HeuristicLab

A Paradigm-Independent and Extensible Environment for Heuristic Optimization

Programming HeuristicLab

Algorithms and Problems

A. Scheibenpflug
Heuristic and Evolutionary Algorithms Laboratory (HEAL)
School of Informatics/Communications/Media, Campus Hagenberg
University of Applied Sciences Upper Austria

HEAL
Heuristic and Evolutionary Algorithms Laboratory

Josef Ressel-Zentrum
HEUREKA!

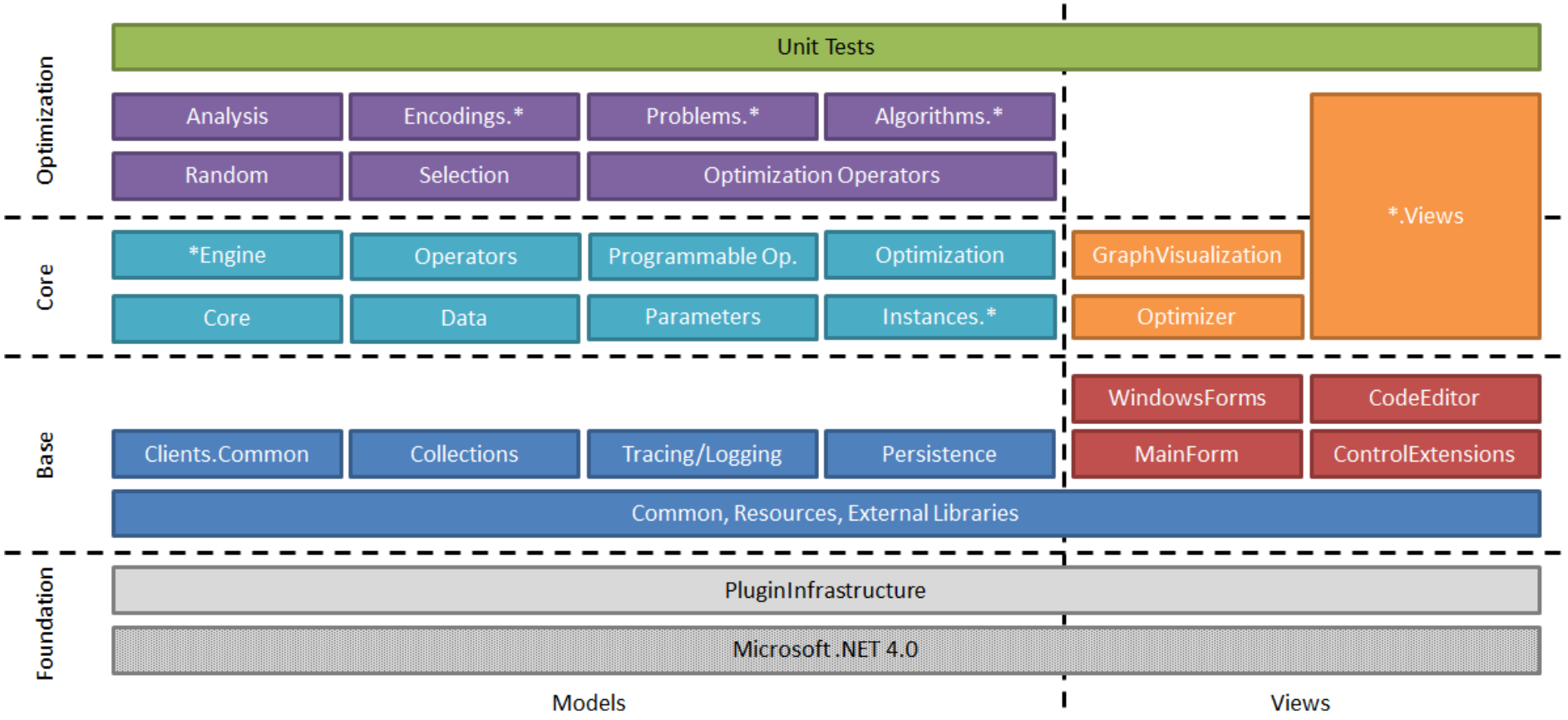
Overview

HeuristicLab

- HL Algorithm Model
- Parameters, Operators and Scopes
- Algorithms
- Problems

Where are we?

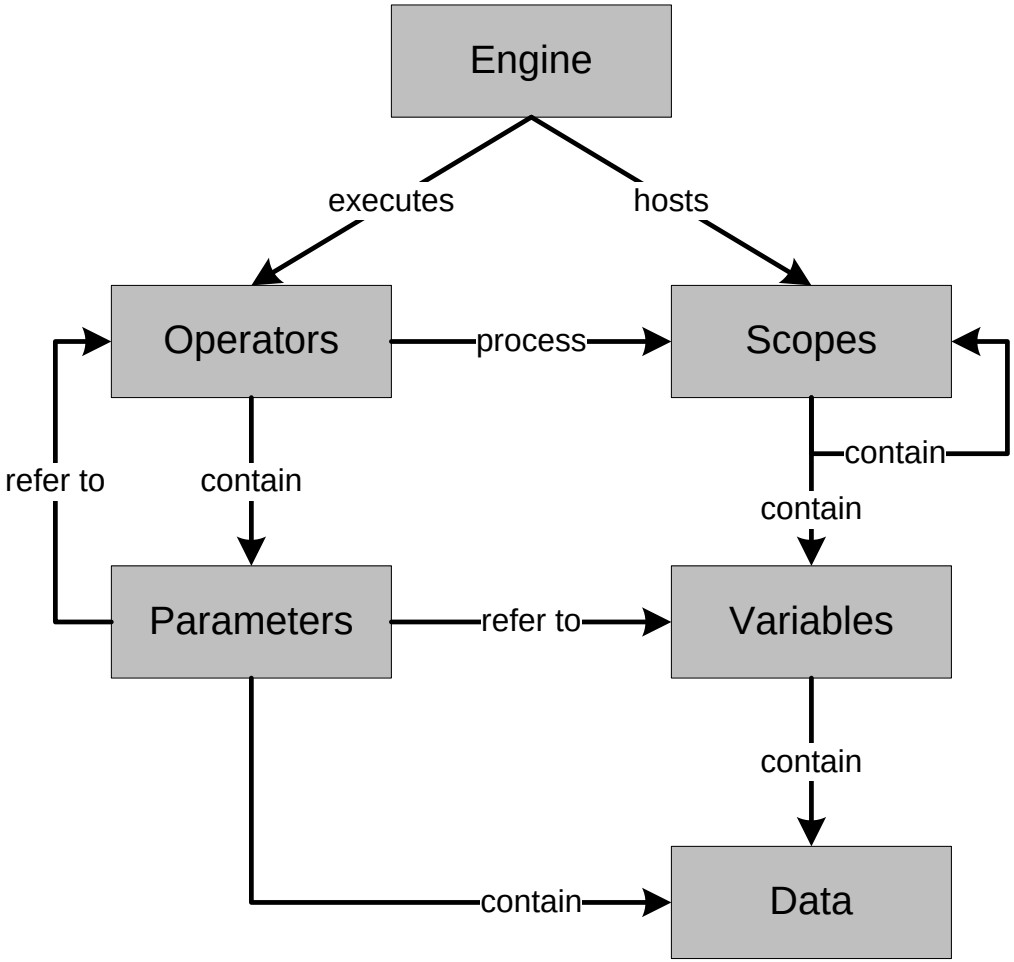
HeuristicLab



HL Algorithm Model

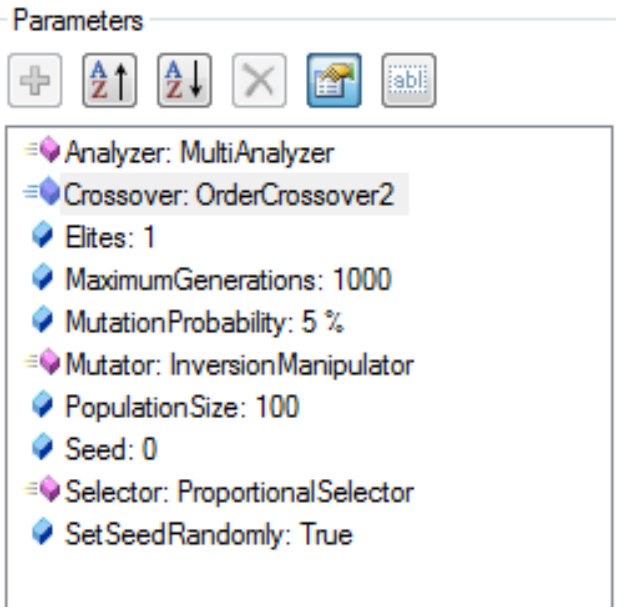
- Typically, HL algorithms are constructed by chaining together operators
- An engine executes these operators
  - Enables pausing and debugging
  - Available engines:
    - Sequential engine
    - Parallel engine
    - Debug engine
    - (Hive engine)

HL Algorithm Model



Parameters

- Used to configure algorithms, problems and operators
- Used for accessing variables in the scope
- E.g. population size, analyzers, crossover operator
- Operators
  - Look up these parameters from the algorithm, problem or scope
  - Use them to store values (in the scope tree)

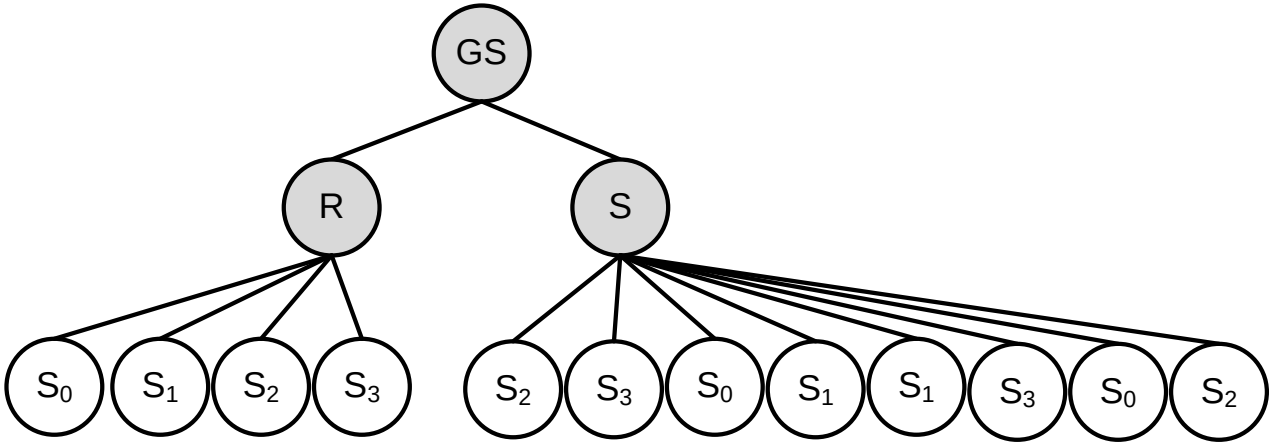


Parameters

- ValueParameter:
  - Stores a value (Item) that can be looked up. E.g. mutation rate, crossover operator,…
- LookupParameter:
  - Looks up parameters/items (variables) from the scope/parent scopes.
- ConstrainedValueParameter:
  - Contains a list of selectable values.
- ScopeTreeLookupParameter:
  - Goes down the scope tree and looks up variables.
- ScopeParameter:
  - Returns the current scope.
- ValueLookupParameter, OptionalConstrainedValueParameter, OperatorParameter, FixedValueParameter, OptionalValueParameter,…

Scopes

- A scope is a node in the scope tree
- Contains link to parent and sub-scopes
- Contains variables (e.g. solutions or their quality)
- Operators usually work on scopes (either directly or through parameters)
- Example - Selection:



Operators

HeuristicLab

- Inherit from SingleSuccessorOperator
- Override the Apply() method
- Must return base.Apply()
  - Returns successor operation
- Use ExecutionContext to access scopes
- Or better: Use parameters to retrieve scopes, values from scopes or manipulate them

Operators

A operator that increments a value from the scope by „Increment“

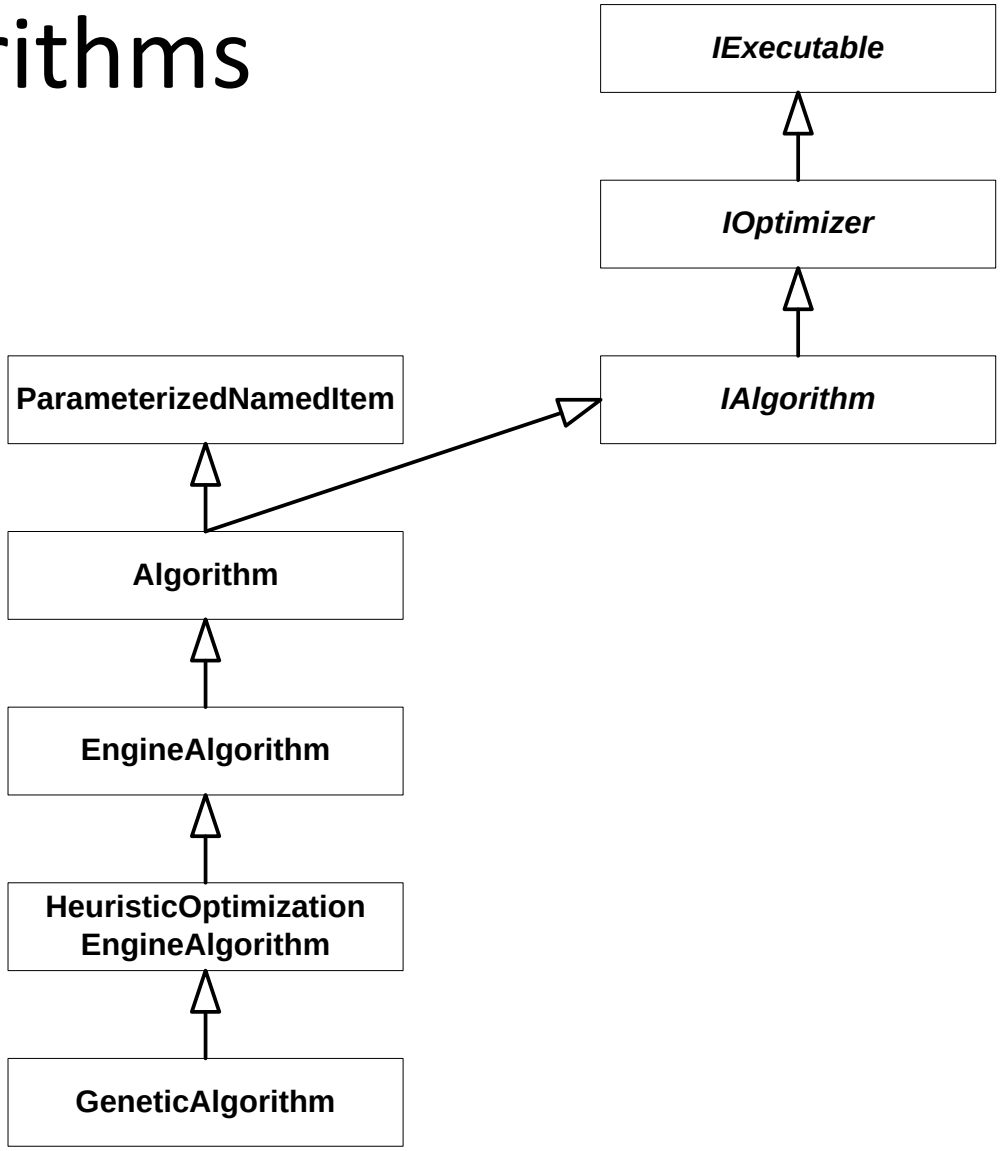
```
[Item("IntCounter", "An operator which increments an integer variable.")]
[StorableClass]
public sealed class IntCounter : SingleSuccessorOperator {
  public LookupParameter<IntValue> ValueParameter {
    get { return (LookupParameter<IntValue>)Parameters["Value"]; }
  }
  public ValueLookupParameter<IntValue> IncrementParameter {
    get { return (ValueLookupParameter<IntValue>)Parameters["Increment"]; }
  }
  public IntValue Increment {
    get { return IncrementParameter.Value; }
    set { IncrementParameter.Value = value; }
  }

  [StorableConstructor]
  private IntCounter(bool deserializing) : base(deserializing) { }
  private IntCounter(IntCounter original, Cloner cloner)
    : base(original, cloner) {
  }
  public IntCounter()
    : base() {
    Parameters.Add(new LookupParameter<IntValue>("Value", "The value which should be incremented."));
    Parameters.Add(new ValueLookupParameter<IntValue>("Increment", "The increment which is added to
the value.", new IntValue(1)));
  }

  public override IDeepCloneable Clone(Cloner cloner) {
    return new IntCounter(this, cloner);
  }

  public override IOperation Apply() {
    if (ValueParameter.ActualValue == null) ValueParameter.ActualValue = new IntValue();
    ValueParameter.ActualValue.Value += IncrementParameter.ActualValue.Value;
    return base.Apply();
  }
}
```

For easier access to parameter values

A parameter for retrieving „Value“ (default name, can be configure with ActualValue) from scope or parent scopes

If the value is not found it can also be created in the scope

Base classes/interfaces for algorithms



Base classes/interfaces for algorithms

- IExecutable (Executable):
  - Defines methods for starting, stopping, etc. of algorithms
- IOptimizer:
  - Contains a run collection
- IAlgorithm:
  - Contains a problem on which the algorithm is applied as well as a result
- Algorithm:
  - Base class, implements IAlgorithm
- EngineAlgorithm:
  - Extensions for execution with an engine (operator graph, scope, engine)
- HeuristicOptimizationEngineAlgorithm:
  - Specifies problem: IHeuristicOptimizationProblem

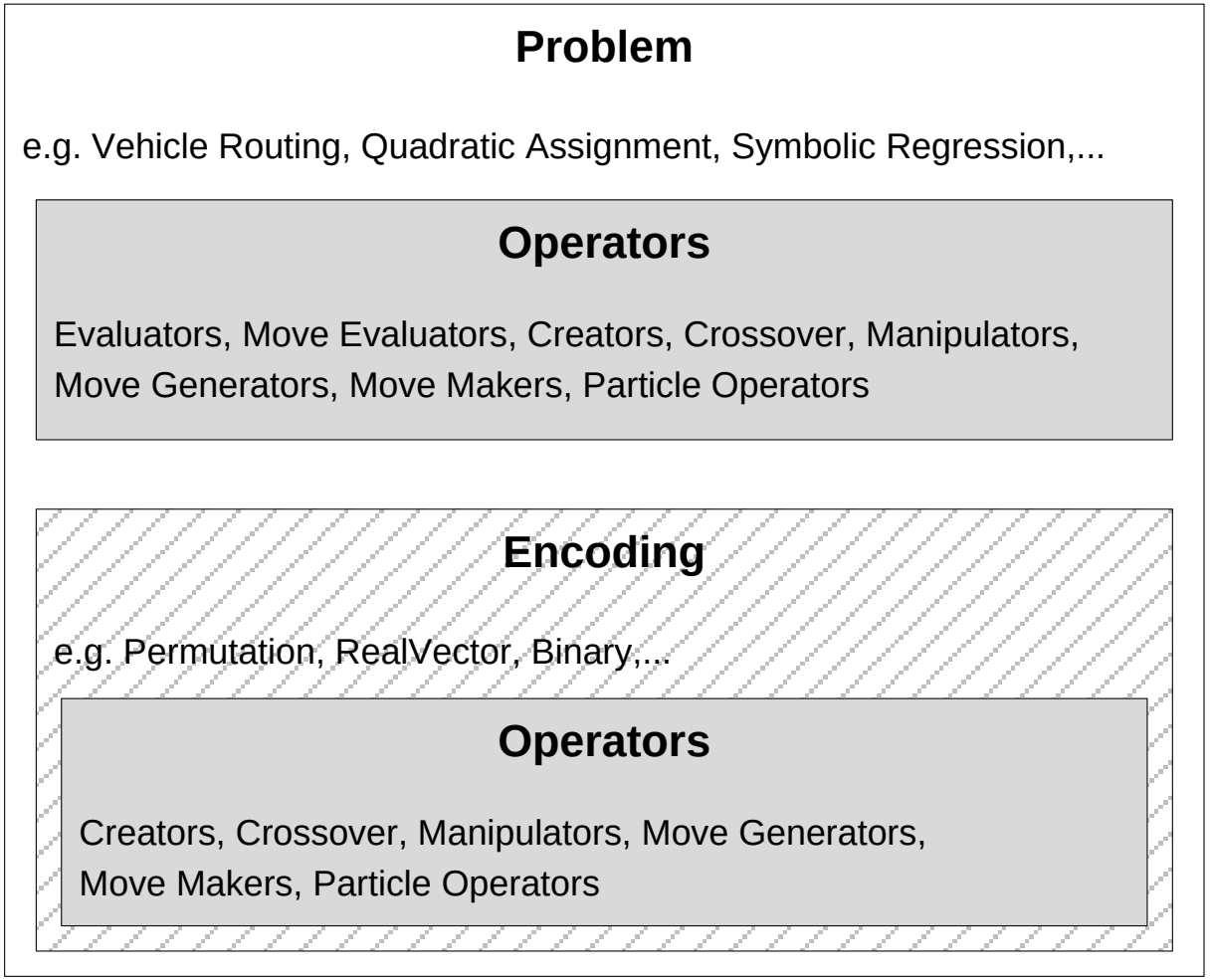
What does an HL algorithm do?

- Create operator graph of algorithm by chaining together operators (the actual algorithm)
- Offer user configuration options through parameters
- Discover operators from the Operators collection of the problem
- Parameterize/wire (react to changes in operators) operators where necessary

Problems

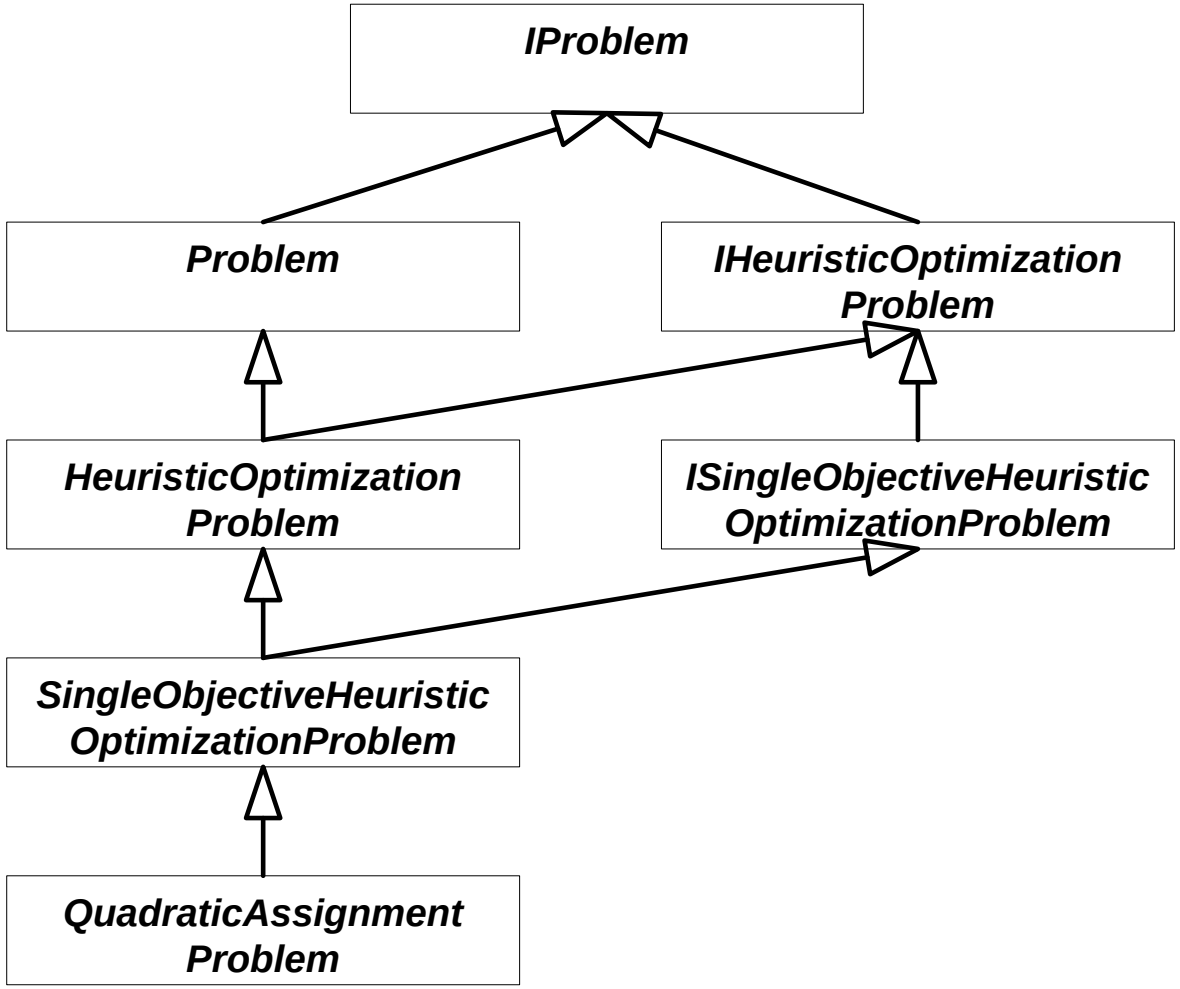
- Use encodings for representing solutions
- Encodings consist of solution candidate definitions and corresponding operators
- Problems contain
  - the evaluator
  - the solution creator
- Define maximization or minimization
- Contain the „problem data“ (e.g. a distance matrix, a simulation, a function definition), usually supplied by a ProblemInstanceProvider
- Can be single- or multi-objective
- Configured with parameters

Problem Architecture



Base classes/interfaces for problems

HeuristicLab



Base classes/interfaces for problems

HeuristicLab

- IProblem:
  - Contains the operators collection; all operators that can be used by the problem, algorithm and user
- IHeuristicOptimizationProblem:
  - Defines solution creator and evaluator
- Problem, HeuristicOptimizationProblem and Single/MultiObjectiveHeuristicOptimizationProblem provide abstract base classes

Recap: What does a HL problem do?

HeuristicLab

- Defines used encoding
- Defines single/multi objective
- Defines min/maximization
- Discovers correct operators
  - Are used by the algorithm
- Wires/parameterizes operators
- Loads problem data using a corresponding problem instance provider

Useful Links

HeuristicLab

http://dev.heuristiclab.com/trac/hl/core/wiki/UsersHowtos

http://dev.heuristiclab.com/trac/hl/core/wiki/Publications

heuristiclab@googlegroups.com

http://www.youtube.com/heuristiclab