# Adding a new application to the MCTS package

In order to implement the MCTS code for a particular problem, we can create a derived class from the bass class "Model" that contains the following members:

- **label**: a string that represents the problem, e.g. 'pricing', 'selling', 'TSP'.
- **state**: a list that contains all the information in the state variable to make the system Markovian. For example, for a selling problem, the state is a two-dimensional list that contains the current inventory level and price.
- **sigma_w**: parameter(s) of the model, such as the standard deviation of a normal distribution.
- **exo_info**: exogenous information of the model that is used in the transition function and the reward function.
- **decision_range**: a list that represents the feasible set of decisions. It can be updated using the state information.

Also, we define the following member functions in the "Model" class:

- **__init__(self, s0, sigma_w)**: a constructor that initializes a "Model" object using the state s0 and parameter sigma_w.
- **__copy__(self)**: a copy constructor that makes a deep copy of an object.
- **exogenous_gen(self, x_t)**: this function generates exogenous information based on the state information (member "state"), the decision x_t, and model parameters (member "sigma_w").
- **transition(self, x_t)**: the transition function updates the state (member "state") and the feasible set of decisions (member "decision_range") based on the decision x_t, the current state (member "state") and the exogenous information (member "exo_info").
- **forward_one_step(self, x_t)**: this function takes a decision x_t, calls exogenous_gen(x_t) to generate the exogenous information, and makes a one-step update by calling transition(x_t).
- **reward(self, x_t)**: the reward function that returns a scalar reward based on a decision x_t and the state information (member "state"). If the reward is stochastic, the exogenous information (member "exo_info") is also used to compute the reward.