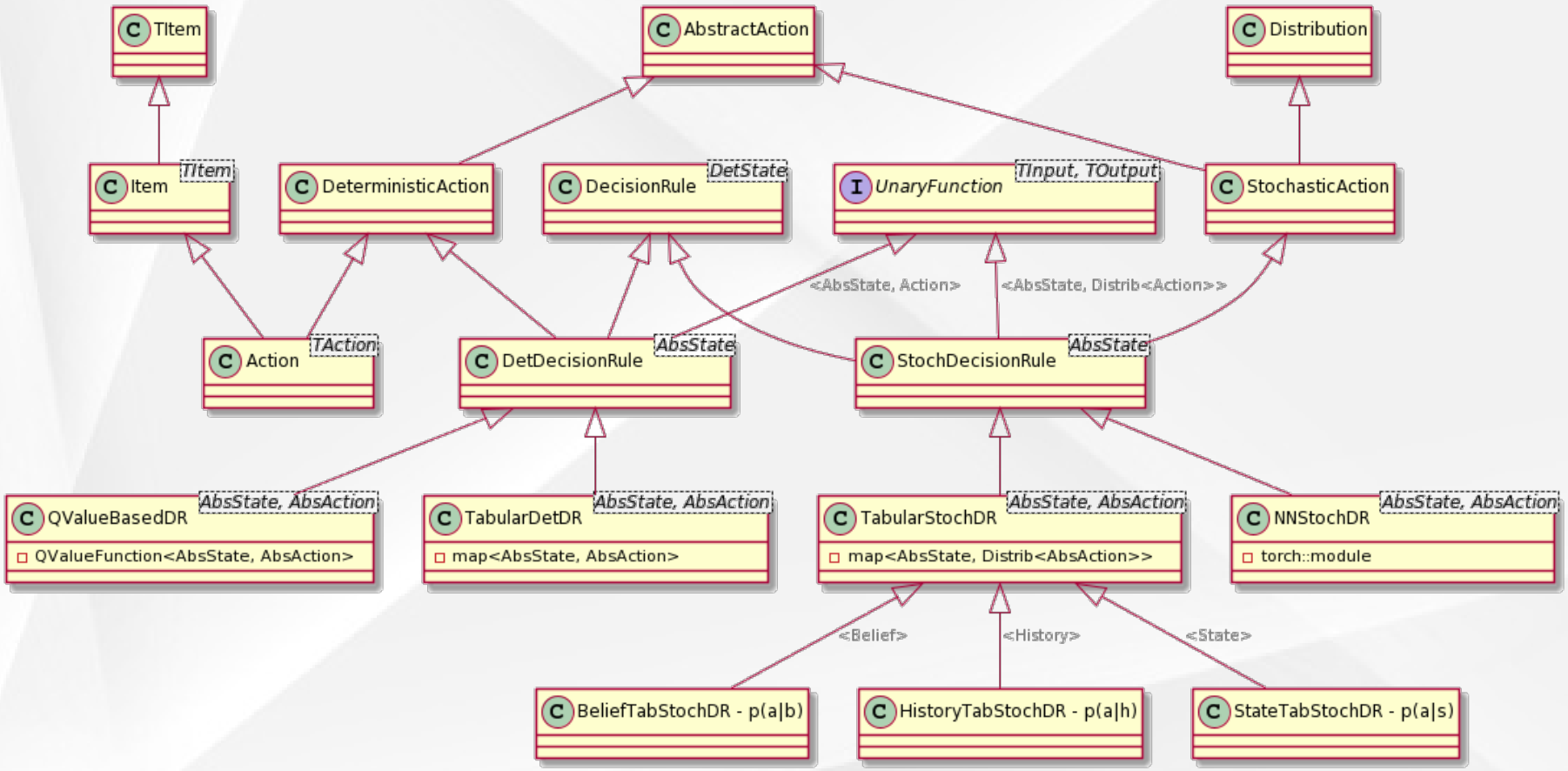
Generic States - Class Diagram



- *AbstractState* $\Leftrightarrow$ *GenericState*
  - *State* $\Leftrightarrow s_t$
  - *History* $\Leftrightarrow$ *GenericHistory*
    - *ObsHistory* $\Leftrightarrow h_t^{indiv} = (z_1, z_2, \ldots, z_t)$
    - *ActObsHistory* $\Leftrightarrow h_t^{indiv} = (a_0, z_1, a_1, \ldots, a_{t-1}, z_t)$
    - *JointObsHistory* $\Leftrightarrow h_t^{joint} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_t)$
    - *JointActObsHistory* $\Leftrightarrow h_t^{joint} = (\mathbf{a}_0, \mathbf{z}_1, \mathbf{a}_1, \ldots, \mathbf{a}_{t-1}, \mathbf{z}_t)$
  - *InformationState* $\Leftrightarrow \iota_t = (b_0, d_1, ..., d_t)$
  - *Belief* $\Leftrightarrow b_t = p\left(s_t \mid h_t^{indiv}\right)$
  - *OccupancyState* $\Leftrightarrow \xi_t = p\left(s_t, h_t^{joint} \mid \iota_t\right)$

© Jilles-Steeve Dibangoye, David Albert

Generic Actions - Class Diagram

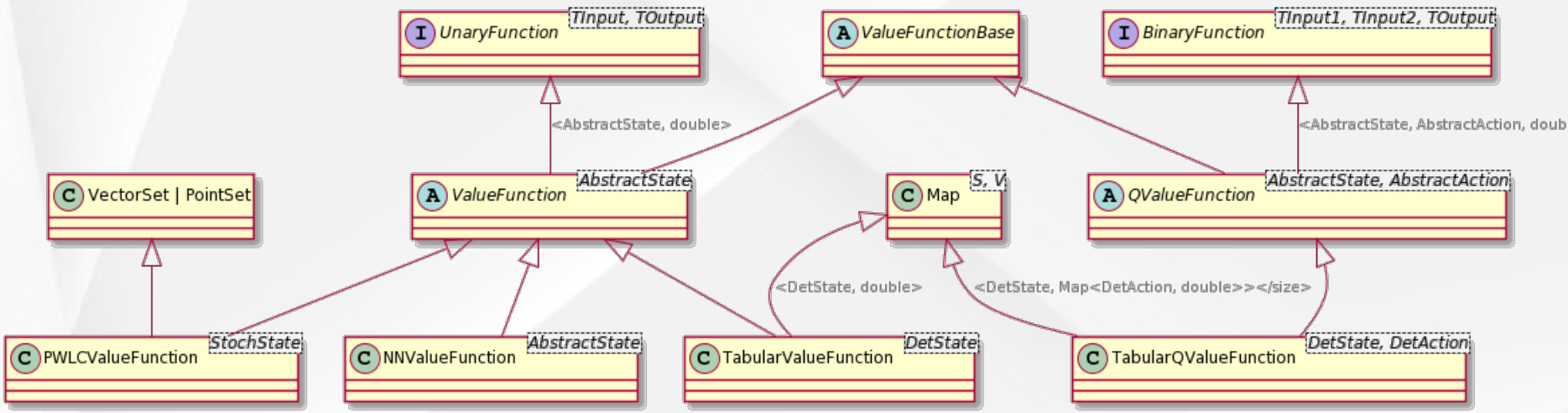


© Jilles-Steeve Dibangoye, David Albert

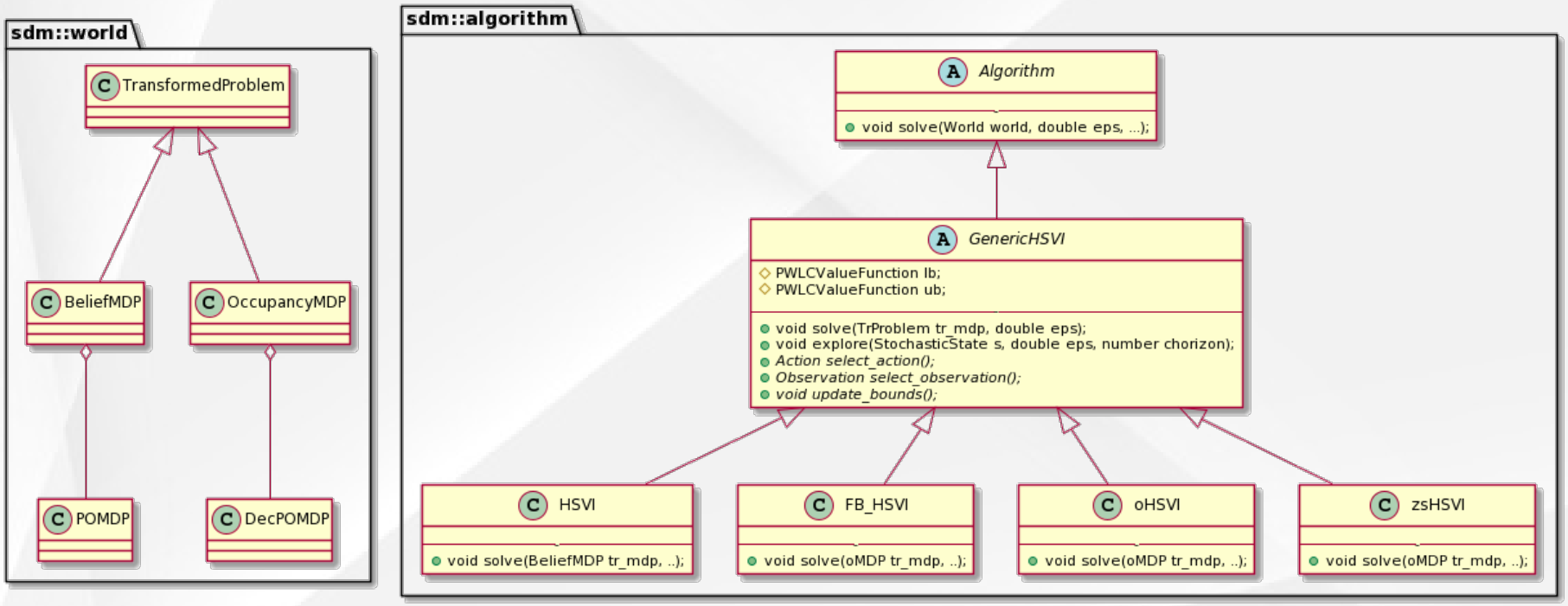
Generic Actions - Class/Theory equivalence

- *AbstractAction* $\Leftrightarrow$ *GenericAction*
  - *Action* $\Leftrightarrow a_t$
  - *DecisionRule* $\Leftrightarrow$ *GenericDecisionRule*
    - *StateBasedDR* $\Leftrightarrow p(a \mid s)$
    - *BeliefBasedDR* $\Leftrightarrow p(a \mid b)$
    - *HistoryBasedDR* $\Leftrightarrow p(a \mid h)$

Value Functions - Class Diagram



HSVI


sdm::world
TransformedProblem
BeliefMDP
OccupancyMDP
POMDP
DecPOMDP
sdm::algorithm
Algorithm
void solve(World world, double eps, ...);
GenericHSVI
PWLCValueFunction lb;
PWLCValueFunction ub;
void solve(TrProblem tr_mdp, double eps);
void explore(StochasticState s, double eps, number chorizon);
Action select_action();
Observation select_observation();
void update_bounds();
HSVI
void solve(BeliefMDP tr_mdp, ..);
FB_HSVI
void solve(oMDP tr_mdp, ..);
oHSVI
void solve(oMDP tr_mdp, ..);
zsHSVI
void solve(oMDP tr_mdp, ..);

Contribute to SDMS