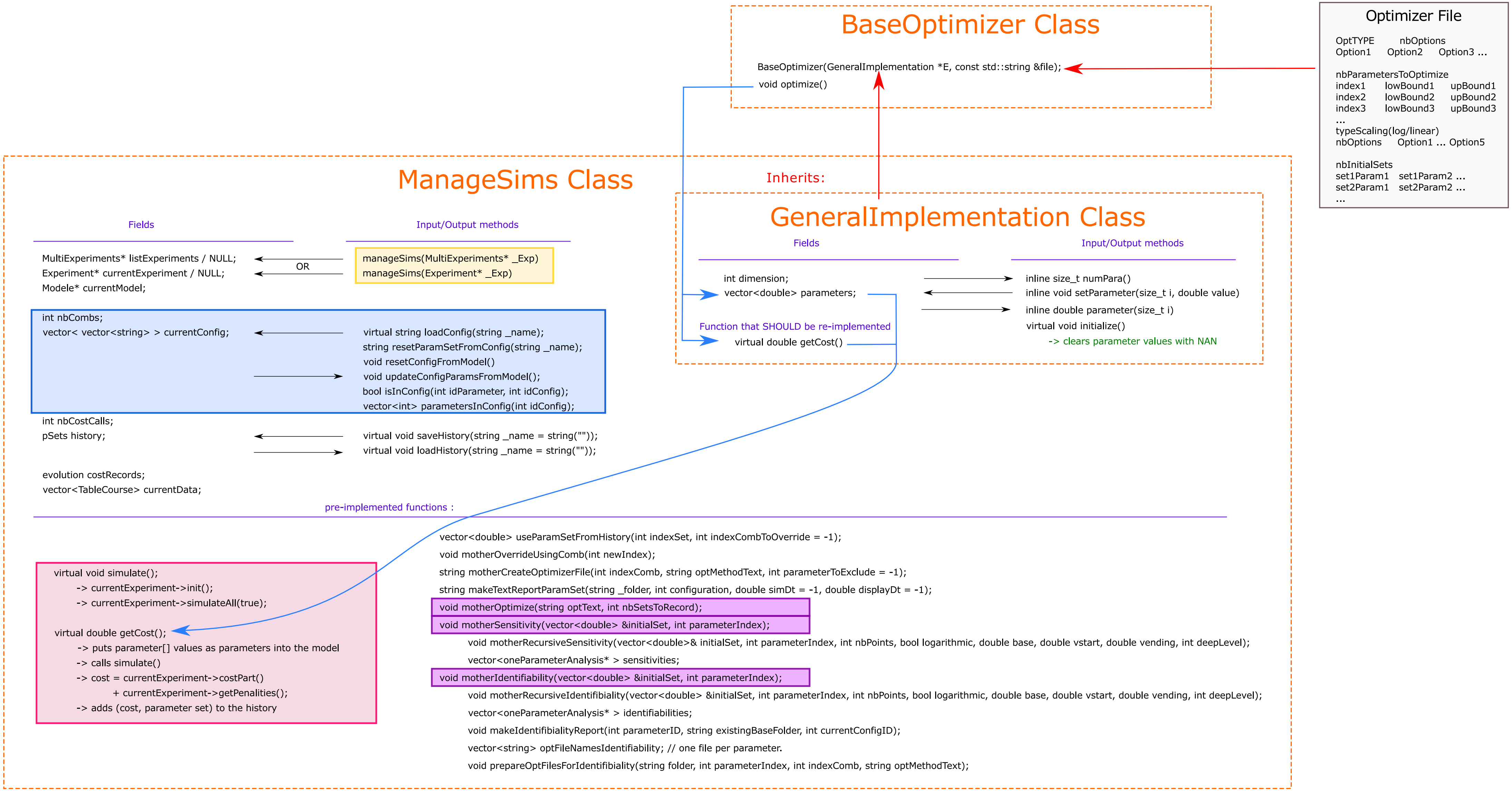

BaseOptimizer Class

BaseOptimizer(GeneralImplementation *E, const std::string &file);

void optimize()

Optimizer File

OptTYPE nbOptions
Option1 Option2 Option3 ...

nbParametersToOptimize
index1 lowBound1 upBound1
index2 lowBound2 upBound2
index3 lowBound3 upBound3
...
typeScaling(log/linear)
nbOptions Option1 ... Option5

nbInitialSets
set1Param1 set1Param2 ...
set2Param1 set2Param2 ...
...

ManageSims Class

Inherits:

GeneralImplementation Class

Fields

int dimension;
vector<double> parameters;

Function that SHOULD be re-implemented

virtual double getCost()

Input/Output methods

inline size_t numPara()
inline void setParameter(size_t i, double value)
inline double parameter(size_t i)
virtual void initialize()
-> clears parameter values with NAN

ManageSims Class

Fields

MultiExperiments* listExperiments / NULL;
Experiment* currentExperiment / NULL;
Modele* currentModel;

OR

Input/Output methods

manageSims(MultiExperiments* _Exp)
manageSims(Experiment* _Exp)

int nbCombs;
vector< vector<string> > currentConfig;

virtual string loadConfig(string _name);
string resetParamSetFromConfig(string _name);
void resetConfigFromModel()
void updateConfigParamsFromModel();
bool isInConfig(int idParameter, int idConfig);
vector<int> parametersInConfig(int idConfig);

int nbCostCalls;
pSets history;

virtual void saveHistory(string _name = string(""));
virtual void loadHistory(string _name = string(""));

evolution costRecords;
vector<TableCourse> currentData;

pre-implemented functions :

virtual void simulate();
-> currentExperiment->init();
-> currentExperiment->simulateAll(true);

virtual double getCost();
-> puts parameter[] values as parameters into the model
-> calls simulate()
-> cost = currentExperiment->costPart()
+ currentExperiment->getPenalities();
-> adds (cost, parameter set) to the history

vector<double> useParamSetFromHistory(int indexSet, int indexCombToOverride = -1);
void motherOverrideUsingComb(int newIndex);
string motherCreateOptimizerFile(int indexComb, string optMethodText, int parameterToExclude = -1);
string makeTextReportParamSet(string _folder, int configuration, double simDt = -1, double displayDt = -1);
void motherOptimize(string optText, int nbSetsToRecord);
void motherSensitivity(vector<double> &initialSet, int parameterIndex);
void motherRecursiveSensitivity(vector<double>& initialSet, int parameterIndex, int nbPoints, bool logarithmic, double base, double vstart, double vending, int deepLevel);
vector<oneParameterAnalysis* > sensitivities;
void motherIdentifiability(vector<double> &initialSet, int parameterIndex);
void motherRecursiveIdentifiability(vector<double> &initialSet, int parameterIndex, int nbPoints, bool logarithmic, double base, double vstart, double vending, int deepLevel);
vector<oneParameterAnalysis* > identifiabilities;
void makeIdentifiabilityReport(int parameterID, string existingBaseFolder, int currentConfigID);
vector<string> optFileNamesIdentifiability; // one file per parameter.
void prepareOptFilesForIdentifiability(string folder, int parameterIndex, int indexComb, string optMethodText);