

Adaptive Building Intelligence Source Documentation

Ueli Rutishauser and Alain Schaefer

December 10'th 2002

Contents

1	Package aha.framework	5
1.1	Classes	7
1.1.1	CLASS AgentIdentifiers	7
1.1.2	CLASS AHAAgent	8
1.1.3	CLASS AHAMessage	9
1.1.4	CLASS Customizer	10
1.1.5	CLASS DirectorySupport	11
1.1.6	CLASS GeneralUtils	12
1.1.7	CLASS LifeCycleSupport	13
1.1.8	CLASS ObjectFIFO	15
1.1.9	CLASS Pipeline	16
1.1.10	CLASS PipelineData	17
1.1.11	CLASS PipelineHandler	17
1.1.12	CLASS RingBuffer	18
1.1.13	CLASS RingBufferEnumerator	21
1.1.14	CLASS RingBufferIterator	22
1.1.15	CLASS ThreadPool	23
1.1.16	CLASS ThreadPoolWorker	24
2	Package aha.bus.lonworks	25
2.1	Classes	26
2.1.1	CLASS LNSController	26
3	Package aha.rules.learning	27
3.1	Classes	28
3.1.1	CLASS DefinitiveRuleSet	28
3.1.2	CLASS DefinitiveRuleSetIterator	30
3.1.3	CLASS FuzzifiedRule	31
3.1.4	CLASS FuzzifiedRuleFactory	35
3.1.5	CLASS FuzzyLearner	36
3.1.6	CLASS TrainingSet	40
3.1.7	CLASS TrainingSetEntry	44
3.1.8	CLASS TrainingSetIterator	46
4	Package aha.test.rules	48
4.1	Classes	49
4.1.1	CLASS RulesTest	49
4.1.2	CLASS RulesTest2	49

-	2
5 Package aha.tools	51
5.1 Classes	52
5.1.1 CLASS LNSExport	52
6 Package aha.control	53
6.1 Interfaces	55
6.1.1 INTERFACE ControlRemote	55
6.2 Classes	55
6.2.1 CLASS Blinds	55
6.2.2 CLASS BlindSwitches	56
6.2.3 CLASS ControlSupport	56
6.2.4 CLASS ElementGroup	57
6.2.5 CLASS ExecutionPipelineData	58
6.2.6 CLASS InputGroup	59
6.2.7 CLASS Inputs	60
6.2.8 CLASS Learner	60
6.2.9 CLASS LearningUnit	61
6.2.10 CLASS LearningUnitBlinds	65
6.2.11 CLASS LearningUnitLight	67
6.2.12 CLASS Lights	68
6.2.13 CLASS LightSwitches	68
6.2.14 CLASS OutputGroup	69
6.2.15 CLASS PipelineHandlerAble	69
6.2.16 CLASS PipelineHandlerHistory	70
6.2.17 CLASS PipelineHandlerOutput	70
6.2.18 CLASS PipelineHandlerProcessing	71
6.2.19 CLASS Presence	71
6.2.20 CLASS PresenceLight	72
6.2.21 CLASS PresenceMovement	73
6.2.22 CLASS ReinforcementGeneratorBlinds	73
6.2.23 CLASS ReinforcementGeneratorLight	74
6.2.24 CLASS Room	75
6.2.25 CLASS RoomControlAgent	78
6.2.26 CLASS SubCluster	78
6.2.27 CLASS TimeSeriesPlotter	80
6.2.28 CLASS TimeSeriesPlotterExtended	81
6.2.29 CLASS Weather	82
7 Package aha.middleware	85
7.1 Classes	86
7.1.1 CLASS Decision	86
7.1.2 CLASS Message	88
8 Package aha.structure	92
8.1 Classes	93
8.1.1 CLASS Cluster	93
8.1.2 CLASS ElementInfo	93
8.1.3 CLASS ElementType	96
8.1.4 CLASS StructureAgent	97

9	Package aha.bus	99
9.1	Interfaces	100
9.1.1	INTERFACE BusController	100
9.2	Classes	101
9.2.1	CLASS BusAgentJas	101
9.2.2	CLASS BusFrontend	102
9.2.3	CLASS PropertyDescription	103
10	Package aha.sim	104
10.1	Interfaces	105
10.1.1	INTERFACE Simulator	105
10.2	Classes	105
10.2.1	CLASS BusSim	105
10.2.2	CLASS BusSim.Controller	106
10.2.3	CLASS Connection	109
10.2.4	CLASS FloorView	110
10.2.5	CLASS ImageUtilities	112
10.2.6	CLASS Mapping	112
10.2.7	CLASS RoomView	114
10.2.8	CLASS SimulatorImpl	115
10.2.9	CLASS SwingSimulator	116
10.2.10	CLASS Variable	119
10.2.11	CLASS VariableTableModel	119
10.2.12	CLASS WakeUpEvent	121
11	Package aha.test.jas	123
11.1	Classes	124
11.1.1	CLASS AHASstarter	124
11.1.2	CLASS JASAgent1	124
11.1.3	CLASS JASAgent2	125
11.1.4	CLASS JASStarter	126
12	Package aha.messaging	127
12.1	Classes	128
12.1.1	CLASS Constants	128
12.1.2	CLASS MessageDistributionAgentJas	130
12.1.3	CLASS MessagingSupport	131
13	Package aha.rules	134
13.1	Classes	135
13.1.1	CLASS AHARuleBlock	135
13.1.2	CLASS AHARuleSet	136
13.1.3	CLASS ControlModel	137
13.1.4	CLASS DecisionHistory	141
13.1.5	CLASS DecisionHistoryEntry	142
13.1.6	CLASS DynamicRuleTest	143
13.1.7	CLASS Model	145
13.1.8	CLASS Rule	151
13.1.9	CLASS RuleExistsAlreadyException	153
13.1.10	CLASS RuleInfo	154

- 13.1.11 CLASS **RuleLabelIllegalException** 155
- 13.1.12 CLASS **RuleSupport** 156
- 14 Package aha.sim.persons** **159**
- 14.1 Classes 160
- 14.1.1 CLASS **Person** 160
- 14.1.2 CLASS **PersonLocator** 161

Chapter 1

Package aha.framework

<i>Package Contents</i>	<i>Page</i>
Classes	
AgentIdentifiers 7 <i>Names (Display-Name) of all Agents as constants.</i>	
AHAAgent 8 <i>base class for all agents in the AHA system.</i>	
AHAMessage 9 <i>General message container for all messages sent arround within the AHA framework.</i>	
Customizer 10 <i>Generic Customizer.</i>	
DirectorySupport 11 <i>Agent Directory Support class.</i>	
GeneralUtils 12 <i>general utilities, converting of numbers.</i>	
LifeCycleSupport 13 <i>support methods for JAS/ABLE Lifecycle system.</i>	
ObjectFIFO 15 <i>...no description...</i>	
Pipeline 16 <i>Pipeline.</i>	
PipelineData 17 <i>generic parent-class for data that can be sent through a processing pipeline.</i>	
PipelineHandler 17 <i>abstract base class for pipeline system (chain of responsibility).</i>	
RingBuffer 18 <i>A RingBuffer can be used to store a limited number of entries of any type within a buffer.</i>	
RingBufferEnumerator 21 <i>Enumerator for the RingBuffer.</i>	
RingBufferIterator 22 <i>Iterator for the RingBuffer.</i>	
ThreadPool 23 <i>...no description...</i>	
ThreadPoolWorker 24	



1.1 Classes

1.1.1 CLASS AgentIdentifiers

Names (Display-Name) of all Agents as constants.

DECLARATION

```
public class AgentIdentifiers
extends java.lang.Object
```

FIELDS

- public static String JASAGENT1
–
- public static String JASAGENT2
–
- public static String MESSAGE_DISTRIBUTION_AGENT
–
- public static String BUS_AGENT
–
- public static String CONTROL_AGENT
–
- public static String DISPLAY_AGENT
–
- public static String BOSS_AGENT
–
- public static String HISTORY_AGENT
–
- public static String SIMULATOR_AGENT
–
- public static String SWING_SIMULATOR_AGENT
–
- public static String STRUCTURE_AGENT
–
- public static String PERSON_LOCATOR_AGENT

 CONSTRUCTORS

- *AgentIdentifiers*
public **AgentIdentifiers**()

1.1.2 CLASS AHAAgent

base class for all agents in the AHA system. Is a JAS agent. All things common to all agents are implemented in here (support methods)

 DECLARATION

```
public class AHAAgent
extends com.ibm.able.jas.AbleJasDefaultAgent
```

 CONSTRUCTORS

- *AHAAgent*
public **AHAAgent**()
- *AHAAgent*
public **AHAAgent**(java.lang.String name)

 METHODS

- *getAgentDescription*
public **AgentDescription** getAgentDescription()
 - *getBusAgent*
public **AgentDescription** getBusAgent()
 - *getStructureAgent*
public **AgentDescription** getStructureAgent()
 - *init*
public void init()
 - *quitAgent*
public void **quitAgent**(com.ibm.able.jas.AgentLifecycleService thePrincipal)
- **Usage**
- * This method is called when the agent is quit (removed) by the lifecycle service. Overwrite this method to unregister interests with the MessageDistribution Agent.

-
- *registerInterest*
protected boolean **registerInterest**(java.lang.String topic)

- *send*
protected void **send**(javax.agent.service.directory.AgentDescription **destination**, aha.framework.AHAMessage **msg**)
- *sendMessageToDistributor*
protected boolean **sendMessageToDistributor**(aha.framework.AHAMessage **content**, java.lang.String **topic**)
- *sendTimedEvent*
protected void **sendTimedEvent**(long **seconds**,
javax.agent.service.directory.AgentDescription **destination**,
aha.framework.AHAMessage **msg**)
- *unregisterInterest*
public boolean **unregisterInterest**(java.lang.String **topic**)

1.1.3 CLASS AHAMessage

General message container for all messages sent around within the AHA framework.

DECLARATION

```
public class AHAMessage
extends java.lang.Object
implements java.io.Serializable
```

CONSTRUCTORS

- *AHAMessage*
public **AHAMessage**(java.lang.String **type**)
 - *AHAMessage*
public **AHAMessage**(java.lang.String **type**,
javax.agent.service.directory.AgentDescription **sender**)
- **Usage**
- * Construct a message and set the parameter aha.messaging.sender = sender.

METHODS

- *addParameter*
public void **addParameter**(java.lang.Object **key**, java.lang.Object **value**)
- **Usage**
- * Incrementally add a parameter to the existing parameter map.
-
- *getParameters*

– **Usage**

* Get parameter map. Note that this map is not cloned.

• *getType*

```
public String getType( )
```

– **Usage**

* Return type of this message (String-ID).

• *setParameters*

```
public void setParameters( java.util.Map params )
```

– **Usage**

* Set parameters map. Note that all parameters already set (with addParameter) will be overridden by this method.

• *toString*

```
public String toString( )
```

1.1.4 CLASS Customizer

Generic Customizer.

This class allows it to generate generic Customizers for every AdaAgent. Variables that should be modifiable/viewable inside of the Customizer are discovered automatically.

The AdaAgent has to provide a getMethod for every variable that it wants to be displayed inside a customizer. The name of the get Method must be according to this pattern:

```
get[A-Z][a-zA-Z0-9]*_Customizer
```

The AdaAgent must furthermore provide a setMethod for every variable that should be modifiable. Not every variable that has a getMethod must have a setMethod. Variables that only have a getMethod are read only. The name of the set Method must be according to this pattern: set[A-Z][a-zA-Z0-9]*_Customizer

The following types of variables are currently supported: String, Boolean, Double, Float, int, float, double, boolean Other types can easily be added (see method setNewValue in the class CustomizerVariable)

This class uses java regular expression support which means that it requires a jdk1.4 or better.

DECLARATION

```
public class Customizer
extends java.lang.Object
```

CONSTRUCTORS

• *Customizer*

- **Usage**
 - * creates a new customizer; content of the customizer is discovered automatically through reflection, see comment for this class
- **Parameters**
 - * `agent` -

METHODS

- *close*

```
public void close( )
```

 - **Usage**
 - * close this customizers

- *getAgent*

```
public Object getAgent( )
```

 - **Returns** -

- *refresh*

```
public void refresh( )
```

 - **Usage**
 - * get new values for all variables (tell frame to do so)

1.1.5 CLASS DirectorySupport

Agent Directory Support class. Provides convenience methods for directory lookups.

DECLARATION

```
public class DirectorySupport
extends java.lang.Object
```

CONSTRUCTORS

- *DirectorySupport*

```
public DirectorySupport( )
```

METHODS

- *lookupAgent*

```
public static AgentDescription lookupAgent( java.lang.String name )
```

 - **Usage**

- **Parameters**

- * `name` -

- **Returns** -

- **Exceptions**

- * `javax.agent.service.directory.DirectoryFailure` -

- * `javax.agent.service.directory.SearchException` -

- *lookupDirectory*

```
public static AgentDescription lookupDirectory( java.lang.String
displayName )
```

- **Usage**

- * lookup a Agent in the JAS Agent directory. Search query is the display name (`javax.agent.agent-display-name`).

- **Parameters**

- * `displayName` - `javax.agent.agent-display-name`

- **Returns** - null or `AgentDescription`

1.1.6 CLASS GeneralUtils

general utilities, converting of numbers.

DECLARATION

```
public class GeneralUtils
extends java.lang.Object
```

CONSTRUCTORS

- *GeneralUtils*

```
public GeneralUtils( )
```

- **Usage**

- * Creates a new instance of `Utils`

METHODS

- *configureLogging*

```
public static void configureLogging( )
```

- *convertFahrenheitToCelsius*

```
public static double convertFahrenheitToCelsius( double fahrenheit )
```

- **Usage**

- * converts fahrenheit degrees to celsius degrees

- **Parameters**

– **Returns** - celsius degrees (double)

- *lookupJASAgentForRMICalls*

```
public static Object lookupJASAgentForRMICalls( java.lang.String
lookup, java.lang.String host )
```

– **Usage**

* this is a compatibly function to lookup agents over the JAS Agent directory to which a direct RMI Connection should later be established.

– **Parameters**

* **lookup** - agent display name as registered with the directory service
 * **host** - host where this agent is running

– **Returns** -

1.1.7 CLASS LifeCycleSupport

support methods for JAS/ABLE Lifecycle system.

DECLARATION

```
public class LifeCycleSupport
extends java.lang.Object
```

CONSTRUCTORS

- *LifeCycleSupport*

```
public LifeCycleSupport( )
```

METHODS

- *createInitializedJasAgent*

```
public static AbleJasAgent createInitializedJasAgent( java.lang.String
classname, java.lang.String host )
```

– **Usage**

* Instanciate an Agent.

– **Parameters**

* **classname** -
 * **host** -

– **Returns** -

– **Exceptions**

* `com.ibm.able.jas.AgentLifecycleException` -
 * `com.ibm.able.jas.AgentLifecycleFailure` -
 * `javax.agent.service.ServiceException` -

- *createInitializedJasAgent*

```
public static AbleJasAgent createInitializedJasAgent( java.lang.String
classname, java.lang.String host, java.lang.Object initArg )
```

- **Usage**

- * Instanciate an agent.

- **Parameters**

- * classname -

- * host -

- * initArg -

- **Returns -**

- **Exceptions**

- * com.ibm.able.jas.AgentLifecycleException -

- * com.ibm.able.jas.AgentLifecycleFailure -

- * javax.agent.service.ServiceException -

- *createJasAgent*

```
public static AbleJasAgent createJasAgent( java.lang.String classname,
java.lang.String host )
```

- **Usage**

- * Instanciate a JAS agent but don't initialize it.

- **Parameters**

- * classname -

- * host -

- **Returns -**

- **Exceptions**

- * com.ibm.able.jas.AgentLifecycleException -

- * com.ibm.able.jas.AgentLifecycleFailure -

- * javax.agent.service.ServiceException -

- *getLifeCycleService*

```
public static AgentLifecycleService getLifeCycleService( )
```

- **Usage**

- * request lifecycle-instance from the JAS platform.

- **Returns -**

- **Exceptions**

- * javax.agent.service.ServiceException -

- *lookupDesc*

```
public static AbleAgentClassDescription lookupDesc( java.lang.String
classname )
```

- **Usage**

- * Lookup Agent in the Agent Naming Directory.

- **Parameters**

- * classname -

- **Returns -**

* com.ibm.able.jas.AgentLifecycleException -
 * com.ibm.able.jas.AgentLifecycleFailure -
 * javax.agent.service.ServiceException -

- *quitAgent*

```
public static void quitAgent( javax.agent.service.directory.AgentDescription
agent )
```

- **Usage**

- * Stop an agent.

- **Parameters**

- * agent -

- **Exceptions**

- * com.ibm.able.jas.AgentLifecycleFailure -
 - * com.ibm.able.jas.AgentLifecycleException -
 - * javax.agent.service.ServiceException -

1.1.8 CLASS ObjectFIFO

DECLARATION

```
public class ObjectFIFO
extends java.lang.Object
```

CONSTRUCTORS

- *ObjectFIFO*

```
public ObjectFIFO( int cap )
```

METHODS

- *add*

```
public synchronized void add( java.lang.Object obj )
```

- *addEach*

```
public synchronized void addEach( java.lang.Object [] list )
```

- *getCapacity*

```
public int getCapacity( )
```

- *getSize*

```
public synchronized int getSize( )
```

- *isEmpty*

```
public synchronized boolean isEmpty( )
```

- *isFull*

- *remove*
public synchronized Object remove()
- *removeAll*
public synchronized Object removeAll()
- *removeAtLeastOne*
public synchronized Object removeAtLeastOne()
- *waitUntilEmpty*
public synchronized void waitUntilEmpty()
- *waitUntilEmpty*
public synchronized boolean waitUntilEmpty(long msTimeout)
- *waitUntilFull*
public synchronized void waitUntilFull()
- *waitWhileEmpty*
public synchronized void waitWhileEmpty()
- *waitWhileFull*
public synchronized void waitWhileFull()

1.1.9 CLASS Pipeline

Pipeline.

Implements an adapt version of the chain of responsibility pattern (processing doesn't stop if one handler processes data, it is nevertheless carried on to the next one in the chain). This enables serial pipeline-based-processing (stream).

be carefull while using `addHandler(...)` , it is important in which order you add your handlers. they will be processed in the same order then `addHandler(...)` is called during initialization

see "design patterns" (chain of responsibility) by gamma et all and doc of this project for more explanations

DECLARATION

```
public class Pipeline
extends java.lang.Object
```

CONSTRUCTORS

- *Pipeline*
public Pipeline()
- **Usage**
 - * Creates a new instance of Pipeline

METHODS

-
- *addHandler*
public void addHandler(int pos, aha.framework.PipelineHandler handler)
 - *addHandler*
public void addHandler(aha.framework.PipelineHandler handler)
 - *empty*
public void empty()
 - *feed*
public void feed(aha.framework.PipelineData data)
 - *removeHandler*
public void removeHandler(int pos)
 - *size*
public int size()

1.1.10 CLASS PipelineData

generic parent-class for data that can be sent through a processing pipeline. See subclasses for more explanations.

DECLARATION

<pre>public class PipelineData extends java.lang.Object</pre>
--

FIELDS

-
- public static int UNDEFINED
–

CONSTRUCTORS

-
- *PipelineData*
public **PipelineData**(int pipelineDataType)
– **Usage**
* Creates a new instance of PipelineData

1.1.11 CLASS PipelineHandler

DECLARATION

```
public abstract class PipelineHandler
extends java.lang.Object
```

CONSTRUCTORS

- *PipelineHandler*
public **PipelineHandler**()

METHODS

- *feed*
public abstract PipelineData **feed**(aha.framework.PipelineData data)

1.1.12 CLASS RingBuffer

A RingBuffer can be used to store a limited number of entries of any type within a buffer.

As soon as the maximum number of entries is reached, the next entry is added to the end of the list and the first entry is removed from it. In this case, all elements are stored in an Object[]. There are variants for primitive types as well.

DECLARATION

```
public class RingBuffer
extends java.lang.Object
implements java.lang.Cloneable, java.util.Collection, java.io.Serializable
```

SERIALIZABLE FIELDS

- private Object theBuffer
—
- private int size
—
- private int position
—

CONSTRUCTORS

- *RingBuffer*
public RingBuffer()
- *RingBuffer*
public RingBuffer(int aSize)

METHODS

- *add*
public boolean add(java.lang.Object anObject)
 - Usage
 - * Adds an element to the ring buffer, potentially removing the first element to make more room.
 - Parameters
 - * anObject - java.lang.Object
- *addAll*
public boolean addAll(java.util.Collection coll)
- *clear*
public void clear()
- *contains*
public boolean contains(java.lang.Object obj)
- *containsAll*
public boolean containsAll(java.util.Collection coll)
- *elementAt*
public Object elementAt(int aPosition)
 - Usage
 - * Returns the element at the specified position. Positioning starts at 0.
 - Parameters
 - * aPosition - int
 - Returns - java.lang.Object
- *elements*
public Enumeration elements()
 - Usage
 - * Returns an enumeration of all elements within this ring buffer.

– **Returns** - java.util.Enumeration

- *getMaximumSize*

public int **getMaximumSize**()

– **Usage**

* Returns the maximum number of elements in the ring buffer.

– **Returns** - int

- *init*

public void **init**()

– **Usage**

* Initializes some fields.

- *isEmpty*

public boolean **isEmpty**()

- *iterator*

public Iterator **iterator**()

- *remove*

public boolean **remove**(java.lang.Object **obj**)

- *removeAll*

public boolean **removeAll**(java.util.Collection **c**)

– **Usage**

* Removes all elements from the buffer by overwriting them with null and sets the position field to 0.

- *removeElement*

public void **removeElement**(int **aPosition**)

– **Usage**

* Removes the element at aPosition from the RingBuffer.

– **Parameters**

* aPosition - int

- *retainAll*

public boolean **retainAll**(java.util.Collection **c**)

- *setMaximumSize*

public void **setMaximumSize**(int **newSize**)

- * Sets the maximum size of elements in the RingBuffer. All Elements currently in the buffer will be copied to the new buffer, if there is enough space.

Otherwise, only the newest elements will be copied.

This method contains a **synchronized** block, which will be used if theBuffer has already been instantiated.

– **Parameters**

- * `newSize` - int

• *size*

`public int size()`

– **Usage**

- * Returns the current number of elements in the ring buffer.

– **Returns** - int

• *toArray*

`public Object toArray()`

• *toArray*

`public Object toArray(java.lang.Object [] a)`

• *toString*

`public String toString()`

– **Usage**

- * Returns a string representation of the RingBuffer and it's contents.

– **Returns** - java.lang.String

1.1.13 CLASS RingBufferEnumerator

Enumerator for the RingBuffer.

DECLARATION

```
public class RingBufferEnumerator
extends java.lang.Object
implements java.util.Enumeration
```

CONSTRUCTORS

- *RingBufferEnumerator*
 public **RingBufferEnumerator**()
 – **Usage**
 * RingBufferEnumerator constructor comment.

- *RingBufferEnumerator*
 public **RingBufferEnumerator**(aha.framework.RingBuffer aRingBuffer)
 – **Usage**
 * RingBufferEnumerator constructor comment.

METHODS

- *hasMoreElements*
 public boolean **hasMoreElements**()
 – **Usage**
 * hasMoreElements method comment.

- *nextElement*
 public Object **nextElement**()
 – **Usage**
 * nextElement method comment.

1.1.14 CLASS **RingBufferIterator**

Iterator for the RingBuffer.

DECLARATION

```
public class RingBufferIterator
extends java.lang.Object
implements java.util.Iterator
```

CONSTRUCTORS

- *RingBufferIterator*
 public **RingBufferIterator**()
 – **Usage**
 * RingBufferEnumerator constructor comment.

- *RingBufferIterator*
 public **RingBufferIterator**(aha.framework.RingBuffer aRingBuffer)
 – **Usage**
 * RingBufferEnumerator constructor comment.

METHODS

-
- *hasNext*
 public boolean **hasNext**()
 – **Usage**
 * hasMoreElements method comment.

 - *next*
 public Object **next**()
 – **Usage**
 * nextElement method comment.

 - *remove*
 public void **remove**()

1.1.15 CLASS ThreadPool

DECLARATION

```
public class ThreadPool
extends java.lang.Object
```

CONSTRUCTORS

- *ThreadPool*
 public **ThreadPool**(int numberOfThreads)

METHODS

-
- *execute*
 public void **execute**(java.lang.Runnable target)

 - *stopRequestAllWorkers*
 public void **stopRequestAllWorkers**()

 - *stopRequestIdleWorkers*
 public void **stopRequestIdleWorkers**()

1.1.16 CLASS ThreadPoolWorker

DECLARATION

```
public class ThreadPoolWorker
extends java.lang.Object
```

CONSTRUCTORS

- *ThreadPoolWorker*
public **ThreadPoolWorker**(aha.framework.ObjectFIFO idleWorkers)

METHODS

- *getNextWorkerID*
public static synchronized int getNextWorkerID()
- *isAlive*
public boolean isAlive()
- *process*
public void process(java.lang.Runnable target)
- *stopRequest*
public void **stopRequest**()

Chapter 2

Package aha.bus.lonworks

<i>Package Contents</i>	<i>Page</i>
<hr/>	
Classes	
LNSController	26
<i>BusController implementation for LonNetwork</i>	

2.1 Classes

2.1.1 CLASS LNSController

BusController implementation for LonNetwork

DECLARATION

```
public class LNSController
extends java.lang.Object
implements aha.bus.BusController
```

CONSTRUCTORS

- *LNSController*
public **LNSController**()

METHODS

- *connect*
public void **connect**()
 - **Usage**
* connect to the lonbus gateway and loads all network variables.
the ip-address and port are read from the config file
 - **See Also**
* aha.Config
* aha.bus.lonworks.LNSController.loadNetworkVariables()
- *getDescription*
public **PropertyDescription** **getDescription**(java.lang.String name)
- *getProperties*
public **Collection** **getProperties**(java.lang.String regex)
- *getProperty*
public **String** **getProperty**(java.lang.String name)
- *registerPropertyChangeListener*
public void **registerPropertyChangeListener**(java.lang.String name,
java.beans.PropertyChangeListener listener)
- *removePropertyChangeListener*
public void **removePropertyChangeListener**(java.lang.String name,
java.beans.PropertyChangeListener listener)
- *setProperty*
public void **setProperty**(java.lang.String name, java.lang.String value)

Chapter 3

Package `aha.rules.learning`

<i>Package Contents</i>	<i>Page</i>
<hr/>	
Classes	
DefinitiveRuleSet 28 <i>Set of rules which forms the definitive/operative set of rules of the learning algorithm.</i>	
DefinitiveRuleSetIterator 30 <i>Iterator for a definitive ruleset.</i>	
FuzzifiedRule 31 <i>Represents a single fuzzified rule during learning.</i>	
FuzzifiedRuleFactory 35 <i>this class contains static support methods for the creation of FuzzifiedRule instances.</i>	
FuzzyLearner 36 <i>implements the fuzzy learning algorithm as documented in the paper "an anytime learning algorithm for learning maximal structure fuzzy rules".</i>	
TrainingSet 40 <i>Represents a trainingset, consisting of TrainingSetEntries.</i>	
TrainingSetEntry 44 <i>One single trainingset entry.</i>	
TrainingSetIterator 46 <i>Iterator for the class TrainingSet.</i>	

3.1 Classes

3.1.1 CLASS DefinitiveRuleSet

Set of rules which forms the definitive/operative set of rules of the learning algorithm.

DECLARATION

```
public class DefinitiveRuleSet
extends java.lang.Object
```

CONSTRUCTORS

- *DefinitiveRuleSet*
public DefinitiveRuleSet()
 - **Usage**
 - * Set of rules.

METHODS

- *add*
public void add(aha.rules.learning.FuzzifiedRule rule)
 - **Usage**
 - * Add a new rule to the set of definitive rules. The rule is only added if not some other rule with the same antecedents and output is already part of this set.
 - **Parameters**
 - * rule -

- *contains*
public boolean contains(aha.rules.learning.FuzzifiedRule rule)
 - **Usage**
 - * Tests whether rule is already part of this set of rules.
 - **Parameters**
 - * rule -
 - **Returns** - true/false

- *get*
public FuzzifiedRule get(int index)
 - **Usage**
 - * returns an element (for use by iterators).
 - **Parameters**
 - * index -

– **Exceptions**

* java.lang.IndexOutOfBoundsException -

• *getAntecedentSubsumations*

```
public Collection getAntecedentSubsumations(
    aha.rules.learning.FuzzifiedRule ruleEj )
```

– **Usage**

* Ri is the rule represented by this rule, Rj the rule of the argument. Ei/Ej are the different fuzzy input variables in the antecedent.

is true if Ei is a subset (or equal) to Ej

• *instanciate*

```
public Model instanciate( )
```

– **Usage**

* Instanciate this ruleset, create a model for it

• *isAmplificationPossible*

```
public boolean isAmplificationPossible( aha.rules.learning.FuzzifiedRule
    ruleToTest )
```

– **Usage**

* tests whether a amplification is possible compared to the initial ruleset

– **Parameters**

* rule -

– **Returns** -

• *iterator*

```
public Iterator iterator( )
```

– **Usage**

* return iterator

– **Returns** - DefinitiveRuleSetIterator

• *length*

```
public int length( )
```

– **Usage**

* Returns number of rules in this set.

– **Returns** -

• *loadFromPersistentStore*

```
public void loadFromPersistentStore( java.lang.String filename )
```

– **Usage**

* restore state from a file

• *makePersistent*

– **Usage**

* saves all FuzzifiedRule instances that are part of this set to a file

• *remove*

```
public FuzzifiedRule remove( aha.rules.learning.FuzzifiedRule rule )
```

– **Usage**

* removes a rule from the set of definitiverules

• *subsumes*

```
public boolean subsumes( aha.rules.learning.FuzzifiedRule testRule )
```

– **Usage**

* Tests whether testRule subsumes in some rule that is part of this ruleset.

– **Parameters**

* testRule -

– **Returns** - true/false

• *toString*

```
public String toString( )
```

– **Usage**

* debug output

– **Returns** - debug output

3.1.2 CLASS DefinitiveRuleSetIterator

Iterator for a definitive ruleset.

DECLARATION

```
public class DefinitiveRuleSetIterator
extends java.lang.Object
implements java.util.Iterator
```

CONSTRUCTORS

• *DefinitiveRuleSetIterator*

```
public DefinitiveRuleSetIterator( aha.rules.learning.DefinitiveRuleSet
ruleSet )
```

– **Usage**

* Creates a new instance of DefinitiveRuleSetIterator

– **Parameters**

* ruleSet -

METHODS

• *hasNext*

```
public boolean hasNext( )
```

– Usage

* Returns `true` if the iteration has more elements. (In other words, returns `true` if `next` would return an element rather than throwing an exception.)

– **Returns** - `true` if the iterator has more elements.

• *next*

```
public Object next( )
```

– Usage

* Returns the next element in the iteration.

– **Returns** - the next element in the iteration.

• *remove*

```
public void remove( )
```

– Usage

* Removes from the underlying collection the last element returned by the iterator (optional operation). This method can be called only once per call to `next`. The behavior of an iterator is unspecified if the underlying collection is modified while the iteration is in progress in any way other than by calling this method.

3.1.3 CLASS FuzzifiedRule

Represents a single fuzzified rule during learning. this rule can not be used for inferencing, a Rule/AbleRule instance is necessary for this. Such an instance can be created on basis of the information encapsulated in this class (FuzzifiedRule).

DECLARATION

```
public class FuzzifiedRule
extends java.lang.Object
implements java.io.Serializable
```

SERIALIZABLE FIELDS

• private Map antecedents

– antecedents contains all conditions. it is structured as following: (key=value):
inputVariableName=Collection This Collection contains a number of labels of fuzzy

- private String outputName
–
- private String outputvalue
–
- private ArrayList originalOrdering
–

CONSTRUCTORS

- *FuzzifiedRule*
public **FuzzifiedRule**(java.util.Map antecedents, java.util.ArrayList originalOrdering, java.lang.String outputName, java.lang.String outputValue)
 - **Usage**
 - * Instantiate a new FuzzifiedRule.
 - **Parameters**
 - * antecedents -
 - * outputName -
 - * outputValue -

METHODS

- *amplify*
public Collection **amplify**(aha.rules.Model model)
 - **Usage**
 - * Compute all amplifications of this rule.
 - Amplification is implemented as described in the paper "Learning Maximal Structure Rules in Fuzzy Logic for Knowledge Acquisition in Expert Systems" by J.L.Castro et al.
 - Returns a Collection of FuzzifiedRules.
 - Beware: This method does not test whether an amplification is possible or not - it just returns all that exist.
 - **Parameters**
 - * model -
 - **Returns** -
-
- *areAntecedentsMatching*
public boolean **areAntecedentsMatching**(aha.rules.learning.FuzzifiedRule ruleEj)

- * R_i is the rule represented by this rule, R_j the rule of the argument. E_i/E_j are the different fuzzy input variables in the antecedent.

is true if all E_j contain at least one element in common with E_i

- *equals*

```
public boolean equals( aha.rules.learning.FuzzifiedRule obj )
```

- **Usage**

- * Tests whether this FuzzifiedRule is equal to an other FuzzifiedRule instance. This equals method does a deep compare.

- **Parameters**

- * `obj` - Rule to compare to.

- **Returns** - true/false

- *equalsOutputs*

```
public boolean equalsOutputs( aha.rules.learning.FuzzifiedRule ruleEj )
```

- **Usage**

- * Tests whether the output of this fuzzifiedRule and ruleEj are the same (outputvariable and outputvalue).

- **Parameters**

- * `ruleEj` -

- **Returns** -

- *getAbleRule*

```
public AbleRule getAbleRule( )
```

- **Usage**

- * returns the able rule that is connected with this fuzzy rule (if there is one)

- **Returns** - AbleRule instance.

- *getAbleRule*

```
public AbleRule getAbleRule( aha.rules.Model model, java.lang.String ruleLabel )
```

- **Usage**

- * constructs a ready-for-inferencing ablerule from this fuzzifiedRule

- **Parameters**

- * `model` -
- * `ruleLabel` -

- **Returns** -

- *getAntecedents*

```
public Map getAntecedents( )
```

- **Usage**

- * Getter for property antecedents.

- **Returns** - Value of property antecedents.

- *getOutputName*

```
public String getOutputName( )
```

- **Usage**

- * Getter for property outputName.

- **Returns** - Value of property outputName.

- *getOutputvalue*

```
public String getOutputvalue( )
```

- **Usage**

- * Getter for property outputvalue.

- **Returns** - Value of property outputvalue.

- *isAntecedentRealSubset*

```
public boolean isAntecedentRealSubset( java.lang.String key,
java.util.Collection compareValues )
```

- **Usage**

- * tests whether the conditions of variable key are a real subset. a subset that is equal is not a real subset. (there needs to be at least one element less to be a real subset)

- **Parameters**

- * key -

- * compareValues -

- **Returns** -

- *isAntecedentSubset*

```
public boolean isAntecedentSubset( java.lang.String key,
java.util.Collection compareValues )
```

- **Usage**

- * tests whether all antecedent conditions of variable "key" with conditions values are subsets of this rule. a valid subset is also a equivalent subset.

- **Parameters**

- * key -

- * compareValues -

- **Returns** -

- *isAntecedentSubsumation*

```
public boolean isAntecedentSubsumation( aha.rules.learning.FuzzifiedRule
ruleEj )
```

- **Usage**

- * Ri is the rule represented by this rule, Rj the rule of the argument. Ei/Ej are the different fuzzy input variables in the antecedent.

is true if Ei is a subset (or equal) to Ej

- *removeAllConditionsFrom*

```
public FuzzifiedRule removeAllConditionsFrom(
```

- **Usage**
 - * removes all conditions from this rule that are part of toRemove and returns the new rule. the existing rule is cloned before the entries are removed.
 - **Parameters**
 - * toRemove -
 - **Returns** -
-

- *setAntecedents*

```
public void setAntecedents( java.util.Map antecedents )
```

- **Usage**
 - * Setter for property antecedents.
 - **Parameters**
 - * antecedents - New value of property antecedents.
-

- *setOutputName*

```
public void setOutputName( java.lang.String outputName )
```

- **Usage**
 - * Setter for property outputName.
 - **Parameters**
 - * outputName - New value of property outputName.
-

- *setOutputvalue*

```
public void setOutputvalue( java.lang.String outputvalue )
```

- **Usage**
 - * Setter for property outputvalue.
 - **Parameters**
 - * outputvalue - New value of property outputvalue.
-

- *subsumes*

```
public boolean subsumes( aha.rules.learning.FuzzifiedRule ruleEj )
```

- **Usage**
 - * tests whether Ej subsumes this rule.
 - **Parameters**
 - * ruleEj -
 - **Returns** - true/false
-

- *toString*

```
public String toString( )
```

- **Usage**
 - * Returns this rule as a string in set notation.
- **Returns** -

3.1.4 CLASS FuzzifiedRuleFactory

DECLARATION

```
public class FuzzifiedRuleFactory
extends java.lang.Object
```

CONSTRUCTORS

- *FuzzifiedRuleFactory*
public **FuzzifiedRuleFactory**()

METHODS

- *create*
public static FuzzifiedRule **create**(java.util.Map antecedents,
java.util.ArrayList originalOrdering, java.lang.String outputName,
java.lang.String outputValue)

– **Usage**
* instantiate a FuzzifiedRule based on an other Fuzzified Rule

antecedents are cloned.

– **Parameters**
* originalOrdering -
* antecedents -
* outputName -
* outputValue -

– **Returns** - null if rule couldn't be instantiated

- *createFromTrainingEntry*
public static FuzzifiedRule **createFromTraingEntry**(java.util.Map
fuzzifiedInputs, java.util.ArrayList originalOrdering, java.lang.String
outputName, java.lang.String outputValue)

– **Usage**
* instantiate a new fuzzyrule, based on a trainingsetentry (which is already fuzzified).

– **Parameters**
* originalOrdering -
* fuzzifiedInputs -
* outputName -
* outputValue -

– **Returns** - null if rule couldn't be instantiated

3.1.5 CLASS FuzzyLearner

implements the fuzzy learning algorithm as documented in the paper "an anytime learning algorithm for learning maximal structure fuzzy rules"

DECLARATION

```
public class FuzzyLearner
extends java.lang.Object
```

CONSTRUCTORS

- *FuzzyLearner*
public **FuzzyLearner**()

METHODS

- *addIncremental*
public boolean **addIncremental**(java.lang.String set, java.lang.String values)
 - **Usage**
* Add a new valueset incrementally (punishment or reward). This triggers a learnin cycle.
 - **Parameters**
* set - set of values as configured in the generalsettings of this fuzzy learner.
* values -
 - **Returns** -

- *constructInitialRuleset*
public boolean **constructInitialRuleset**()
 - **Usage**
* step 1 of the algorithm: fuzzify the training set and create initial ruleset
 - **Returns** -

- *createRuleset*
public void **createRuleset**()
 - **Usage**
* transform definitive ruleset into a ruleset that can be used for inferencing

- *debugOut*
public void **debugOut**()
 - **Usage**
* printout the trainingset (all entries)

- *getFuzzifiedDefinitiveRuleSet*
public Collection **getFuzzifiedDefinitiveRuleSet**()
 - **Usage**

– **Returns** -

- *getProductionModel*

```
public Model getProductionModel( )
```

– **Usage**

- * returns the productionModel for use as inferencing engine. only returns a valid model if everything has been initialized/generated before.

this model is not cloned and gets automatically updated when something new is learned from feedback. all access to the productionModel needs to be synchronized to prevent the model from being changed while in use by some other thread.

– **Returns** -

- *getTrainingSetInitial*

```
public TrainingSet getTrainingSetInitial( )
```

– **Usage**

- * Getter for property trainingSetInitial.

– **Returns** - Value of property trainingSetInitial.

- *learn*

```
public void learn( )
```

– **Usage**

- * generate definitive ruleset from initialruleset through amplification and subsumption
-

- *load*

```
public void load( java.lang.String fileNameData, java.lang.String
fileNameSettings )
```

– **Usage**

- * load an initial training set from a file.

– **Parameters**

- * fileName -

– **Exceptions**

- * java.io.IOException -
 - * java.io.FileNotFoundException -
-

- *loadFromPersistentStore*

```
public void loadFromPersistentStore( java.lang.String filenameDataSet,
java.lang.String filenameDefinitiveRules, java.lang.String filenameARL,
java.lang.String filenamePROP, java.lang.String filenameSettings )
```

– **Usage**

- * restore fuzzy learner from files

– **Parameters**

- * filenameDataSet - trainingSet
- * filenameDefinitiveRules - all definitive rules (FuzzifiedRule)

- * filenamePROP - (additional properties belonging to the ARL rule file)
- * filenameSettings - (settings, specification of inputs, outputs)

- *loadIncremental*

```
public void loadIncremental( java.lang.String fileName )
```

- **Usage**

- * loads incremental updates from file (for debugging purposes)

- **Parameters**

- * fileName -

- **Exceptions**

- * java.io.IOException -
 - * java.io.FileNotFoundException -
-

- *main*

```
public static void main( java.lang.String [] args )
```

- **Usage**

- * - for testing, there is a main method here -

- **Parameters**

- * args -
-

- *makePersistent*

```
public void makePersistent( java.lang.String filenameDataSet,
java.lang.String filenameDefinitiveRules, java.lang.String filenameARL,
java.lang.String filenamePROP )
```

- **Usage**

- * make this FuzzyLearner persistent so that it can be completely reconstructed from scratch from this files.

- **Parameters**

- * filenameDataSet - trainingSet
 - * filenameDefinitiveRules - all definitive rules (FuzzifiedRule)
 - * filenameARL - (ARL rule file)
 - * filenamePROP - (additional properties belonging to the ARL rule file)
-

- *printARL*

```
public void printARL( )
```

- **Usage**

- * debug output.
-

- *printRuleset*

```
public void printRuleset( )
```

- **Usage**

- * Debug output.
-

- *process*

```
public void process( double daylight, double radiationEast, double
```

- **Usage**
 - * Inferencing process (for debugging).
- **Parameters**
 - * `daylight` -
 - * `radiationEast` -
 - * `presence` -

- *setTrainingSetInitial*

```
public void setTrainingSetInitial( aha.rules.learning.TrainingSet
trainingSetInitial )
```

- **Usage**
 - * Setter for property `trainingSetInitial`.
- **Parameters**
 - * `trainingSetInitial` - New value of property `trainingSetInitial`.

- *testTrainingEntries*

```
public void testTrainingEntries( )
```

- **Usage**
 - * verify all trainingset entries (for debugging/testing).

3.1.6 CLASS TrainingSet

Represents a trainingset, consisting of TrainingSetEntries. Supports Operations necessary for learning fuzzy rules automatically and incrementally.

DECLARATION

```
public class TrainingSet
extends java.lang.Object
```

CONSTRUCTORS

- *TrainingSet*

```
public TrainingSet( )
```

 - **Usage**
 - * Instanciate new TrainingSet.

METHODS

- *addEntry*

```
public void addEntry( aha.rules.learning.TrainingSetEntry entry )
```

 - **Usage**

– **Parameters**

* entry -

• *contains*

public boolean **contains**(aha.rules.learning.TrainingSetEntry testEntry)

– **Usage**

* fuzzy contains.

tests whether this training set already contains an entry that has the same fuzzified input and output values. all trainingset entries must be fuzzified before this method can be called.

– **Parameters**

* testEntry -

– **Returns** -

• *fuzzify*

public boolean **fuzzify**()

– **Usage**

* fuzzify all entries of this set (to create initial ruleset)

– **Returns** -

• *get*

public TrainingSetEntry **get**(int index)

– **Usage**

* returns an element

– **Parameters**

* index -

– **Returns** -

– **Exceptions**

* java.lang.IndexOutOfBoundsException -

• *getInputsCol*

public Collection **getInputsCol**()

– **Usage**

* Getter for property inputsCol.

– **Returns** - Value of property inputsCol.

• *getModel*

public Model **getModel**()

– **Usage**

* returns the Model instance connected to this training set (if there is any).

– **Returns** - null or Model

• *getOutputsCol*

- **Usage**
 - * Getter for property outputsCol.
 - **Returns** - Value of property outputsCol.
-

- *initializeRules*

```
public void initializeRules( )
```

- **Usage**
 - * load initial ruleset from file
-

- *isAmplificationPossible*

```
public boolean isAmplificationPossible( aha.rules.learning.FuzzifiedRule rule
)
```

- **Usage**
 - * tests whether a amplification is possible compared to the initial ruleset
 - **Parameters**
 - * rule -
 - **Returns** -
-

- *iterator*

```
public Iterator iterator( )
```

- **Usage**
 - * return iterator
 - **Returns** -
-

- *length*

```
public int length( )
```

- **Usage**
 - * returns number of elements that this set contains.
 - **Returns** -
-

- *loadFromFile*

```
public void loadFromFile( java.lang.String fileNameData, java.lang.String
fileNameSettings )
```

- **Usage**
 - * loads a training set from a file
 - **Parameters**
 - * fileNameData -
 - * fileNameSettings -
 - **Exceptions**
 - * java.io.IOException -
 - * java.io.FileNotFoundException -
-

- *loadFromPersistentStore*

```
public void loadFromPersistentStore( java.lang.String filename )
```

- * restore state from a file
 - **Parameters**
 - * filename -

- *loadSettings*

```
public void loadSettings( java.lang.String filenameSettings )
```

 - **Usage**
 - * load general settings, specification of outputs and inputs
 - **Parameters**
 - * filenameSettings - filename+path

- *loadTrainingSetEntryIncremental*

```
public TrainingSetEntry loadTrainingSetEntryIncremental( java.lang.String
setLabel, java.lang.String values )
```

 - **Parameters**
 - * setLabel -
 - * values -
 - **Returns** -

- *makePersistent*

```
public void makePersistent( java.lang.String filename )
```

 - **Usage**
 - * saves all FuzzifiedRule instances that are part of this set to a file
 - **Parameters**
 - * filename - full path+filename

- *removeEntry*

```
public void removeEntry( aha.rules.learning.TrainingSetEntry entry )
```

 - **Usage**
 - * remove a entry from this trainingset
 - **Parameters**
 - * entry -

- *setInputsCol*

```
public void setInputsCol( java.util.Collection inputsCol )
```

 - **Usage**
 - * Setter for property inputsCol.
 - **Parameters**
 - * inputsCol - New value of property inputsCol.

- *setOutputsCol*

```
public void setOutputsCol( java.util.Collection outputsCol )
```

 - **Usage**
 - * Setter for property outputsCol.

* outputsCol - New value of property outputsCol.

- *toString*

```
public String toString( )
```

- **Usage**

* for debugging output whole trainingset in real and fuzzy terms

- **Returns** -

3.1.7 CLASS TrainingSetEntry

One single trainingset entry. Supports the fuzzification process.

DECLARATION

```
public class TrainingSetEntry
extends java.lang.Object
```

CONSTRUCTORS

- *TrainingSetEntry*

```
public TrainingSetEntry( java.lang.String label, java.util.ArrayList
pinputs, java.util.ArrayList pinputsFuzzify, java.util.ArrayList poutputs,
java.lang.String values )
```

- **Parameters**

* pinputsFuzzify -
 * label -
 * pinputs -
 * poutputs -
 * values -

METHODS

- *equalsFuzzy*

```
public boolean equalsFuzzy( aha.rules.learning.TrainingSetEntry testEntry )
```

- **Usage**

* compares the fuzzified version of two training set entries.

- *getFuzzifiedRule*

```
public FuzzifiedRule getFuzzifiedRule( )
```

- **Usage**

* Get the FuzzifiedRule instance version of this entry.

Returns - FuzzifiedRule instance or null

- *getInputs*
public Map getInputs()
 - **Usage**
* Getter for property inputs.
 - **Returns** - Value of property inputs.

- *getLabel*
public String getLabel()
 - **Usage**
* Getter for property label.
 - **Returns** - Value of property label.

- *getValues*
public String getValues()
 - **Usage**
* Getter for property values.
 - **Returns** - Value of property values.

- *getValuesForProcessing*
public Object getValuesForProcessing()
 - **Usage**
* Get real values for fuzzification.
 - **Returns** -

- *isFuzzified*
public boolean isFuzzified()
 - **Usage**
* true if this rule has been fuzzified.
 - **Returns** -

- *setFuzzificationResult*
public void setFuzzificationResult(java.lang.Object [] res)
 - **Usage**
* Set result of fuzzification. Called by FuzzyLearner.
 - **Parameters**
* res -

- *setFuzzifiedRule*
public void setFuzzifiedRule(aha.rules.learning.FuzzifiedRule fuzzifiedRule)
 - **Usage**
* associate a FuzzifiedRule with this trainingSet entry.
 - **Parameters**

-
- *setInputs*
public void **setInputs**(java.util.Map **inputs**)
 - **Usage**
 - * Setter for property inputs.
 - **Parameters**
 - * **inputs** - New value of property inputs.
-
- *setLabel*
public void **setLabel**(java.lang.String **label**)
 - **Usage**
 - * Setter for property label.
 - **Parameters**
 - * **label** - New value of property label.
-
- *setValues*
public void **setValues**(java.lang.String **values**)
 - **Usage**
 - * Setter for property values.
 - **Parameters**
 - * **values** - New value of property values.
-
- *toString*
public String **toString**()
 - **Usage**
 - * debug output.
 - **Returns** -

3.1.8 CLASS TrainingSetIterator

Iterator for the class TrainingSet.

DECLARATION

```
public class TrainingSetIterator
extends java.lang.Object
implements java.util.Iterator
```

CONSTRUCTORS

- *TrainingSetIterator*
`public TrainingSetIterator(aha.rules.learning.TrainingSet trainingSet)`
 - **Usage**
 - * Creates a new instance of TrainingSetIterator
 - **Parameters**
 - * `trainingSet` -

METHODS

- *hasNext*
`public boolean hasNext()`
 - **Usage**
 - * Returns `true` if the iteration has more elements. (In other words, returns `true` if `next` would return an element rather than throwing an exception.)
 - **Returns** - `true` if the iterator has more elements.

- *next*
`public Object next()`
 - **Usage**
 - * Returns the next element in the iteration.
 - **Returns** - the next element in the iteration.

- *remove*
`public void remove()`
 - **Usage**
 - * Removes from the underlying collection the last element returned by the iterator (optional operation). This method can be called only once per call to `next`. The behavior of an iterator is unspecified if the underlying collection is modified while the iteration is in progress in any way other than by calling this method.
 - **Exceptions**
 - * `java.lang.UnsupportedOperationException` - if the `remove` operation is not supported by this Iterator.

Chapter 4

Package `aha.test.rules`

<i>Package Contents</i>	<i>Page</i>
<hr/>	
Classes	
RulesTest <i>...no description...</i>	49
RulesTest2 <i>test file for testing ibm's claim that getComment/setComment is working.</i>	49

4.1 Classes

4.1.1 CLASS RulesTest

DECLARATION

```
public class RulesTest
extends java.lang.Object
```

CONSTRUCTORS

- *RulesTest*
public **RulesTest**()
– **Usage**
* Creates a new instance of RulesTest

METHODS

- *generateNewRules*
public void **generateNewRules**()
- *init*
public void **init**()
- *main*
public static void **main**(java.lang.String [] args)

4.1.2 CLASS RulesTest2

test file for testing ibm's claim that getComment/setComment is working. ->it works with /** but not with /* and // as start of comment

DECLARATION

```
public class RulesTest2
extends java.lang.Object
```

CONSTRUCTORS

- *RulesTest2*
public **RulesTest2**()
– **Usage**

METHODS

- *main*
public static void **main**(java.lang.String [] args)

Chapter 5

Package aha.tools

<i>Package Contents</i>	<i>Page</i>
<hr/>	
Classes	
LNSExport	52
<i>...no description...</i>	

5.1 Classes

5.1.1 CLASS LNSEExport

DECLARATION

```
public class LNSEExport
extends java.lang.Object
```

CONSTRUCTORS

- *LNSEExport*
public LNSEExport()

METHODS

- *connect*
public void connect()
 - **Usage**
 - * connect to the lonbus gateway and loads all network variables.
the ip-address and port are read from the config file
 - **See Also**
 - * aha.Config
 - * aha.tools.LNSEExport.loadNetworkVariables()

- *main*
public static void main(java.lang.String [] arguments)

Chapter 6

Package aha.control

<i>Package Contents</i>	<i>Page</i>
Interfaces	
ControlRemote 55 <i>remote interface for rmi calls to ControlAgent.</i>	
Classes	
Blinds 55 <i>The class Blinds represents a group of output variables.</i>	
BlindSwitches 56 <i>Handling of blind switches.</i>	
ControlSupport 56 <i>Various static support methods for the control agent.</i>	
ElementGroup 57 <i>Represents a group of variables.</i>	
ExecutionPipelineData 58 <i>data to be transfered through pipeline to initiate processing of decisions.</i>	
InputGroup 59 <i>Represents a group of input values.</i>	
Inputs 60 <i>Some rule specific constants that are specific for this agent.</i>	
Learner 60 <i>Independent Thread that processes rewards and punishments.</i>	
LearningUnit 61 <i>Generic learning unit.</i>	
LearningUnitBlinds 65 <i>Learning unit for blinds.</i>	
LearningUnitLight 67 <i>Learning unit for dealing with lights.</i>	
Lights 68 <i>represents a group of lights.</i>	
LightSwitches 68 <i>Handling of light switches.</i>	
OutputGroup 69 <i>Represents a number of output variables.</i>	
PipelineHandlerAble 69	

	<i>this pipeline handler acts as a gateway between ada and the internal processing structure data that is available in the pipeline is converted and put into the able message delivery system for processing by other agents.</i>	
PipelineHandlerHistory		70
	<i>Pipeline handler that acts as supplier for the history agent</i>	
PipelineHandlerOutput		70
	<i>Pipeline handler for debug output</i>	
PipelineHandlerProcessing		71
	<i>Pipeline handler that executes decisions taken by the inference engine (feeder of the pipeline).</i>	
Presence		71
	<i>presence detector handling normalizes HVAC (movement) and daylight values</i>	
	<i>filters HVAC pulses so that presence yes/no decisions are only taken with a low frequency</i>	
	<i>runs as an independent thread that is being notified of changes by the controlAgent that is responsible for this room</i>	
	<i>one instance of this class handles all the presence detectors for one room</i>	
PresenceLight		72
	<i>represents a number of daylight input variables.</i>	
PresenceMovement		73
	<i>Represents a number of HVAC input values.</i>	
ReinforcementGeneratorBlinds		73
	<i>global reinforcement generator; continuously evaluates a goal function and punishes/rewards the system by feeding data samples to it for learning.</i>	
ReinforcementGeneratorLight		74
	<i>global reinforcement generator; continuously evaluates a goal function and punishes/rewards the system by feeding data samples to it for learning.</i>	
Room		75
	<i>general interface to a room.</i>	
RoomControlAgent		78
	<i>Agent that is responsible to control one specific room.</i>	
SubCluster		78
	<i>\$Id: SubCluster.java,v 1.6 2002/12/05 09:46:43 urut Exp \$</i>	
TimeSeriesPlotter		80
	<i>plotting time series (with one data set).</i>	
TimeSeriesPlotterExtended		81
	<i>plotting time series (with multiple data sets).</i>	
Weather		82
	<i>Represents all data of global sensors (not related to a room, outside weather sensors).</i>	

6.1 Interfaces

6.1.1 INTERFACE ControlRemote

remote interface for rmi calls to ControlAgent.

This interface doesn't contain any methods so far but it is required to identify classes who have certain capabilities.

DECLARATION

```
public interface ControlRemote
implements java.rmi.Remote
```

6.2 Classes

6.2.1 CLASS Blinds

The class Blinds represents a group of output variables. It encapsulates all blinds-specific output variable handling.

DECLARATION

```
public class Blinds
extends aha.control.OutputGroup
```

CONSTRUCTORS

- *Blinds*

```
public Blinds( javax.agent.service.directory.AgentDescription sender,
               javax.agent.Locator senderLocator )
```

METHODS

- *allChange*

```
public boolean allChange( java.lang.String parameter1, java.lang.String
                           parameter2 )
```

 - **Usage**
 * currently not implemented.
- *allDown*

```
public boolean allDown( )
```

* send command to BusAgent: All blinds down

- *allStop*

```
public boolean allStop( )
```

- **Usage**

* send command to BusAgent: All blinds stop

- *allUp*

```
public boolean allUp( )
```

- **Usage**

* send command to BusAgent: All blinds up.

6.2.2 CLASS BlindSwitches

Handling of blind switches. Generates punishment input for learning.

DECLARATION

```
public class BlindSwitches
extends aha.control.InputGroup
```

CONSTRUCTORS

- *BlindSwitches*

```
public BlindSwitches( javax.agent.service.directory.AgentDescription
sender, javax.agent.Locator senderLocator, aha.control.LearningUnitBlinds
learningUnit )
```

METHODS

- *receive*

```
public void receive( aha.framework.AHAMessage msg )
```

6.2.3 CLASS ControlSupport

Various static support methods for the control agent.

DECLARATION

```
public class ControlSupport
extends java.lang.Object
```

CONSTRUCTORS

- *ControlSupport*
public **ControlSupport**()

METHODS

- *getSecSinceMidnight*
public static double **getSecSinceMidnight**()

6.2.4 CLASS ElementGroup

Represents a group of variables.

DECLARATION

```
public class ElementGroup
extends java.lang.Object
```

CONSTRUCTORS

- *ElementGroup*
public **ElementGroup**(javax.agent.service.directory.AgentDescription
sender, javax.agent.Locator senderLocator)

METHODS

- *addVariable*
public void **addVariable**(java.lang.String varName)
 - **Usage**
* add a variable to this group
- *contains*
public boolean **contains**(java.lang.String name)
 - **Usage**
* evaluates if the variable called "name" is registered with this element group
- *getElementType*
public String **getElementType**()
 - **Usage**
* getter

- *receive*

```
public void receive( aha.framework.AHAMessage msg )
```

- *removeRegistration*

```
public void removeRegistration( )
```

 - **Usage**
 * unregister all listeners for this elementgroup

- *removeVariable*

```
public Object removeVariable( java.lang.String varName )
```

 - **Usage**
 * remove a variable from this group

- *requestUpdateOfVariables*

```
public void requestUpdateOfVariables( )
```

 - **Usage**
 * request an update for all variables of this group

- *requestVariables*

```
public void requestVariables( int clusterNr )
```

 - **Usage**
 * request all variables of this elementtype

- *requestVariables*

```
public void requestVariables( int clusterNr, java.lang.String  
elementType )
```

- *setElementType*

```
public void setElementType( java.lang.String elementType )
```

 - **Usage**
 * setter

- *size*

```
public int size( )
```

 - **Usage**
 * number of variables

6.2.5 CLASS ExecutionPipelineData

data to be transfered through pipeline to initiate processing of decisions.

DECLARATION

```
public class ExecutionPipelineData  
extends aha.framework.PipelineData
```

CONSTRUCTORS

- *ExecutionPipelineData*

```
public ExecutionPipelineData( aha.middleware.Decision decision,
    AbleDefaultAgent sender )
```

 - **Usage**
 - * Creates a new instance of ExecutionPipelineData

METHODS

- *getDecision*

```
public Decision getDecision( )
```

 - **Usage**
 - * return content of this data holder
 - **Returns** - Decision
-
- *getSender*

```
public AbleDefaultAgent getSender( )
```

 - **Usage**
 - * to find out which agent put this decision into the pipeline.
 - **Returns** - agent

6.2.6 CLASS InputGroup

Represents a group of input values.

DECLARATION

```
public class InputGroup
extends aha.control.ElementGroup
```

CONSTRUCTORS

- *InputGroup*

```
public InputGroup( javax.agent.service.directory.AgentDescription sender,
    javax.agent.Locator senderLocator )
```

METHODS

- *getAllValues*

```
public Collection getAllValues( )
```

 - **Usage**

6.2.7 CLASS Inputs

Some rule specific constants that are specific for this agent. Most of the variables defined here are just convenient aliases for variables defined elsewhere.

DECLARATION

```
public class Inputs
extends java.lang.Object
```

FIELDS

- public static String Temperature
–
- public static String DayTime
–
- public static String Presence
–
- public static String DayLightIndoor
–
- public static String RadiationEast
–
- public static String RadiationWest
–
- public static String RadiationSouth
–

CONSTRUCTORS

- *Inputs*
public **Inputs**()

6.2.8 CLASS Learner

Independent Thread that processes rewards and punishments. After a dynamic rule change has occurred as an effect of a reward/punishment an announcement with the new rulebase is sent out to the DistributionAgent. Rules sent out to the DistributionAgent aren't meant to be used for processing; There only purpose is for display/GUI usage.

DECLARATION

```
public class Learner
extends java.lang.Thread
```

CONSTRUCTORS

- *Learner*

```
public Learner( javax.agent.service.directory.AgentDescription sender,
                javax.agent.Locator senderLocator, aha.control.LearningUnit learningUnit )
```

METHODS

- *addSampleToLearn*

```
public void addSampleToLearn( java.lang.String sample )
```

 - **Usage**
 * add a new sample to learn from to the queue.
 - **Parameters**
 * **sample** - A sample containing all inputs and output value.

- *run*

```
public void run( )
```

 - **Usage**
 * thread run.

- *setFuzzyLearner*

```
public void setFuzzyLearner( aha.rules.learning.FuzzyLearner fuzzyLearner
                             )
```

 - **Usage**
 * part of initialization:set callback to fuzzylearner.
 - **Parameters**
 * **fuzzyLearner** - fuzzyLearner instance

- *stopThisThread*

```
public void stopThisThread( )
```

 - **Usage**
 * stop Thread.

6.2.9 CLASS LearningUnit

Generic learning unit. This class needs to be inherited and some methods need to be overwritten for a

DECLARATION

```
public class LearningUnit
extends java.lang.Object
```

CONSTRUCTORS

- *LearningUnit*

```
public LearningUnit( javax.agent.service.directory.AgentDescription
sender, javax.agent.Locator senderLocator, int clusterNr,
aha.control.Weather weather, aha.control.PresenceLight presenceLight,
aha.control.PresenceMovement presenceMovement, aha.control.Presence
presence )
```

 - **Usage**
 - * construct a new learning Unit.
 - **Parameters**
 - * sender -
 - * senderLocator -
 - * clusterNr - clusterNR of this learning unit.
 - * weather -
 - * presenceLight -
 - * presenceMovement -
 - * presence -

METHODS

- *addSampleToLearn*

```
public void addSampleToLearn( java.lang.String output )
```

 - **Usage**
 - * add a new sample to the queue of the independent learner thread.
 - **Parameters**
 - * output -

- *getClusterNr*

```
public int getClusterNr( )
```

 - **Usage**
 - * Getter for property clusterNr.
 - **Returns** - Value of property clusterNr.

- *getHumidity_Customizer*

```
public String getHumidity_Customizer( )
```

 - **Usage**
 - * special function with special name for customizer (because of reflection).

-
- *getIlluminance1_Customizer*
 public String **getIlluminance1_Customizer**()
 – **Usage**
 * special function with special name for customizer (because of reflection).
 – **Returns** -

 - *getPresenceLightState_Customizer*
 public String **getPresenceLightState_Customizer**()
 – **Usage**
 * special function with special name for customizer (because of reflection).
 – **Returns** -

 - *getPresenceMovementState_Customizer*
 public boolean **getPresenceMovementState_Customizer**()
 – **Usage**
 * special function with special name for customizer (because of reflection).
 – **Returns** -

 - *getRadiationState_Customizer*
 public String **getRadiationState_Customizer**()
 – **Usage**
 * special function with special name for customizer (because of reflection).
 – **Returns** -

 - *getStateChanged*
 public boolean **getStateChanged**()
 – **Usage**
 * getter for state.
 – **Returns** - state. true/false.

 - *getSubClusterNr_Customizer*
 public String **getSubClusterNr_Customizer**()
 – **Usage**
 * special function with special name for customizer (because of reflection).
 – **Returns** -

 - *getSunEast_Customizer*
 public String **getSunEast_Customizer**()
 – **Usage**
 * special function with special name for customizer (because of reflection).
 – **Returns** -

 - *getSunSouth_Customizer*

- **Usage**
 - * special function with special name for customizer (because of reflection).
- **Returns** -

- *getSunWest_Customizer*

```
public String getSunWest_Customizer( )
```

- **Usage**
 - * special function with special name for customizer (because of reflection).
- **Returns** -

- *getTemperature_Customizer*

```
public String getTemperature_Customizer( )
```

- **Usage**
 - * special function with special name for customizer (because of reflection).
- **Returns** -

- *init*

```
public void init( )
```

- **Usage**
 - * initialize this agent and request structure information.

- *makePersistent*

```
public void makePersistent( )
```

- **Usage**
 - * stores state to files and stops all processing.

- *postProcessing*

```
public void postProcessing( java.lang.Object [] outBuffer )
```

- **Usage**
 - * needs to be overwritten in subclasses, individual handling.
- **Parameters**
 - * outBuffer -

- *process*

```
public void process( )
```

- **Usage**
 - * start inferencing process

- *receive*

```
public void receive( aha.framework.AHAMessage msg )
```

- **Usage**
 - * receive messages and forward them to the respective classes for processing
- **Parameters**
 - * msg

- *setClusterNr*

```
public void setClusterNr( int clusterNr )
```

 - **Usage**
 - * Setter for property clusterNr.
 - **Parameters**
 - * clusterNr - New value of property clusterNr.

- *setStateChanged*

```
protected void setStateChanged( boolean stateChanged )
```

 - **Usage**
 - * setter.
 - **Parameters**
 - * stateChanged -

- *startThreads*

```
protected void startThreads( )
```

 - **Usage**
 - * Start all threads that are necessary to control a room

- *stopThreads*

```
protected void stopThreads( )
```

 - **Usage**
 - * Stop all threads of this learning unit.

- *updateCustomizer*

```
protected void updateCustomizer( )
```

 - **Usage**
 - * update GUI

6.2.10 CLASS LearningUnitBlinds

Learning unit for blinds.

DECLARATION

```
public class LearningUnitBlinds
extends aha.control.LearningUnit
```

CONSTRUCTORS

- *LearningUnitBlinds*

```
public LearningUnitBlinds( javax.agent.service.directory.AgentDescription
sender, javax.agent.Locator senderLocator, int clusterNr,
aha.control.Weather weather, aha.control.PresenceLight presenceLight,
aha.control.PresenceMovement presenceMovement, aha.control.Presence
```

METHODS

- *getBlindAction_Customizer*
public String getBlindAction_Customizer()
 - **Usage**
 * special function with special name for customizer (because of reflection)

- *getBlindAction*
public String getBlindAction()
 - **Usage**
 * returns the last decision taken for the light

- *getLearningUnitType_Customizer*
public String getLearningUnitType_Customizer()

- *getNumberOfBlinds_Customizer*
public String getNumberOfBlinds_Customizer()

- *getNumberOfBlindSwitches_Customizer*
public String getNumberOfBlindSwitches_Customizer()

- *init*
public void init()

- *makePersistent*
public void makePersistent()

- *postProcessing*
public void postProcessing()
 - **Usage**
 * process results and execute decisions specific for this type of learning unit

- *postProcessing*
public void postProcessing(java.lang.Object [] outBuffer)
 - **Usage**
 * process results and execute decisions specific for this type of learning unit

- *receive*
public void receive(aha.framework.AHAMessage msg)
 - **Usage**
 * receive messages and forward them to the respective classes for processing

- *startThreads*
protected void startThreads()

- *stopThreads*
protected void stopThreads()

6.2.11 CLASS LearningUnitLight

Learning unit for dealing with lights.

DECLARATION

```
public class LearningUnitLight
extends aha.control.LearningUnit
```

CONSTRUCTORS

- *LearningUnitLight*

```
public LearningUnitLight( javax.agent.service.directory.AgentDescription
sender, javax.agent.Locator senderLocator, int clusterNr,
aha.control.Weather weather, aha.control.PresenceLight presenceLight,
aha.control.PresenceMovement presenceMovement, aha.control.Presence
presence )
```

METHODS

- *getLearningUnitType_Customizer*

```
public String getLearningUnitType_Customizer( )
```
- *getLightAction_Customizer*

```
public String getLightAction_Customizer( )
```

 - **Usage**
* special function with special name for customizer (because of reflection)
- *getLightAction*

```
public String getLightAction( )
```

 - **Usage**
* returns the last decision taken for the blinds.
- *getNumberOfLights_Customizer*

```
public String getNumberOfLights_Customizer( )
```
- *getNumberOfLightSwitches_Customizer*

```
public String getNumberOfLightSwitches_Customizer( )
```
- *init*

```
public void init( )
```
- *makePersistent*

```
public void makePersistent( )
```
- *postProcessing*

– **Usage**

* process results and execute decisions specific for this type of learning unit

- *receive*

```
public void receive( aha.framework.AHAMessage msg )
```

– **Usage**

* receive messages and forward them to the respective classes for processing

- *startThreads*

```
protected void startThreads( )
```

- *stopThreads*

```
protected void stopThreads( )
```

6.2.12 CLASS Lights

represents a group of lights.

DECLARATION

```
public class Lights
extends aha.control.OutputGroup
```

CONSTRUCTORS

- *Lights*

```
public Lights( javax.agent.service.directory.AgentDescription sender,
javax.agent.Locator senderLocator )
```

METHODS

- *allOff*

```
public boolean allOff( )
```

- *allOn*

```
public boolean allOn( )
```

6.2.13 CLASS LightSwitches

Handling of light switches. Generates punishments.

DECLARATION

```
public class LightSwitches
extends aha.control.InputGroup
```

CONSTRUCTORS

- *LightSwitches*

```
public LightSwitches( javax.agent.service.directory.AgentDescription
sender, javax.agent.Locator senderLocator, aha.control.LearningUnitLight
learningUnit )
```

METHODS

- *receive*

```
public void receive( aha.framework.AHAMessage msg )
```

6.2.14 CLASS OutputGroup

Represents a number of output variables.

DECLARATION

```
public class OutputGroup
extends aha.control.ElementGroup
```

CONSTRUCTORS

- *OutputGroup*

```
public OutputGroup( javax.agent.service.directory.AgentDescription
sender, javax.agent.Locator senderLocator )
```

METHODS

- *sendToBus*

```
protected void sendToBus( java.lang.String value )
```

6.2.15 CLASS PipelineHandlerAble

this pipeline handler acts as a gateway between ada and the internal processing structure data that is available in the pipeline is converted and put into the able message delivery system for processing by other agents. the event delivery queues of the all the subscribers to this messages act as continuation of the local pipeline on a remote location.

DECLARATION

```
public class PipelineHandlerAble
extends aha.framework.PipelineHandler
```

CONSTRUCTORS

- *PipelineHandlerAble*
public **PipelineHandlerAble**()

METHODS

- *feed*
public PipelineData feed(aha.framework.PipelineData data)

6.2.16 CLASS PipelineHandlerHistory

Pipeline handler that acts as supplier for the history agent

DECLARATION

```
public class PipelineHandlerHistory
extends aha.framework.PipelineHandler
```

CONSTRUCTORS

- *PipelineHandlerHistory*
public **PipelineHandlerHistory**(aha.history.HistoryRemote historyRemote)

METHODS

- *feed*
public PipelineData feed(aha.framework.PipelineData data)

6.2.17 CLASS PipelineHandlerOutput

Pipeline handler for debug output

DECLARATION

```
public class PipelineHandlerOutput
extends aha.framework.PipelineHandler
```

CONSTRUCTORS

- *PipelineHandlerOutput*

METHODS

- *feed*

```
public PipelineData feed( aha.framework.PipelineData data )
```

6.2.18 CLASS PipelineHandlerProcessing

Pipeline handler that executes decisions taken by the inference engine (feeder of the pipeline). takes care of oscillation avoidance, timeouts, local history, limits

DECLARATION

```
public class PipelineHandlerProcessing
extends aha.framework.PipelineHandler
```

CONSTRUCTORS

- *PipelineHandlerProcessing*

```
public PipelineHandlerProcessing( aha.control.Room theRoom )
```

METHODS

- *feed*

```
public PipelineData feed( aha.framework.PipelineData data )
```

6.2.19 CLASS Presence

presence detector handling normalizes HVAC (movement) and daylight values

filters HVAC pulses so that presence yes/no decisions are only taken with a low frequency

runs as an independent thread that is being notified of changes by the controlAgent that is responsible for this room

one instance of this class handles all the presence detectors for one room

DECLARATION

```
public class Presence
extends java.lang.Object
implements java.lang.Runnable
```

CONSTRUCTORS

- *Presence*

```
public Presence( aha.control.SubCluster subCluster,
aha.control.PresenceMovement presenceMovement )
```

METHODS

- *isPresent*

```
public synchronized boolean isPresent( )
```

 - **Usage**
* true = someone is present false = no one is present
 - **Returns** -

- *run*

```
public void run( )
```
- *setGraphActive*

```
public void setGraphActive( boolean graphActive )
```

6.2.20 CLASS PresenceLight

represents a number of daylight input variables.

DECLARATION

```
public class PresenceLight
extends aha.control.InputGroup
```

CONSTRUCTORS

- *PresenceLight*

```
public PresenceLight( javax.agent.service.directory.AgentDescription
sender, javax.agent.Locator senderLocator )
```

METHODS

- *getMaxValue*

```
public double getMaxValue( )
```

 - **Usage**
* get the maximum value of all presence detectors available. if there is only one presence detector available then this method has no effect (just returns the value for the single available presence detector)

- *getPresenceLightValues*

```
public Collection getPresenceLightValues( )
```

– **Usage**

* get all values of all available presence detector light network variables

– **Returns** -

6.2.21 CLASS PresenceMovement

Represents a number of HVAC input values.

DECLARATION

```
public class PresenceMovement
extends aha.control.InputGroup
```

CONSTRUCTORS

- *PresenceMovement*

```
public PresenceMovement( javax.agent.service.directory.AgentDescription
sender, javax.agent.Locator senderLocator )
```

METHODS

- *getPresenceMovementValueMax*

```
public String getPresenceMovementValueMax( )
```

– **Usage**

* get max value if there are several presence detectors present

- *getPresenceMovementValues*

```
public Collection getPresenceMovementValues( )
```

– **Usage**

* get all values of all available presence detector movement network variables

– **Returns** -

6.2.22 CLASS ReinforcementGeneratorBlinds

global reinforcement generator; continuously evaluates a goal function and punishes/rewards the system by feeding data samples to it for learning.

This class runs in it's own thread. It continuously (every cycle) checks the current inputs and outputs of the system and sends a punishment to the learning fuzzy logic controller if there are wrong outputs. The ReinforcementGenerated pursues it's own goals like saving energy that aren't necessarily the interests of other users of the system (like humans). As such it only punishes in situations that are unsuitable for the

DECLARATION

```
public class ReinforcementGeneratorBlinds
extends java.lang.Thread
```

CONSTRUCTORS

- *ReinforcementGeneratorBlinds*

```
public ReinforcementGeneratorBlinds( aha.control.LearningUnitBlinds
learningUnit, aha.control.Presence presence )
```

METHODS

- *run*

```
public void run( )
```
- *stopThisThread*

```
public void stopThisThread( )
```
- *work*

```
public void work( )
```

6.2.23 CLASS ReinforcementGeneratorLight

global reinforcement generator; continuously evaluates a goal function and punishes/rewards the system by feeding data samples to it for learning.

This class runs in it's own thread. It continuously (every cycle) checks the current inputs and outputs of the system and sends a punishment to the learning fuzzy logic controller if there are wrong outputs. The ReinforcementGenerated pursues it's own goals like saving energy that aren't necessarily the interests of other users of the system (like humans). As such it only punishes in situations that are unsuitable for the goal of saving energy but not in case of situations that may be unsuitable for users of the building.

DECLARATION

```
public class ReinforcementGeneratorLight
extends java.lang.Thread
```

CONSTRUCTORS

- *ReinforcementGeneratorLight*

```
public ReinforcementGeneratorLight( aha.control.LearningUnitLight
learningUnit, aha.control.Presence presence )
```

METHODS

- *run*
public void run()
- *stopThisThread*
public void stopThisThread()
- *work*
public void work()

6.2.24 CLASS Room

general interface to a room. encapsulates all room-specific processing. Each room is related to an arbitrary number of network variables that represent blind controlers, lights, switches and presence detectors.

DECLARATION

```
public class Room
extends AbleObject
```

FIELDS

- public static int LIGHT1
—
- public static int LIGHT2
—
- public static int UNDEFINED
—
- public static int ON
—
- public static int OFF
—
- public static int BLIND_UP
—
- public static int BLIND_DOWN
—
- public static int BLIND_STOP

- public static int BLIND_CHANGE
–
- public static int LIGHT_ON
–
- public static int LIGHT_OFF
–

CONSTRUCTORS

- *Room*
public **Room**(aha.control.RoomControlAgent roomControlAgent,
javax.agent.service.directory.AgentDescription sender, javax.agent.Locator
senderLocator)

METHODS

- *changeRoom*
public void **changeRoom**(java.lang.String newRoom)
– **Usage**
* assign new id to this room

- *distributeMessage*
public void **distributeMessage**(aha.framework.AHAMessage msg)

- *getClusterNr*
public int **getClusterNr**()
– **Usage**
* Getter for property ClusterNr.
– **Returns** - Value of property ClusterNr.

- *getRoomControlAgent*
public RoomControlAgent **getRoomControlAgent**()
– **Usage**
* returns the roomcontrolagent to which this instance of room is associated

- *getWeatherVariables*
public Weather **getWeatherVariables**()
– **Usage**
* Getter for property weatherVariables.
– **Returns** - Value of property weatherVariables.

- *handleAbleEvent*

– **Usage**

* for compatibility reasons

• *init*

public void **init**()

• *makePersistent*

public void **makePersistent**()

– **Usage**

* make state persistent, stop everything for exit of agent.

• *process*

public void **process**()

– **Usage**

* start inferencing for all subclusters

• *receive*

public void **receive**(aha.framework.AHAMessage msg)

• *removeRoom*

public void **removeRoom**()

• *reset*

public void **reset**()

• *setClusterNr*

public void **setClusterNr**(int clusterNr)

– **Usage**

* Setter for property ClusterNr.

– **Parameters**

* **ClusterNr** - New value of property ClusterNr.

• *setWeatherVariables*

public void **setWeatherVariables**(aha.control.Weather weatherVariables)

– **Usage**

* Setter for property weatherVariables.

– **Parameters**

* **weatherVariables** - New value of property weatherVariables.

• *stopThreads*

public void **stopThreads**()

– **Usage**

* stop all threads

• *updateVariableValues*

public void **updateVariableValues**()

– **Usage**

6.2.25 CLASS RoomControlAgent

Agent that is responsible to control one specific room. One instance of this agent just controls one room. This agent processes all inputs to one room and takes decisions.

DECLARATION

```
public class RoomControlAgent
extends aha.framework.AHAAgent
```

CONSTRUCTORS

- *RoomControlAgent*
public **RoomControlAgent**()

METHODS

- *handleAbleEvent*
public void **handleAbleEvent**(AbleEvent **theAbleEvent**)
– **Usage**
* events received from able are handed over to this method by able
- *init*
public void **init**()
- *init*
public void **init**(java.lang.Object **initParams**)
- *processTimerEvent*
public void **processTimerEvent**()
- *quitAgent*
public void **quitAgent**(com.ibm.able.jas.AgentLifecycleService **thePrincipal**)
- *receiveMessage*
public void **receiveMessage**(javax.agent.TransportMessage **theTransportMessage**)
- *setActive*
public void **setActive**(boolean **a**)
- *start*
public void **start**()

6.2.26 CLASS SubCluster

DECLARATION

```
public class SubCluster
extends java.lang.Object
```

CONSTRUCTORS

- *SubCluster*

```
public SubCluster( javax.agent.service.directory.AgentDescription sender,
  javax.agent.Locator senderLocator, int clusterNr, aha.control.Weather
  weather )
```

METHODS

- *distributeMessage*

```
public void distributeMessage( aha.framework.AHAMessage msg )
```

 - **Usage**
 * forward this message (broadcast) to all learning units. every learning unit decides if it is interested in the message.

- *init*

```
public void init( )
```

 - **Usage**
 * init method that requests structure from the structureagent related to this subcluster. please not that this method can NOT be called from the constructor because this would cause a deadlock.

- *makePersistent*

```
public void makePersistent( )
```

 - **Usage**
 * make all associated learning units makePersistent. After this method has been called the learning units can no longer be used (must be instanciated again).

- *process*

```
public void process( )
```

 - **Usage**
 * start inferencing process of all associated learning units

- *receive*

```
public void receive( aha.framework.AHAMessage msg )
```

 - **Usage**
 * process received messages and forward all messages to the learning units that aren't processed in this SubCluster

- *setStateChanged*

```
public void setStateChanged( boolean state )
```

 - **Usage**
 - * notify all LearningUnits about a StateChange

- *startThreads*

```
public void startThreads( )
```

 - **Usage**
 - * Start all threads.

- *stopThreads*

```
public void stopThreads( )
```

 - **Usage**
 - * stop all threads of this SubCluster and all associated learning units

6.2.27 CLASS TimeSeriesPlotter

plotting time series (with one data set). The TimePeriod is 1 Millisecond (maximum 1 data value per millisecond).

DECLARATION

```
public class TimeSeriesPlotter
extends ApplicationFrame
implements java.awt.event.ActionListener
```

CONSTRUCTORS

- *TimeSeriesPlotter*

```
public TimeSeriesPlotter( java.lang.String title, java.lang.String
valueDesc, java.lang.String xDesc, java.lang.String yDesc, double
yRangeLower, double yRangeUpper, double fixedRange )
```

 - **Parameters**
 - * **title** - Title of the Plot/Chart
 - * **valueDesc** - Description of the data
 - * **xDesc** - x axis description
 - * **yDesc** - y axis description
 - * **yRangeLower** - lower limit of the y axis
 - * **yRangeUpper** - upper limit of the y axis
 - * **fixedRange** - in milliseconds, range of the x axis

METHODS

-
- *actionPerformed*
 public void **actionPerformed**(java.awt.event.ActionEvent e)
 – **Parameters**
 * e -

 - *addNewDataRecord*
 public void **addNewDataRecord**(double value)
 – **Usage**
 * adds a new data record. this record is automatically timestamped with the actual time that this method is called
 – **Parameters**
 * value - double

 - *close*
 public void **close**()

6.2.28 CLASS TimeSeriesPlotterExtended

plotting time series (with multiple data sets).

DECLARATION

```
public class TimeSeriesPlotterExtended
extends ApplicationFrame
implements java.awt.event.ActionListener
```

CONSTRUCTORS

-
- *TimeSeriesPlotterExtended*
 public **TimeSeriesPlotterExtended**()

METHODS

-
- *actionPerformed*
 public void **actionPerformed**(java.awt.event.ActionEvent e)
 – **Parameters**
 * e -

 - *addDataPlot*
 public void **addDataPlot**(java.lang.String descr)

- *addNewDataRecord*

```
public void addNewDataRecord( java.lang.String descr, double value )
```

- **Usage**

- * adds a new data record. this record is automatically timestamped with the actual time that this method is called

The parameter descr specifies to which data series this value is added.

- **Parameters**

- * **descr** - Name of this time series
- * **value** - value to add

- *close*

```
public void close( )
```

- *init*

```
public void init( java.lang.String title, java.lang.String xDesc,
java.lang.String yDesc, double yRangeLower, double yRangeUpper,
double fixedRange )
```

- **Parameters**

- * **title** - Title of the Plot/Chart
- * **xDesc** - x axis description
- * **yDesc** - y axis description
- * **yRangeLower** - lower limit of the y axis
- * **yRangeUpper** - upper limit of the y axis
- * **fixedRange** - in milliseconds, range of the x axis

- *removeOldDataRecords*

```
public void removeOldDataRecords( java.lang.String descr, int maxCount
)
```

- **Usage**

- * removes old data records from a time series.

- **Parameters**

- * **descr** - Name of this time series
- * **maxCount** - max number of entries in this timeseries. counted from the latest entry towards the oldest entry.

6.2.29 CLASS Weather

Represents all data of global sesonsors (not related to a room, outside weather sensors).

DECLARATION

```
public class Weather
extends aha.control.InputGroup
```

FIELDS

- public static final int ORIENTATION_NORTH
–
- public static final int ORIENTATION_SOUTH
–
- public static final int ORIENTATION_EAST
–
- public static final int ORIENTATION_WEST
–

CONSTRUCTORS

- *Weather*
public **Weather**(javax.agent.service.directory.AgentDescription sender,
 javax.agent.Locator senderLocator)

METHODS

- *getHumidity*
public String **getHumidity**()
– **Usage**
 * return humidity.
– **Returns** -

- *getIlluminance1*
public String **getIlluminance1**()
– **Usage**
 * return illuminance value 1.
– **Returns** -

- *getIlluminance2*
public String **getIlluminance2**()
– **Usage**
 * return illuminance value 2.
– **Returns** -

- *getRadiation*
public String **getRadiation**()
– **Usage**

-
- *getSunEast*
public String **getSunEast**()
 - **Usage**
 - * return radiation east.
 - **Returns** -
-
- *getSunSouth*
public String **getSunSouth**()
 - **Usage**
 - * return readiation south.
 - **Returns** -
-
- *getSunWest*
public String **getSunWest**()
 - **Usage**
 - * return radiation west.
 - **Returns** -
-
- *getTemp*
public String **getTemp**()
 - **Usage**
 - * Return temperature.
 - **Returns** -
-
- *receive*
public void **receive**(aha.framework.AHAMessage msg)
-
- *requestVariables*
public void **requestVariables**(int clusterNr)
-
- *setOrientation*
public void **setOrientation**(int orientation)
 - **Usage**
 - * set the orientation of this instance. After the orientation is set, **getRadiation()** returns the respective value for this direction (if there is one).

Chapter 7

Package `aha.middleware`

<i>Package Contents</i>	<i>Page</i>
<hr/>	
Classes	
Decision	86
<i>Instances of this class are sent out by the control agent as content of a instance of the class Message.</i>	
Message	88
<i>generic able message (serializable), serves as generic container for values that should be sent arround between different agents.</i>	

7.1 Classes

7.1.1 CLASS Decision

Instances of this class are sent out by the control agent as content of a instance of the class Message. it contains all values associates with a decision. this class furthermore defines all possible decisions of the system as static variables.

DECLARATION

```
public class Decision
extends java.lang.Object
implements java.io.Serializable
```

SERIALIZABLE FIELDS

- private Map parameters
 -
- private int decision
 -
- private int type
 -
- private String roomName
 -

FIELDS

- public static int INVALID
 - invalid/unkown decision taken
- public static int DECISION_BLIND
 - possible type for a message
 - decisions of this type contain only decisions concerning Blind's
- public static int DECISION_BLIND_UP
 - DECISION_BLIND_UP = decision that blinds should go up
- public static int DECISION_BLIND_DOWN
 - DECISION_BLIND_DOWN = decision that blinds should go down

- DECISION_BLIND_CHANGE = change position of blinds (neither 100% up nor 100% down, something between). A parameter supplied with this decision specifies the details of the decision.
- public static int DECISION_LIGHT
 - possible type for a message
 - decisions of this type contain only decisions concerning Light's
- public static int DECISION_LIGHT_ON
 - DECISION_LIGHT_ON = light should be switched on
- public static int DECISION_LIGHT_OFF
 - DECISION_LIGHT_OFF = light should be switched off
- public static int DECISION_RADIATION
 - possible type for a message
 - decisions of this type contain only decisions concerning Radiation's
- public static int DECISION_OUTSIDE_RADIATION
 - DECISION_OUTSIDE_RADIATION: new decision regarding the intensity of the outside radiation (taxonomically)

CONSTRUCTORS

- *Decision*

```
public Decision( java.lang.String roomName, int decision, int type )
```

 - **Usage**
 - * Creates a new instance of Decision.
 - **Parameters**
 - * **roomName** - to which room does this decision belong to
 - * **type** - type of decision (defined as static int's in this class, other values are invalid.
 - * **decision** - decision taken. All possible decisions are defined as static fields of this class.

METHODS

- *addParameter*

```
public void addParameter( java.lang.String name, java.lang.String value )
```

 - **Usage**
 - * adds a parameter to this decision.
 - **Parameters**
 - * **name** - name of the parameter (arbitrary)
 - * **value** - value of this parameter (arbitrary)

- *getDecision*
public int getDecision()
 - **Usage**
 * returns the decision identification code.
 - **Returns** - id of this decision

- *getParameters*
public Map getParameters()
 - **Usage**
 * returns a map of all parameters carried by this decision.
 - **Returns** - map of all parameters, map is empty if there are none.

- *getRoomName*
public String getRoomName()
 - **Usage**
 * Gets the roomName.
 - **Returns** - Returns a String

- *getType*
public int getType()
 - **Usage**
 * returns the type of this decision
 - **Returns** - int (defined as static constants in this class)

7.1.2 CLASS Message

generic able message (serializable), serves as generic container for values that should be sent around between different agents.

DECLARATION

```
public class Message
extends java.lang.Object
implements java.io.Serializable
```

SERIALIZABLE FIELDS

- private Object arg
 –
- private String name
 –
- private int contentQualifier

FIELDS

- public static int MESSAGEID_PROPERTY_CHANGE
 - messageid, used as identification of the AbleEvent containing this message
- public static int MESSAGEID_VARIABLE_UPDATE
 - messageid, used as identification of the AbleEvent containing this message
- public static int MESSAGEID_DECISION
 - messageid, used as identification of the AbleEvent containing this message
- public static int MESSAGEID_LOG
 - messageid, used as identification of the AbleEvent containing this message
- public static int ILLUMINANCE1
 - Content identifier
- public static int ILLUMINANCE2
 - Content identifier
- public static int HUMIDITY
 - Content identifier
- public static int SUNEAST
 - Content identifier
- public static int SUNWEST
 - Content identifier
- public static int SUNSOUTH
 - Content identifier
- public static int TEMPERATURE
 - Content identifier
- public static int PRESENCE_MOVEMENT
 - Content identifier
- public static int LIGHT
 - Content identifier
- public static int BLIND
 - Content identifier
- public static int PRESENCE_LIGHT
 - Content identifier
- public static int DECISION_OBJECT

CONSTRUCTORS

- *Message*

```
public Message( java.lang.String name, java.lang.Object arg )
```

- **Usage**

- * construct a message without a type identifier

- **Parameters**

- * **name** - name of this messages
 - * **arg** - Content of the message

- *Message*

```
public Message( java.lang.String name, java.lang.Object arg, int
contentQualifier )
```

- **Usage**

- * Construct a message with content Qualifier

- **Parameters**

- * **name** - name of this message
 - * **arg** - Content of the message
 - * **contentQualifier** - id identifying the content (arguments) of this message.

METHODS

- *getArgument*

```
public Object getArgument( )
```

- **Usage**

- * return the content of this message.

- **Returns** - content

- *getContentQualifier*

```
public int getContentQualifier( )
```

- **Usage**

- * return the content identifier of this message instance.

The content identifier is a int that is defined as a static field of this class.

- **Returns** - id identifying the content (arguments) of this message.

- *getName*

```
public String getName( )
```

- **Usage**

- * return the name of this message.

- **Returns** - name

- *setContentQualifier*

– **Usage**

- * set content identifier (id).

The content identifier is a int that is defined as a static field of this class.

– **Parameters**

- * `contentQualifier` - id identifying the content (arguments) of this message.

Chapter 8

Package aha.structure

<i>Package Contents</i>	<i>Page</i>
<hr/>	
Classes	
Cluster 93 <i>A value object for clusters.</i>	
ElementInfo 93 <i>Value object to store the element information.</i>	
ElementType 96 <i>class with static constants for the message types</i>	
StructureAgent 97 <i>Delivers structure information about the building to its client agents.</i>	

8.1 Classes

8.1.1 CLASS Cluster

A value object for clusters. Stores its sub clusters.

DECLARATION

```
public class Cluster
extends java.lang.Object
```

CONSTRUCTORS

- *Cluster*
public Cluster()
- *Cluster*
public Cluster(java.util.Collection elements)

METHODS

- *getElements*
public Collection **getElements**()
 - **Usage**
* Returns the _elements.
 - **Returns** - Collection

8.1.2 CLASS ElementInfo

Value object to store the element information. This object is serializable to be able to transport it over RMI.

DECLARATION

```
public class ElementInfo
extends java.lang.Object
implements java.io.Serializable
```

SERIALIZABLE FIELDS

- private Integer _clusterId

- private Integer *_id*
–
- private String *_type*
–
- private String *_displayCName*
–
- private boolean *_effector*
–
- private boolean *_sensor*
–
- private boolean *_feedback*
–
- private String *_info*
–

CONSTRUCTORS

- *ElementInfo*
public **ElementInfo**(int *clusterId*, int *id*, java.lang.String *displayCName*, java.lang.String *type*, boolean *effector*, boolean *sensor*, boolean *feedback*)

METHODS

- *getClusterId*
public Integer **getClusterId**()
– **Usage**
 * Returns the *_clusterId*.
– **Returns** - int

- *getDisplayCName*
public String **getDisplayCName**()
– **Usage**
 * Returns the *_displayCName*.
– **Returns** - String

- *getId*
public Integer **getId**()
– **Usage**

– **Returns** - int

• *getInfo*

public String getInfo()

– **Usage**

* Returns the `_info`.

– **Returns** - String

• *getName*

public String getName()

• *getType*

public String getType()

– **Usage**

* Returns the `_type`.

– **Returns** - String

• *isEffector*

public boolean isEffector()

– **Usage**

* Returns the `_effector`.

– **Returns** - boolean

• *isFeedback*

public boolean isFeedback()

– **Usage**

* Returns the `_feedback`.

– **Returns** - boolean

• *isSensor*

public boolean isSensor()

– **Usage**

* Returns the `_sensor`.

– **Returns** - boolean

• *setClusterId*

public void setClusterId(java.lang.Integer clusterId)

• *setInfo*

public void setInfo(java.lang.String info)

– **Usage**

* Sets the `_info`.

– **Parameters**

* `_info` - The `_info` to set

• *toString*

8.1.3 CLASS **ElementType**

class with static constants for the message types

DECLARATION

```
public class ElementType  
extends java.lang.Object
```

FIELDS

- public static final String LIGHT
–
- public static final String LIGHT_SWITCH
–
- public static final String BLINDS
–
- public static final String BLIND_SWITCH
–
- public static final String WEATHER
–
- public static final String PRESENCE
–
- public static final String DAYLIGHT
–
- public static final String ILLUMINANCE
–
- public static final String ILLUMINANCE2
–
- public static final String HUMIDITY
–
- public static final String RADIATION_EAST
–
- public static final String RADIATION_SOUTH

- public static final String RADIATION_WEST
–
- public static final String TEMPERATURE
–
- public static final String CLUSTER
–
- public static final String INFO
–

CONSTRUCTORS

- *ElementType*
public **ElementType**()

8.1.4 CLASS StructureAgent

Delivers structure information about the building to its client agents. This implementation reads this info from a xml config file. The file is read using the castor data binding framework

DECLARATION

```
public class StructureAgent
extends aha.framework.AHAAgent
```

CONSTRUCTORS

- *StructureAgent*
public **StructureAgent**()
 - **Usage**
* reads the config file. Recursively calls addClusters
 - **See Also**
* java.lang.Object.Object()

METHODS

- *addClusters*
public Collection **addClusters**(int **parentId**, aha.structure.config.Cluster [] **clusters**)
 - **Usage**

– **Parameters**

- * `parentId` - the Id for the parent cluster
- * `clusters` - all the clusters to process, passed from castor

– **Returns** - Collection all the clusters contained in this cluster

- *getElements*

```
public void getElements( java.util.Collection result, java.lang.String
type, java.util.Collection el, java.lang.Boolean doRecursive )
```

– **Usage**

- * looks for elements of the desired type in the provided collection. If param `doRecursive` is true, calls itself recursively if the element is of type cluster

– **Parameters**

- * `result` - the elements found are added to this collection
 - * `type` - the type of message to search for
 - * `el` - the collection in which elements should be searched
 - * `doRecursive` - should it be called recursively on other cluster
-

- *init*

```
public void init( )
```

- *processGetElements*

```
public void processGetElements( aha.framework.AHAMessage msg )
```

– **Usage**

- * processes a new GET_ELEMENTS message. This message does not check the type of the message. Expects it beeing of the correct type

– **Parameters**

- * `msg` - the AHAMessage to process
-

- *receiveMessage*

```
public void receiveMessage( javax.agent.TransportMessage
theTransportMessage )
```

– **Usage**

- * recieves GET_ELEMENTS messages and processes them.

– **Parameters**

- * `theTransportMessage` - the message to process

– **See Also**

- * `javax.agent.service.transport.MessageListener.receiveMessage(TransportMessage)`

Chapter 9

Package aha.bus

<i>Package Contents</i>	<i>Page</i>
<hr/>	
Interfaces	
BusController	100
<i>Interface to abstract the access to different implementations of building buses.</i>	
Classes	
BusAgentJas	101
<i>Responsible for the access to the building bus.</i>	
BusFrontend	102
<i>Frontend for the BusAgent/BusSim.</i>	
PropertyDescription	103
<i>...no description...</i>	

9.1 Interfaces

9.1.1 INTERFACE BusController

Interface to abstract the access to different implementations of building buses.

It makes the assumption that a bus allows the access to a bunch of variables which can be queried and set

DECLARATION

```
public interface BusController
```

METHODS

- *getDescription*

```
public PropertyDescription getDescription( java.lang.String name )
```

 - **Usage**
 - * what the does this method do ???

- *getProperties*

```
public Collection getProperties( java.lang.String regex )
```

 - **Usage**
 - * returns the names of all bus variables matching a certain regular expression
 - **Parameters**
 - * **regex** - Regular expression for the variable names
 - **Returns** - Collection of variable names

- *getProperty*

```
public String getProperty( java.lang.String name )
```

 - **Usage**
 - * gets the value of a bus variable
 - **Parameters**
 - * **name** - the bus variables name

- *registerPropertyChangeListener*

```
public void registerPropertyChangeListener( java.lang.String name,
java.beans.PropertyChangeListener listener )
```

 - **Usage**
 - * registers a listener for a bus variable
 - **Parameters**
 - * **name** - the bus variable name
 - * **listener** - for variable changes

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  name,
java.beans.PropertyChangeListener  listener )
```

 - **Usage**
 - * removes a listener from the list of listeners for a bus variable
 - **Parameters**
 - * **name** - the bus variable name
 - * **listener** - the listener which should be removed

- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.String  value )
```

 - **Usage**
 - * sets the value for a bus variable
 - **Parameters**
 - * **name** - the bus variables name
 - * **value** - the value to be set

9.2 Classes

9.2.1 CLASS BusAgentJas

Responsible for the access to the building bus. Provides methods to get variable values for the home automation devices and subscribing for variable changes. Tries to mimic the JavaBeans naming convention.

The implementation for the businterface is read from the config file

DECLARATION

```
public class BusAgentJas
extends aha.framework.AHAAgent
```

CONSTRUCTORS

- *BusAgentJas*

```
public BusAgentJas( )
```

METHODS

- *getLogQueue*

```
public Collection getLogQueue( )
```

 - **Usage**

– **Returns** - Collection of LogArgs Objects which should be logged

– **See Also**

* aha.history.LogArgs

- *init*

```
public void init( )
```

– **Usage**

* Initializes the Agent.
Connects to BusController, reads from Config file which BusController implementation should be used.

- *quitAgent*

```
public void quitAgent( com.ibm.able.jas.AgentLifecycleService thePrincipal )
```

- *receiveMessage*

```
public void receiveMessage( javax.agent.TransportMessage theTransportMessage )
```

– **Usage**

* processes messages

– **Parameters**

* msg -

9.2.2 CLASS BusFrontend

Frontend for the BusAgent/BusSim. This class handles all common tasks (messaging) for all sorts of available BusAgents. It processes incoming messages und sends back answers. This class itself does no processing, it does just handle the messages and their encoding. All processing is delegated to the specific Controller which must implement the BusController interface.

DECLARATION

```
public class BusFrontend
extends java.lang.Object
implements java.beans.PropertyChangeListener
```

CONSTRUCTORS

- *BusFrontend*

```
public BusFrontend( com.ibm.able.jas.AbleJasDefaultAgent theAgent,
aha.bus.BusController theController,
javax.agent.service.directory.AgentDescription sender, javax.agent.Locator
senderLocator, java.util.HashMap vars, java.util.HashMap rvars )
```

– **Usage**

METHODS

- *getVariableId*
public String getVariableId(java.lang.String varName)
- *propertyChange*
public void propertyChange(java.beans.PropertyChangeEvent evt)
 - **Usage**
 - * forwards changes received from BusController instance to MessageDistributor
- *receiveMessage*
public void receiveMessage(aha.framework.AHAMessage msg)
 - **Usage**
 - * processes busagent specific messages: TOPIC_CREATE, TOPIC_DESTROY, QUERY_VARIABLES, GET_PROPERTY
 - **Parameters**
 - * msg -
- *unregister*
public void unregister()

9.2.3 CLASS PropertyDescription

DECLARATION

```
public class PropertyDescription
extends java.lang.Object
```

CONSTRUCTORS

- *PropertyDescription*
public PropertyDescription()

Chapter 10

Package aha.sim

<i>Package Contents</i>	<i>Page</i>
<hr/>	
Interfaces	
Simulator	105
<i>Interface to abstract the access to different implementations of building buses.</i>	
Classes	
BusSim	105
<i>Simple simulator for an aha bus, variable names are just random int values and should not be considered having any meaning.</i>	
BusSim.Controller	106
<i>Does the simulation of a field bus.</i>	
Connection	109
<i>Models the connection between variables</i>	
FloorView	110
<i>Swing component which displays a building's floor.</i>	
ImageUtilities	112
<i>Helper class to load and display images with Swing</i>	
Mapping	112
<i>Models a transferfunction or mapping for a connection</i>	
RoomView	114
<i>shows a window with a list of all variables in a room</i>	
SimulatorImpl	115
<i>Simulator Agent is responsible for the controll of the whole simulation It start, stops and pauses the simulation.</i>	
SwingSimulator	116
<i>Implements the Swing UI to display the state of a building/floor.</i>	
Variable	119
<i>...no description...</i>	
VariableTableModel	119
<i>Custom data model for variables to be displayed in a JList.</i>	
WakeUpEvent	121
<i>Stores timed events for a certain timestamp and knows how to send them to their recievers.</i>	

10.1 Interfaces

10.1.1 INTERFACE Simulator

Interface to abstract the access to different implementations of building buses.

It makes the assumption that a bus allows the access to a bunch of variables which can be queried and set

DECLARATION

```
public interface Simulator
implements java.rmi.Remote
```

METHODS

- *getTime*
public long getTime()
- *loadSimulation*
public void loadSimulation(java.lang.String cfgFile)
- *run*
public void run()
- *setDelay*
public void setDelay(int delay)
- *stop*
public void stop()

10.2 Classes

10.2.1 CLASS BusSim

Simple simulator for an aha bus, variable names are just random int values and should not be considered having any meaning. This class does only the initialization work, the actual meat is in its inner class Controller

DECLARATION

```
public class BusSim
extends aha.framework.AHAAgent
```

FIELDS

- public static final char DELIMITER

CONSTRUCTORS

- *BusSim*
public **BusSim**()

METHODS

- *getLogQueue*
public Collection **getLogQueue**()
 - **Usage**
* returns and empties the Queue of events which have not been logged so far
 - **Returns** - Collection of LogArgs Objects which should be logged
 - **See Also**
* aha.history.LogArgs

- *init*
public void **init**()

- *init*
public void **init**(java.lang.Object **cfg**)

- *quitAgent*
public void **quitAgent**(com.ibm.able.jas.AgentLifecycleService **thePrincipal**)
 - **Usage**
* unregisters all topics it is listening for via frontend
 - **See Also**
* com.ibm.able.jas.AbleJasAgent.quitAgent(AgentLifecycleService)

- *receiveMessage*
public void **receiveMessage**(javax.agent.TransportMessage **theTransportMessage**)
 - **Usage**
* processes the incoming messages
 - **Parameters**
* msg -

10.2.2 CLASS BusSim.Controller

Does the simulation of a field bus. Checks all the connections on updates, uses Frontend to interface with the messaging system.

DECLARATION

```
public class BusSim.Controller
extends java.lang.Object
implements aha.bus.BusController
```

CONSTRUCTORS

- *BusSim.Controller*
public **BusSim.Controller**()

METHODS

- *addFilter*
public void **addFilter**(java.lang.String **name**, java.lang.String **source**, double **value**)
 - **Usage**
* Adds a new filter to a variable
 - **Parameters**
* **name** - of the variable which gets a new filter
* **source** - variable for this filter
* **value** - by which the source is lessend (factor)

- *doFilter*
public void **doFilter**(java.lang.String **name**)
 - **Usage**
* executes the filters for a certain variable
 - **Parameters**
* **name** - of the variable for which the filters should be executed

- *doUpdate*
public void **doUpdate**(java.lang.String **name**, java.lang.String **newValue**, java.lang.String **oldval**)
 - **Usage**
* applies the transferfunctions for a certain variable. This method is also used for the update action in connection mappings.
 - **Parameters**
* **name** - variable name which we should apply the transferfunction for
* **newValue** - the new variable value
* **oldval** - the old variable value

- *getDescription*
public PropertyDescription **getDescription**(java.lang.String **name**)

- *getDevices*
 public Collection **getDevices**()
 – **Usage**
 * returns collection of all devices found in the config file
 – **Returns** - Collection

- *getProperties*
 public Collection **getProperties**(java.lang.String regex)
 – **Usage**
 * returns a collection of variables whichs names match a certain regular expression
 – **See Also**
 * aha.bus.BusController.getProperties(String)

- *getProperty*
 public String **getProperty**(java.lang.String name)
 – **Usage**
 * gets the value of a certain variable
 – **See Also**
 * aha.bus.BusController.getProperty(String)

- *init*
 public void **init**(java.lang.String cfg)
 – **Usage**
 * loads the configuration file
 – **Parameters**
 * cfg -

- *registerPropertyChangeListener*
 public void **registerPropertyChangeListener**(java.lang.String name,
 java.beans.PropertyChangeListener listener)
 – **See Also**
 * aha.bus.BusController.registerPropertyChangeListener(String,
 PropertyChangeListener)

- *removePropertyChangeListener*
 public void **removePropertyChangeListener**(java.lang.String name,
 java.beans.PropertyChangeListener listener)
 – **See Also**
 * aha.bus.BusController.removePropertyChangeListener(String,
 PropertyChangeListener)

- *setProperty*
 public void **setProperty**(java.lang.String name, java.lang.String svalue)

- **Usage**
 - * sets the value of a field bus variable
- **See Also**
 - * `aha.bus.BusController.setProperty(String, String)`

10.2.3 CLASS Connection

Models the connection between variables

DECLARATION

```
public class Connection
extends java.lang.Object
```

CONSTRUCTORS

- *Connection*

```
public Connection( aha.sim.BusSim.Controller sim, java.lang.String target
)
```

METHODS

- *addMapping*

```
public void addMapping( aha.sim.Mapping m )
```

 - **Usage**
 - * adds another mapping to this connection
 - **Parameters**
 - * `m` - the mapping

- *doUpdate*

```
public void doUpdate( java.lang.String oldValue, java.lang.String value )
```

 - **Usage**
 - * update the targets for this connections. Checks all the mappings if they change something, and applies this change.
 - **Parameters**
 - * `oldValue` - the source variables old value
 - * `value` - the source variables new value

- *init*

```
public void init( )
```

 - **Usage**

10.2.4 CLASS FloorView

Swing component which displays a building's floor.

DECLARATION

```
public class FloorView
extends javax.swing.JPanel
implements java.awt.event.MouseListener
```

CONSTRUCTORS

- *FloorView*
public FloorView(aha.sim.SwingSimulator sim)
 - **Usage**
 - * initializes the component . Loads the images for light, blind, person and floor.
Reads the config file for the locations of the rooms.
 - **Parameters**
 - * **sim** -

METHODS

- *drawBlind*
public void drawBlind(aha.structure.ElementInfo ei, java.lang.String value)
 - **Usage**
 - * adds and removes a room to/from the list of to be displayed blind icons
 - **Parameters**
 - * **ei** - ElementInfo of the blind
 - * **value** - of the blind

- *drawLight*
public void drawLight(aha.structure.ElementInfo ei, java.lang.String value)
 - **Usage**
 - * adds and removes a room to/from the list of to be displayed light icons
 - **Parameters**
 - * **ei** - ElementInfo of the light
 - * **value** - of the light

- *mouseClicked*
public void mouseClicked(java.awt.event.MouseEvent e)

- * check which room was clicked on and shows a room view
- **See Also**
- * java.awt.event.MouseListener.mouseClicked(java.awt.event.MouseEvent)

- *mouseEntered*
- public void **mouseEntered**(java.awt.event.MouseEvent e)
- **See Also**
- * java.awt.event.MouseListener.mouseEntered(java.awt.event.MouseEvent)

- *mouseExited*
- public void **mouseExited**(java.awt.event.MouseEvent e)
- **See Also**
- * java.awt.event.MouseListener.mouseExited(java.awt.event.MouseEvent)

- *mousePressed*
- public void **mousePressed**(java.awt.event.MouseEvent e)
- **See Also**
- * java.awt.event.MouseListener.mousePressed(java.awt.event.MouseEvent)

- *mouseReleased*
- public void **mouseReleased**(java.awt.event.MouseEvent e)
- **See Also**
- * java.awt.event.MouseListener.mouseReleased(java.awt.event.MouseEvent)

- *paintComponent*
- public void **paintComponent**(java.awt.Graphics g)
- **Usage**
- * paints the floor image with all the icons on it
- **See Also**
- * javax.swing.JComponent.paintComponent(java.awt.Graphics)

- *personEntered*
- public void **personEntered**(java.lang.Integer **personId**, java.lang.Integer **clusterId**)
- **Usage**
- * adds room the list where a person icon should be displayed
- **Parameters**
- * **personId** -
- * **clusterId** -

- *personLeft*
- public void **personLeft**(java.lang.Integer **personId**, java.lang.Integer **clusterId**)
- **Usage**
- * removes room of list where we should display the person icon
- **Parameters**
- * **personId**

10.2.5 CLASS ImageUtilities

Helper class to load and display images with Swing

DECLARATION

```
public class ImageUtilities
extends java.lang.Object
```

CONSTRUCTORS

- *ImageUtilities*
public **ImageUtilities**()

METHODS

- *getBufferedImage*
public static BufferedImage **getBufferedImage**(java.lang.String **imageFile**,
java.awt.Component **c**)

– **Usage**
* Create Image from a file, then turn that into a BufferedImage.

- *waitForImage*
public static boolean **waitForImage**(java.awt.Image **image**,
java.awt.Component **c**)

– **Usage**
* Take an Image associated with a file, and wait until it is done loading. Just a simple application of MediaTracker. If you are loading multiple images, don't use this consecutive times; instead use the version that takes an array of images.

- *waitForImages*
public static boolean **waitForImages**(java.awt.Image [] **images**,
java.awt.Component **c**)

– **Usage**
* Take some Images associated with files, and wait until they are done loading. Just a simple application of MediaTracker.

10.2.6 CLASS Mapping

Models a transferfunction or mapping for a connection

DECLARATION

```
public class Mapping
extends java.lang.Object
```

FIELDS

- public static final int EQUAL
–
- public static final int GREATER
–
- public static final int LESSER
–
- public static final int CONTAINS
–
- public static final int ALWAYS
–
- public static final int SET
–
- public static final int ADD
–
- public static final int SUB
–
- public static final int ADDDIFF
–
- public static final int FILTER
–
- public static final int UPDATE
–

CONSTRUCTORS

- *Mapping*
public **M**apping(aha.sim.BusSim.Controller bus, java.lang.String cmp,
java.lang.String value, java.lang.String action, java.lang.String

METHODS

- *doUpdate*
public String doUpdate(java.lang.String oldValue, java.lang.String val,
java.lang.String target)
 - **Usage**
 - * checks if this mapping changes the value of target
 - **Parameters**
 - * oldValue -
 - * val -
 - * target -
 - **Returns** - String

10.2.7 CLASS RoomView

shows a window with a list of all variables in a room

DECLARATION

```
public class RoomView
extends javax.swing.JFrame
```

SERIALIZABLE FIELDS

- private String room
 -
- private Integer clusterId
 -
- private JTable variableTable
 -
- private Vector columnNames
 -

CONSTRUCTORS

- *RoomView*
public **RoomView**(java.lang.Integer clusterId, java.lang.String
displayName, aha.sim.SwingSimulator sim)
 - **Usage**

– **Parameters**

- * clusterId -
- * displayName -
- * sim -

10.2.8 CLASS SimulatorImpl

Simulator Agent is responsible for the controll of the whole simulation

It start, stops and pauses the simulation. It controls the simulation time and manages the timed events

DECLARATION

```
public class SimulatorImpl
extends aha.framework.AHAAgent
```

FIELDS

- public static int DEFAULT_DELAY

–

CONSTRUCTORS

- *SimulatorImpl*
public **SimulatorImpl**()

METHODS

- *init*
public void **init**()
 - **Usage**
 - * initializes the simulation
 - **See Also**
 - * com.ibm.able.AbleBean.init()
 - *processTimerEvent*
public void **processTimerEvent**()
 - **Usage**
 - * advances the simulation one second
 - **See Also**
 - * com.ibm.able.AbleEventQueueProcessor.processTimerEvent()
-

- *receiveMessage*

```
public void receiveMessage( javax.agent.TransportMessage
theTransportMessage )
```

 - **Usage**
 - * receives simulation control messages and new timed events
 - **See Also**
 - *

```
javax.agent.service.transport.MessageListener.receiveMessage(TransportMessage)
```

- *run*

```
public void run( )
```

 - **Usage**
 - * Start the Simulation Start all the agents using Able Agent Lifecycle Service

- *setDelay*

```
public void setDelay( int delay )
```

 - **Usage**
 - * sets the simulation delay
 - **Parameters**
 - * delay -
 - **Exceptions**
 - * java.rmi.RemoteException -

- *stop*

```
public void stop( )
```

 - **Usage**
 - * Stop the Simulation Stop all the agents using Able Agent Lifecycle Service

- *triggerEvent*

```
public void triggerEvent( long seconds,
javax.agent.service.directory.AgentDescription sender, javax.agent.Locator
locator, javax.agent.service.directory.AgentDescription agent,
aha.framework.AHAMessage eventName )
```

 - **Usage**
 - * adds a new timed event to the event queue
 - **Parameters**
 - * seconds -
 - * sender -
 - * locator -
 - * agent -
 - * eventName -

DECLARATION

```
public class SwingSimulator
extends aha.framework.AHAAgent
```

CONSTRUCTORS

- *SwingSimulator*
public **SwingSimulator**()
 - **Usage**
 - * constructs the user interface
 - **See Also**
 - * java.lang.Object.Object()

METHODS

- *createComponents*
public Component **createComponents**()
 - **Usage**
 - * creates the components of the user interface
 - **Returns** - Component

- *CreateMenuItem*
public JMenuItem **CreateMenuItem**(javax.swing.JMenu menu, int iType, java.lang.String sText, javax.swing.ImageIcon image, int acceleratorKey, java.lang.String sToolTip, java.awt.event.ActionListener al)
 - **Usage**
 - * helper method to create a menu item.
 - **Parameters**
 - * menu -
 - * iType -
 - * sText -
 - * image -
 - * acceleratorKey -
 - * sToolTip -
 - * al -
 - **Returns** - JMenuItem

- *createWeatherView*
public void **createWeatherView**(aha.structure.ElementInfo ei)
 - **Usage**
 - * creates the list of weather variables
 - **Parameters**

- *init*

public void **init**()

- **Usage**

- * initializes the agent and the UI

- **See Also**

- * com.ibm.able.AbleBean.init()

- *main*

public static void **main**(java.lang.String [] args)

- **Usage**

- * main method to start the ui

- **See Also**

- * com.ibm.able.agents.AbleDefaultAgent.main(String[])

- *quit*

public void **quit**()

- **Usage**

- * unregister all variable interests and quits the application

- *receiveMessage*

public void **receiveMessage**(javax.agent.TransportMessage
theTransportMessage)

- **Usage**

- * receives the messages the ui needs to keep its state up to date.

- **See Also**

- *
javax.agent.service.transport.MessageListener.receiveMessage(TransportMessage)

- *registerTopVariables*

public void **registerTopVariables**()

- **Usage**

- * Queries for the variables we are interested in and registers the interest for them.

- *registerVariableViewer*

public void **registerVariableViewer**(java.lang.String displayCName,
java.lang.Integer clusterId, aha.sim.VariableTableModel md)

- **Usage**

- * register interest for some variables for a VariableTableModel

- **Parameters**

- * displayCName - the displayName for this roomViewer
 - * clusterId - the cluster we are interested in
 - * md - the model which want's to be informed of updates

- *sendSetProperty*

– **Usage**

* sends a property update to busAgent

– **Parameters**

* var -

* val -

• *timerChanged*

public void **timerChanged**(int **time**)

– **See Also**

*

java.awt.event.ActionListener.actionPerformed(java.awt.event.ActionEvent)

10.2.10 CLASS Variable

DECLARATION

```
public class Variable
extends java.lang.Object
```

CONSTRUCTORS

• *Variable*

public **Variable**()

10.2.11 CLASS VariableTableModel

Custom data model for variables to be displayed in a JList.

DECLARATION

```
public class VariableTableModel
extends javax.swing.table.AbstractTableModel
```

SERIALIZABLE FIELDS

• private TreeSet *_variables*

–

• private HashMap *_values*

–

-
- private HashMap _clusters
-
- private Integer _cluster
-
- private SwingSimulator _swingSimulator
-

CONSTRUCTORS

- *VariableTableModel*
public **VariableTableModel**(aha.sim.SwingSimulator **sim**, java.lang.Integer **cluster**, java.lang.String **displayCName**)

METHODS

- *getColumnCount*
public int **getColumnCount**()
– See Also
* javax.swing.table.TableModel.getColumnCount()

- *getColumnName*
public String **getColumnName**(int **col**)
– See Also
* javax.swing.table.TableModel.getColumnName(int)

- *getRowCount*
public int **getRowCount**()
– See Also
* javax.swing.table.TableModel.getRowCount()

- *getValueAt*
public Object **getValueAt**(int **rowIndex**, int **columnIndex**)
– See Also
* javax.swing.table.TableModel.getValueAt(int, int)

- *isCellEditable*
public boolean **isCellEditable**(int **row**, int **col**)
– See Also
* javax.swing.table.TableModel.isCellEditable(int, int)

- *setData*

```
public void setData( java.lang.Integer  clusterId, java.lang.String  id,
java.lang.String  type, java.lang.String  value )
```

 - Usage
 - * adds a new entry for a variable
 - Parameters
 - * clusterId -
 - * id -
 - * type -
 - * value -
-
- *setValueAt*

```
public void setValueAt( java.lang.Object  value, int  row, int  col )
```

 - See Also
 - * javax.swing.table.TableModel.setValueAt(Object, int, int)

10.2.12 CLASS WakeUpEvent

Stores timed events for a certain timestamp and knows how to send them to their receivers.

DECLARATION

```
public class WakeUpEvent
extends java.lang.Object
```

CONSTRUCTORS

- *WakeUpEvent*

```
public WakeUpEvent( javax.agent.service.directory.AgentDescription
sender, javax.agent.Locator  locator,
javax.agent.service.directory.AgentDescription  destination,
aha.framework.AHAMessage  msg )
```

METHODS

- *addEvent*

```
public void addEvent( javax.agent.service.directory.AgentDescription
sender, javax.agent.Locator  locator,
javax.agent.service.directory.AgentDescription  destination,
aha.framework.AHAMessage  msg )
```

 - Usage
 - * adds a new event for this timestamp

- * sender -
- * locator -
- * destination -
- * msg -

- *sendEvents*

public void sendEvents(long time)

- **Usage**

- * send all the events

- **Parameters**

- * time -

Chapter 11

Package aha.test.jas

<i>Package Contents</i>	<i>Page</i>
<hr/>	
Classes	
AHASTarter 124 <i>...no description...</i>	
JASAgent1 124 <i>...no description...</i>	
JASAgent2 125 <i>...no description...</i>	
JASStarter 126 <i>general JAS monitor for the agent directory service and the able lifecycle service</i>	

11.1 Classes

11.1.1 CLASS AHASstarter

DECLARATION

```
public class AHASstarter
extends java.lang.Object
```

CONSTRUCTORS

- *AHASstarter*
public **AHASstarter**()
 - **Usage**
 - * Creates a new instance of JASstarter

METHODS

- *main*
public static void **main**(java.lang.String [] args)
- *start*
public void **start**()

11.1.2 CLASS JASAgent1

DECLARATION

```
public class JASAgent1
extends com.ibm.able.jas.AbleJasDefaultAgent
```

CONSTRUCTORS

- *JASAgent1*
public **JASAgent1**()
 - **Usage**
 - * Creates a new instance of JASAgent1

METHODS

- *debugOut*
public void debugOut()
- *init*
public void init()
- *process*
public Object process(java.lang.Object theArg)
- *quitAgent*
public void quitAgent(com.ibm.able.jas.AgentLifecycleService thePrincipal)
- *receiveMessage*
public void receiveMessage(javax.agent.TransportMessage theTransportMessage)
- *register*
public void register()

11.1.3 CLASS JASAgent2

DECLARATION

```
public class JASAgent2
extends com.ibm.able.jas.AbleJasDefaultAgent
```

CONSTRUCTORS

- *JASAgent2*
public **JASAgent2**()
– **Usage**
* Creates a new instance of JASAgent2

METHODS

- *checkDirectory*
public void checkDirectory()
- *init*
public void init()
- *send*
public void send()

11.1.4 CLASS JASStarter

general JAS monitor for the agent directory service and the able lifecycle service

DECLARATION

```
public class JASStarter
extends java.lang.Object
```

CONSTRUCTORS

- *JASStarter*
public **JASStarter**(java.lang.String host)
 - **Usage**
 - * Creates a new instance of JASStarter

- *JASStarter*
public **JASStarter**(java.lang.String host, java.lang.String classname, java.lang.String hostAlias)

METHODS

- *main*
public static void **main**(java.lang.String [] args)
- *start*
public void **start**(java.lang.String agentClass, java.lang.String host)
 - **Usage**
 - * start a agent (lifecycle service)
 - **Parameters**
 - * **agentClass** - full qualified class name of a class that inherits from JasDefaultAgent
 - * **host** - host alias (as configured in JAS)
 - **Exceptions**
 - * java.lang.Exception -

Chapter 12

Package aha.messaging

<i>Package Contents</i>	<i>Page</i>
<hr/>	
Classes	
Constants	128
<i>Defines various constants for messaging.</i>	
MessageDistributionAgentJas	130
<i>Distributes messages asynchronously and interest based.</i>	
MessagingSupport	131
<i>Support for messaging.</i>	

12.1 Classes

12.1.1 CLASS Constants

Defines various constants for messaging.

DECLARATION

```
public final class Constants
extends java.lang.Object
```

FIELDS

- public static final String SUBSCRIBE_INTEREST
 - message type
- public static final String DESUBSCRIBE_INTEREST
 - message type
- public static final String DISTRIBUTE_MESSAGE
 - message type
- public static final String TIMED_EVENT
 - message type
- public static final String SET_PROPERTY
 - message type
- public static final String GET_PROPERTY
 - message type
- public static final String TOPIC_CREATE
 - Topic Name for Distribution Agent
- public static final String TOPIC_REMOVE
 - Topic Name for Distribution Agent
- public static final String QUERY_VARIABLES
 - message type
- public static final String QUERY_TOPICS
 - message type
- public static final String QUERY_VARIABLES_RESULT
 - message type

- message type
- public static final String MESSAGE_TEST_1
 - message type
- public static final String VARIABLE_UPDATE
 - message type
- public static final String GET_ELEMENTS
 - message type
- public static final String GET_ELEMENTS_RESULT
 - message type
- public static final String GET_CLUSTERS
 - message type
- public static final String GET_CLUSTERS_RESULT
 - message type
- public static final String PERSON_ENTERED
 - message type
- public static final String PERSON_LEFT
 - message type
- public static final String UPDATE_PERSONS
 - message type
- public static final String ANNOUNCE_RULE
 - message type
- public static final String TOPIC_GLOBAL
 - Topic Name for Distribution Agent
- public static final String TOPIC_RULES
 - Topic Name for Distribution Agent
- public static final String TOPIC_PREFIX_VARIABLE
 - Topic Name for Distribution Agent
- public static final String SIMULATION_RUN
 - message type
- public static final String SIMULATION_STOP
 - message type
- public static final String SIMULATION_SET_DELAY
 - message type
- public static final String TIME

 CONSTRUCTORS

- *Constants*
public Constants()

12.1.2 CLASS MessageDistributionAgentJas

Distributes messages asynchronously and interest based.

supported type of operations: destinations: global, individual or group (=interests). A group consists of all agents that are interested in a certain type of date, qualified as a variable string (fully qualified variable, can be a normal variable or also more high level data like decisions)

every agent that wants to receive messages from the messagedistributionAgent needs to subscribe itself to all groups that it is interest in. subscription/unsubscription works with special messages.

agents that aren't interest in any special group but want to receive global messages need to register with the MessageDistributionAgent as well (subscribe to the group global).

This Agent uses a threadpool of max size THREAD_POOL_SIZE to distribute messages fully asynchronously. To prevent loss of efficiency due to creation times of new threads a large number of threads is used in the pool.

 DECLARATION

```
public class MessageDistributionAgentJas
extends aha.framework.AHAAgent
```

 CONSTRUCTORS

- *MessageDistributionAgentJas*
public MessageDistributionAgentJas()
 - **Usage**
 - * Instanciate a new MessageDistributionAgent.

 METHODS

- *init*
public void init()
 - **Usage**
 - * initialize the agent with default parameters.
 - **Exceptions**
 - * java.rmi.RemoteException

- *init*

```
public void init( java.lang.Object initParams )
```

 - **Usage**
 - * initialize the agent with parameters.
 - **Parameters**
 - * **initParams** -
 - **Exceptions**
 - * `java.rmi.RemoteException` -

- *main*

```
public static void main( java.lang.String [] args )
```

 - **Usage**
 - * for testing.
 - **Parameters**
 - * **args** -

- *receiveMessage*

```
public void receiveMessage( javax.agent.TransportMessage theTransportMessage )
```

 - **Usage**
 - * process incoming message.
 - **Parameters**
 - * **theTransportMessage** -

- *sendMessageToIndividual*

```
public void sendMessageToIndividual(  

  javax.agent.service.directory.AgentDescription receiver, java.lang.Object  

content )
```

 - **Usage**
 - * send a message to an individual agent.
 - **Parameters**
 - * **receiver** -
 - * **content** -

- *sendMessageToTopic*

```
public void sendMessageToTopic( java.lang.String topic, java.lang.Object  

content )
```

 - **Usage**
 - * send a message to a topic.
 - **Parameters**
 - * **topic** -
 - * **content** -

DECLARATION

```
public class MessagingSupport
extends java.lang.Object
```

CONSTRUCTORS

- *MessagingSupport*
public **MessagingSupport**()

METHODS

- *registerInterest*
public static boolean **registerInterest**(
javax.agent.service.directory.AgentDescription sender,
javax.agent.service.directory.AgentDescription interestedAgent,
java.lang.String topic)
 - **Usage**
 - * register an interest in a topic with the distribution agent.
 - **Parameters**
 - * sender -
 - * interestedAgent - Interested Agent
 - * topic -
 - **Returns** -

- *requestVariables*
public static void **requestVariables**(
javax.agent.service.directory.AgentDescription sender, javax.agent.Locator
senderLocator, int clusterNr, java.lang.String pelementType)
 - **Usage**
 - * send a GET_ELEMENTS message
 - **Parameters**
 - * sender -
 - * senderLocator -
 - * clusterNr -
 - * pelementType -

- *sendMessage*
public static boolean **sendMessage**(
javax.agent.service.directory.AgentDescription sender, javax.agent.Locator
senderLocator, javax.agent.service.directory.AgentDescription destination,
aha.framework.AHAMessage content)
 - **Usage**
 - * Send a message to a single destination.

- * senderLocator -
- * sender - source
- * destination -
- * content -

– Returns -

- *sendMessageToDistributor*

```
public static boolean sendMessageToDistributor(
    javax.agent.service.directory.AgentDescription sender, javax.agent.Locator
    senderLocator, aha.framework.AHAMessage content, java.lang.String topic )
```

– Usage

- * send message to distributor for later distribution to all agents that registered an interest in a certain topic.

– Parameters

- * senderLocator -
- * topic - topic name
- * sender - source
- * content -

– Returns -

- *sendSetPropertyToBus*

```
public static void sendSetPropertyToBus(
    javax.agent.service.directory.AgentDescription sender, javax.agent.Locator
    senderLocator, java.lang.String name, java.lang.String value )
```

– Usage

- * Convenience Method: send SET_PROPERTY message to BusAgent

– Parameters

- * sender -
- * senderLocator -
- * name -
- * value -

- *unregisterInterest*

```
public static boolean unregisterInterest(
    javax.agent.service.directory.AgentDescription sender,
    javax.agent.service.directory.AgentDescription interestedAgent,
    java.lang.String topic )
```

– Usage

- * unregister an interest in a topic with the distribution agent.

– Parameters

- * topic -
- * sender -
- * interestedAgent - Interested Agent

– Returns -

Chapter 13

Package aha.rules

<i>Package Contents</i>	<i>Page</i>
Classes	
AHARuleBlock	135
<i>This class represents a ruleblock which contains a number of rules.</i>	
AHARuleSet	136
<i>This class represents a RuleSet.</i>	
ControlModel	137
<i>Model of the control agent.</i>	
DecisionHistory	141
<i>History of events taken in the past.</i>	
DecisionHistoryEntry	142
<i>One entry of the Decision History.</i>	
DynamicRuleTest	143
<i>Testclass that demonstrates the major features of the dynamic rules processing during runtime.</i>	
Model	145
<i>Represents a model, consisting of a number of fuzzy rules.</i>	
Rule	151
<i>A rule represents one single Fuzzy rule (Conditional if-then fuzzy rule) inside the system.</i>	
RuleExistsAlreadyException	153
<i>...no description...</i>	
RuleInfo	154
<i>Contains all data about a rule and allows access to it (Antecedents, Output variables).</i>	
RuleLabelIllegalException	155
<i>Illegal Rule label</i>	
RuleSupport	156
<i>This class contains methods that support the dynamic creation of able rules during runtime.</i>	

13.1 Classes

13.1.1 CLASS AHARuleBlock

This class represents a ruleblock which contains a number of rules. It supports the dynamic creation and removal of new rules during runtime.

AHARuleBlock is an extension of AbleRuleBlock and thus inherits directly from it. This is the only way to get access to the internal state of AbleRuleBlock as all these attributes are protected.

The only inference engine type currently supported is Fuzzy.

DECLARATION

```
public class AHARuleBlock
extends AbleRuleBlock
```

CONSTRUCTORS

- *AHARuleBlock*

```
public AHARuleBlock( )
```

 - **Usage**
 - * Creates a new instance of AHARuleBlock

METHODS

- *arlString*

```
public String arlString( )
```

 - **Usage**
 - * Return ARL string of this ruleblock. Rules are shown in the original creation order.
 - **Returns** - String
-
- *copyFromAbleRuleBlock*

```
public void copyFromAbleRuleBlock( AbleRuleSet ruleSet, AbleRuleBlock ruleBlockOld, AbleLogger trace )
```

 - **Usage**
 - * Copy data over from an existing ruleblock. This method takes all data from an existing RuleSet/RuleBlock and stores it in this instance. This is used to re-use RuleSet's/RuleBlock's that were instantiated by Able (and thus aren't instances of AHARuleBlock) as an AHARuleBlock instance. After all data was copied over from the existing AbleRuleBlock instance the reference to the AbleRuleBlock in the internal ABLE datastructures can be exchanged with a reference to a

– **Parameters**

- * **ruleSet** - existing ruleset that will be used for inferencing with the new ruleblock instance.
- * **ruleBlockOld** - existing instance from where the data is copied (including all rules and settings).
- * **trace** - the tracer that is assigned to the new AHARuleBlock instance.

• *getConditionalRules*

```
public Vector getConditionalRules( )
```

– **Usage**

- * Returns a vector of all conditional rules of this ruleblock.

– **Returns** - Collection of AHARule/AbleRule.

• *process*

```
public void process( )
```

– **Usage**

- * Make an inference. The input values must be set before this method is called.

– **Exceptions**

- * **AbleDataException** - in case inputs were set incorrectly/can't be processed/converted.
-

• *removeRule*

```
public void removeRule( java.lang.String label )
```

– **Usage**

- * Remove a rule from this ruleset.

– **Parameters**

- * **label** - label of the rule that should be removed.

13.1.2 CLASS AHARuleSet

This class represents a RuleSet. This class doesn't have any functionality itself. The only aim of this class is to provide access to the protected attributes to AbleRuleSet as this is the only way to change certain properties of AbleRuleSet.

DECLARATION

```
public class AHARuleSet
extends AbleRuleSet
```

CONSTRUCTORS

• *AHARuleSet*

```
public AHARuleSet( )
```

* Creates a new instance of AHARuleSet.

- *AHARuleSet*

```
public AHARuleSet( java.lang.String name )
```

- **Usage**

* creates a new ruleset with a specified name.

- **Parameters**

* **name** - name (label) of this ruleset.

METHODS

- *getRuleBlocksOrder*

```
public Vector getRuleBlocksOrder( )
```

- **Usage**

* Returns the field myRuleBlocksOrder out of the internal Able Datastructures for later processing.

- **Returns** - Vector of Rule's

13.1.3 CLASS ControlModel

Model of the control agent. This specific model (which is specifically programmed for one rule file) inherits from the generic model "Model". Only things specific for this model/rulebase are specified in this class.

DECLARATION

```
public class ControlModel
extends aha.rules.Model
```

FIELDS

- public static final int firedRules_INDEX

– Index of the field firedRules in the OutputBuffer Array of the inferencing process.

- public static final int LightAction_INDEX

– Index of the field lightAction in the OutputBuffer Array of the inferencing process.

- public static final int BlindAction_INDEX

– Index of the field blindAction in the OutputBuffer Array of the inferencing process.

- public static final int TemperatureFuzzyState_INDEX

– Index of the field TemperatureFuzzyState in the OutputBuffer Array of the inferencing process.

- Index of the field DayTimeFuzzyState in the OutputBuffer Array of the inferencing process.
- public static final int DayLightIndoorFuzzyState_INDEX
 - Index of the field DayLightIndoorFuzzyState in the OutputBuffer Array of the inferencing process.
- public static final int RadiationEastFuzzyState_INDEX
 - Index of the field RadiationEastFuzzyState in the OutputBuffer Array of the inferencing process.
- public static final int RadiationWestFuzzyState_INDEX
 - Index of the field RadiationWestFuzzyState in the OutputBuffer Array of the inferencing process.
- public static final int RadiationSouthFuzzyState_INDEX
 - Index of the field RadiationSouthFuzzyState in the OutputBuffer Array of the inferencing process.
- public static final int Illumination1FuzzyState_INDEX
 - Index of the field Illumination1FuzzyState in the OutputBuffer Array of the inferencing process.
- public static final int PresenceState_INDEX
 - Index of the field PresenceState in the OutputBuffer Array of the inferencing process.
- public static final int INPUT_ARRAY_SIZE
 - Number of elements in the InputBuffer array of the inferencing process.
- public static final int OUTPUT_ARRAY_SIZE
 - Number of Elements in the OutputBuffer array of the inferencing process.
- public static final int INPUT_Temperature_INDEX
 - Position of this field in the InputBuffer of the inferencing process.
- public static final int INPUT_DayTime_INDEX
 - Position of this field in the InputBuffer of the inferencing process.
- public static final int INPUT_Presence_INDEX
 - Position of this field in the InputBuffer of the inferencing process.
- public static final int INPUT_DayLightIndoor_INDEX
 - Position of this field in the InputBuffer of the inferencing process.
- public static final int INPUT_RadiationEast_INDEX
 - Position of this field in the InputBuffer of the inferencing process.
- public static final int INPUT_RadiationWest_INDEX
 - Position of this field in the InputBuffer of the inferencing process.

- Position of this field in the InputBuffer of the inferencing process.
- public static final int INPUT_Illumination1_INDEX
 - Position of this field in the InputBuffer of the inferencing process.
- public static final String INPUT_Temperature
 - Fullname (String) of this input variable.
- public static final String INPUT_DayTime
 - Fullname (String) of this input variable.
- public static final String INPUT_Presence
 - Fullname (String) of this input variable.
- public static final String INPUT_DayLightIndoor
 - Fullname (String) of this input variable.
- public static final String INPUT_RadiationEast
 - Fullname (String) of this input variable.
- public static final String INPUT_RadiationWest
 - Fullname (String) of this input variable.
- public static final String INPUT_RadiationSouth
 - Fullname (String) of this input variable.
- public static final String INPUT_Illumination1
 - Fullname (String) of this input variable.
- public static final String OUTPUT_LIGHT_ACTION
 - Fullname (String) of this output variable.
- public static final String OUTPUT_BLIND_ACTION
 - Fullname (String) of this output variable.
- public static final String OUTPUT_TEMPERATURE
 - Fullname (String) of this output variable.
- public static final String OUTPUT_DAYTIME
 - Fullname (String) of this output variable.
- public static final String OUTPUT_DAYLIGHTINDOOR
 - Fullname (String) of this output variable.
- public static final String OUTPUT_RADIATIONEAST
 - Fullname (String) of this output variable.
- public static final String OUTPUT_RADIATIONWEST
 - Fullname (String) of this output variable.

- Fullname (String) of this output variable.
- public static final String OUTPUT_ILLUMINATION1
 - Fullname (String) of this output variable.
- public static final String OUTPUT_PRESENCE
 - Fullname (String) of this output variable.
- public static final Map FuzzyOutputs
 - key=string, value=position (Integer)
- public static final Map FuzzyInputs
 - key=string, value=position (Integer)

CONSTRUCTORS

- *ControlModel*
public ControlModel()
 - **Usage**
 - * Creates a new instance of ControlModel.

METHODS

- *initFuzzyInputs*
public static Map initFuzzyInputs()
 - **Usage**
 - * Initializes a static map which contains all associations between Input-Variables (Strings) and there positions in the InputBuffer array of the fuzzy inferencing process.
 - **Returns -**
-
- *initFuzzyOutputs*
public static Map initFuzzyOutputs()
 - **Usage**
 - * Initializes a static map which contains all associations between Output-Variables (Strings) and there positions in the OutputBuffer array of the fuzzy inferencing process.
 - **Returns -**
-
- *postProcess*
public void postProcess()
 - **Usage**
 - * Update state variables for every rule. Process results of inferencing process. Adds an entry to the decision history.

- *preProcess*
public void preProcess()
 - **Usage**
 - * Prepare inferencing process.

13.1.4 CLASS DecisionHistory

History of events taken in the past. used for SSA (success story algorithm) for learning.

DECLARATION

```
public class DecisionHistory
extends java.lang.Object
```

CONSTRUCTORS

- *DecisionHistory*
public DecisionHistory()
 - **Usage**
 - * Creates a new instance of DecisionHistory

METHODS

- *addDecisionHistoryEntry*
public double addDecisionHistoryEntry(aha.rules.DecisionHistoryEntry entry)
 - **Usage**
 - * add a new entry to the decision history. automatically sets the id of the entry.
 - **Parameters**
 - * **entry** - DecisionHistory entry
 - **Returns** - id of this entry

- *getDecisionHistoryEntry*
public DecisionHistoryEntry getDecisionHistoryEntry(double id)
 - **Usage**
 - * return a sigle entry.
 - **Parameters**
 - * **id** - DecisionHistoryEntry ID
 - **Returns** - null or instance of DecisionHistoryEntry

- *removeDecisionHistoryEntry*

- **Usage**
 - * remove a single entry.
- **Parameters**
 - * *id* - decisionHistoryEntry ID
- **Returns** - null if not found, the removed entry if found.

13.1.5 CLASS DecisionHistoryEntry

One entry of the Decision History.

DECLARATION

```
public class DecisionHistoryEntry
extends java.lang.Object
```

CONSTRUCTORS

- *DecisionHistoryEntry*
public DecisionHistoryEntry()
 - **Usage**
 - * Creates a new instance of DecisionHistoryEntry

METHODS

- *addFiringRule*
public void addFiringRule(java.lang.String ruleLabel)
 - **Usage**
 - * add a label of a rule that has fired (evaluated to true)
 - **Parameters**
 - * *ruleLabel* - label of a rule.
-
- *addInputValue*
public void addInputValue(java.lang.String key, java.lang.String value)
 - **Usage**
 - * add an input value to this entry.
 - **Parameters**
 - * *key* - String, defined as constant.
 - * *value* - String, describing a fuzzy state.

-
- *getFiringRules*
public Collection getFiringRules()

- **Usage**
 - * return a collection of all rules (labels=String).
 - **Returns** - Collection of Strings
-
- *getId*

```
public double getId( )
```

 - **Usage**
 - * return the id of this entry.
 - **Returns** - id of this entry
-
- *getInputValues*

```
public Map getInputValues( )
```

 - **Usage**
 - * return all input values.
 - **Returns** - Map
-
- *getTimeStamp*

```
public Date getTimeStamp( )
```

 - **Usage**
 - * return timestamp.
 - **Returns** - Date
-
- *setId*

```
public void setId( double id )
```

 - **Usage**
 - * set the ID of this entry. needs to be unique for this decisionHistory.
 - **Parameters**
 - * id - double
-
- *toString*

```
public String toString( )
```

 - **Usage**
 - * debug output.
 - **Returns** - debug output.

13.1.6 CLASS DynamicRuleTest

TestClass that demonstrates the major features of the dynamic rules processing during runtime. stores state persistently (rules->ARL) and loads it again.

DECLARATION

```
public class DynamicRuleTest
extends java.lang.Object
```

CONSTRUCTORS

- *DynamicRuleTest*
public **DynamicRuleTest**()
 - **Usage**
 - * Creates a new instance of *DynamicRuleTest*.

METHODS

- *change1*
public void **change1**()
 - **Usage**
 - * add a new rule.

- *change2*
public void **change2**()
 - **Usage**
 - * deactivate an existing rule.

- *change3*
public void **change3**()
 - **Usage**
 - * Remove some rules.

- *debugOut*
public void **debugOut**()
 - **Usage**
 - * Print features of fuzzy variable Temperature.

- *load*
public void **load**(java.lang.String filenameARL, java.lang.String filenamePROP)
 - **Usage**
 - * instanciate from a ARL rulefile.
 - **Parameters**
 - * filenameARL - full path+filename for ARL file.
 - * filenamePROP - full path+filename for properties file.

- *main*
public static void **main**(java.lang.String [] args)
 - **Usage**
 - * Main for testing.

* **args** - no arguments necessary.

- *ruleInfo*

```
public void ruleInfo( )
```

- **Usage**

- * Printout RuleInfo class of some rules.

- *run*

```
public void run( )
```

- **Usage**

- * Run inferencing process.

- *store*

```
public void store( java.lang.String filenameARL, java.lang.String
filenamePROP )
```

- **Usage**

- * Store rules as an ARL file.

- **Parameters**

- * **filenameARL** - full path+filename for ARL file.

- * **filenamePROP** - full path+filename for properties file.

13.1.7 CLASS Model

Represents a model, consisting of a number of fuzzy rules. A model describes a part of the world and how AHA sees this part of the world. The model is the basis for all decisions taken and all learning taking place. Feedback from the real world changes the model.

DECLARATION

```
public class Model
extends java.lang.Object
```

CONSTRUCTORS

- *Model*

```
public Model( )
```

- **Usage**

- * Creates a new instance of Model.

METHODS

- *addRule*

```
public boolean addRule( AbleRule newRule )
```

- **Usage**

- * Add a new rule to this existing ruleset. This method only works if previously a existing ruleset (which may be empty) was loaded from an ARL file.

Not all types of Rules that are supported by Able are supported by this Method.
Currently Supported are: AbleConditionalRule.

The label of the rule must start with the prefix RULE_.

- **Parameters**

- * *newRule* -

- **Returns** - true if successfull.

- **Exceptions**

- * `aha.rules.RuleExistsAlreadyException` - If the rule exists already (same label).

- *arlString*

```
public String arlString( )
```

- **Usage**

- * returns the arl version of this model.

- *disableRule*

```
public boolean disableRule( java.lang.String label )
```

- **Usage**

- * disable a rule. Disabled rules aren't considered during decision making.

- **Parameters**

- * *label* - label of the rule.

- **Returns** - true if successfull.

- *enableRule*

```
public boolean enableRule( java.lang.String label )
```

- **Usage**

- * set a rules state to enable

- **Parameters**

- * *label* - label of the rule.

- **Returns** - true if successfull.

- *evaluateResult*

```
public void evaluateResult( )
```

- **Usage**

- * find out which rule contributed to the final result and how much it did so

- *getDefinedVariable*

```
public AbleVariable getDefinedVariable( java.lang.String  variableName )
```

- **Usage**

- * return a variable defined in this model/ruleset.

- **Parameters**

- * variableName -

- **Returns -**

- *getFuzzySetOfFuzzyVariable*

```
public AbleFuzzySet getFuzzySetOfFuzzyVariable( java.lang.String
variableName, java.lang.String  fuzzyssetName )
```

- **Usage**

- * extract and return a fuzzysset from a fuzzy variable.

- **Parameters**

- * variableName -

- * fuzzyssetName -

- **Returns -**

- *getInputBuffer*

```
public Object getInputBuffer( )
```

- **Usage**

- * return the input buffer of the inferencing process. the returned object array contains the input data in the same order as declared in the element Inputs in the ARL specification of the ruleset underlying this model.

- **Returns -**

- **Exceptions**

- * java.rmi.RemoteException -

- *getOutputBuffer*

```
public Object getOutputBuffer( )
```

- **Usage**

- * get output buffer. can only be called after process was called.

- **Returns -**

- **Exceptions**

- * java.rmi.RemoteException -

- *getRule*

```
public Rule getRule( java.lang.String  label )
```

- **Usage**

- * Get the rule identified by label.

- **Parameters**

- * label - label of the rule.

- **Returns -** Rule instance if it exists, null if not.

- *getRuleEnabledState*

– **Usage**

* returns true if the rule identified by label is set to true. A rule that is enabled is being considered during every decision making process/inference.

– **Parameters**

* `label` - label of the rule.

– **Returns** - true if rule is enabled.

• *getRuleSet*

```
public AHARuleSet getRuleSet( )
```

– **Usage**

* Return the ruleset instance of this model.

– **Returns** - Null if there is no active ruleset. AHARuleSet instance otherwise.

• *getSortedListOfFuzzySets*

```
public SortedSet getSortedListOfFuzzySets( java.lang.String variableName )
```

– **Usage**

* return a sorted list of all sets. because of the special names of the fuzzy sets (A,B,C as first letter) this list returns the sets in the same order as they appear in the membership function from left to right on the x-axis.

– **Parameters**

* `variableName` -

– **Returns** -

• *getTriggeringRule*

```
public String getTriggeringRule( )
```

– **Usage**

* returns the name of the rule that triggered this result (caused). returned result is null in case no processing took place before or in case no rule triggered.

– **Returns** -

• *initRuleset*

```
public void initRuleset( )
```

– **Usage**

* Init a ruleset. The ruleset needs to be instantiated before this method has any effect.

• *loadFromFile*

```
public void loadFromFile( java.lang.String fileNameRule, java.lang.String fileNameProperties )
```

– **Usage**

* Load this model from a ARL file and a properties file.

– **Parameters**

* `fileNameRule` - ARL file.

* `fileNameProperties` - Properties file with additional properties for this ARL file.

- *postProcess*
protected void **postProcess**()
 - **Usage**
 - * update state variables for every rule. process results of inferencing process.

- *preProcess*
protected void **preProcess**()
 - **Usage**
 - * prepare inference process

- *process*
public void **process**()
 - **Usage**
 - * start inferencing. The input buffer needs to be set before this method can be called.
 - **Exceptions**
 - * java.rmi.RemoteException -

- *removeRule*
public void **removeRule**(java.lang.String label)
 - **Usage**
 - * remove a rule from this Model.
 - **Parameters**
 - * label - label of the rule.

- *reset*
public void **reset**()
 - **Usage**
 - * re-initialize all rules with "not fired".
 - **Exceptions**
 - * java.rmi.RemoteException -

- *returnNextHigherSet*
public AbleFuzzySet **returnNextHigherSet**(java.lang.String variableName, java.lang.String setName)
 - **Usage**
 - * returns the next higher set (on the x-axis) of the supplied fuzzy variable relativ to the supplied fuzzy set. if there is no higher one returns null.
 - **Parameters**
 - * variableName -
 - * setName -
 - **Returns** -

- *returnNextLowerSet*
public AbleFuzzySet **returnNextLowerSet**(java.lang.String variableName,

- **Usage**
 - * returns the next lower set (on the x-axis) of the supplied fuzzy variable relativ to the supplied fuzzy set. if there is no lower one returns null.
 - **Parameters**
 - * `variableName` -
 - * `setName` -
 - **Returns** -
-

- *ruleExists*

```
public boolean ruleExists( java.lang.String label )
```

- **Usage**
 - * Tests whether a rule exists already or not.
 - **Parameters**
 - * `label` - Label of the rule.
 - **Returns** - true if the rule exists already.
-

- *saveToFile*

```
public void saveToFile( java.lang.String fileNameRule, java.lang.String
fileNameProperties )
```

- **Usage**
 - * save this model back to an ARL and a properties file.
 - **Parameters**
 - * `fileNameRule` - ARL file.
 - * `fileNameProperties` - Propertiesfile with additional properties for this ARL file.
-

- *setInputBuffer*

```
public void setInputBuffer( java.lang.Object [] obj )
```

- **Usage**
 - * Set input values for inferencing process.
 - **Parameters**
 - * `obj` -
 - **Exceptions**
 - * `java.rmi.RemoteException` -
-

- *setTriggeringRule*

```
protected void setTriggeringRule( java.lang.String triggeringRuleLabel )
```

- **Usage**
 - * set label of the triggering rule. internal function for Model.
 - **Parameters**
 - * `triggeringRuleLabel` -
-

- *toString*

```
public String toString( )
```

- **Usage**

13.1.8 CLASS Rule

A rule represents one single Fuzzy rule (Conditional if-then fuzzy rule) inside the system. For every AbleRule there is one Rule instance present. This class (Rule) holds additional attributes that are required during the learning process but that aren't part of the Rule itself (AbleRule). All additional attributes stored in this class are saved as comments for the corresponding AbleRule instance.

DECLARATION

```
public class Rule
extends java.lang.Object
```

FIELDS

- public static final String RULE_STATE_VARIABLE_DYNAMIC
 - Tells whether a rule is dynamic or static. dynamic rules are rules that can possibly be changed by a learning algorithm. possible values: true/false.
- public static final String RULE_STATE_VARIABLE_FIREDCOUNT
 - FiredCount tells how many times this rule has fired (evaluated to true).
- public static final String RULE_STATE_VARIABLE_ENABLED
 - Tells whether a rule is enabled (used for inferencing) or disabled (not used for inferencing). Possible values: true/false.
- public static final String RULE_STATE_VARIABLE_EVALUATEDCOUNT
 - Tells how many times this rule has been evaluated so far.
- public static final String RULE_STATE_VARIABLE_PUNISH_COUNTER
 - punish counter (feedback)
- public static final String RULE_STATE_VARIABLE_REWARD_COUNTER
 - reward counter (feedback)
- public static final String RULE_STATE_VARIABLE_ORIGINAL_RULE
 - original rule for learning purposes

CONSTRUCTORS

- *Rule*

```
public Rule( AbleRule ableRule, AbleRuleSet ruleSet )
```

 - **Usage**
 - * create a new Rule and associate it with a corresponding ABLE rule
 - **Parameters**
 - * `ableRule` - reference to AbleRule which instantiates this rule.

METHODS

• *getAbleRule*

```
public AbleRule getAbleRule( )
```

– **Usage**

* Return original AbleRule instance of this rule.

– **Returns** - rule

• *getLabel*

```
public String getLabel( )
```

– **Usage**

* return label of this rule.

– **Returns** - label

• *getParameter*

```
public String getParameter( java.lang.String key )
```

– **Usage**

* Returns the parameter identified by key

– **Parameters**

* **key** - any string, all possible strings are defined as static constants in this Class.

– **Returns** - String or null if not found.

• *getRuleInfo*

```
public RuleInfo getRuleInfo( )
```

– **Usage**

* extract RuleInfo.

works only for AbleRule's of type AbleConditionalRules.

– **Returns** - ruleinfo (which is empty in case of error or null).

• *increasePunishCounter*

```
public void increasePunishCounter( )
```

– **Usage**

* increase punish counter.

• *increaseRewardCounter*

```
public void increaseRewardCounter( )
```

– **Usage**

* increase reward counter

• *incrementParameter*

```
public void incrementParameter( java.lang.String key )
```

– **Usage**

* increments a parameter (+ 1) in case the parameter already exists or sets the parameter=1 in case it doesn't exist yet.

– **Parameters**

* **key** - any string, all possible strings are defined as static constants in this Class.

• *prepareSave*

public void **prepareSave**()

– **Usage**

* generates the comment line for this rule and sets it to the corresponding able rule. this method should be called before saving the rules to file.

• *resetRuleInfo*

public void **resetRuleInfo**()

– **Usage**

* resets ruleinfo so that ruleinfo gets regenerated when it is requested the next time

• *setParameter*

public void **setParameter**(java.lang.String **key**, java.lang.String **value**)

– **Usage**

* change/add a new parameter. This Parameter will automatically be stored persistently when this rule is stored to disc.

– **Parameters**

* **key** - any string, all possible strings are defined as static constants in this Class.
* **value** - any String

13.1.9 CLASS RuleExistsAlreadyException

DECLARATION

```
public class RuleExistsAlreadyException
extends java.lang.Exception
```

CONSTRUCTORS

• *RuleExistsAlreadyException*

public **RuleExistsAlreadyException**()

– **Usage**

* Creates a new instance of RuleExistsAlreadyException without detail message.

• *RuleExistsAlreadyException*

public **RuleExistsAlreadyException**(java.lang.String **msg**)

Usage

* Constructs an instance of `RuleExistsAlreadyException` with the specified detail message.

– **Parameters**

* `msg` - the detail message.

13.1.10 CLASS RuleInfo

Contains all data about a rule and allows access to it (Antecedents, Output variables).

DECLARATION

```
public class RuleInfo
extends java.lang.Object
```

CONSTRUCTORS

- *RuleInfo*
public RuleInfo()
 - **Usage**
 - * Creates a new instance of RuleInfo.

METHODS

- *addAntecedentVariable*
public void addAntecedentVariable(java.lang.String varName, java.lang.String value)
 - **Usage**
 - * Adds an additional condition to the list of antecedents.
 - **Parameters**
 - * `varName` -
 - * `value` -

- *addOutputVariable*
public void addOutputVariable(java.lang.String varName, java.lang.String varValue)
 - **Usage**
 - * Add an additional output variable and it's value. This will be added to the list of consequences of this rule.
 - **Parameters**
 - * `varName` -
 - * `varValue` -

- *getAntecedentVariables*

- **Usage**
 - * Getter for property antecedentVariables.
- **Returns** - Value of property antecedentVariables.

- *getOutputVariables*

```
public Map getOutputVariables( )
```

- **Usage**
 - * Getter for property outputVariables.
- **Returns** - Value of property outputVariables.

- *setAntecedentVariables*

```
public void setAntecedentVariables( java.util.Map antecedentVariables )
```

- **Usage**
 - * Setter for property antecedentVariables.
- **Parameters**
 - * antecedentVariables - New value of property antecedentVariables.

- *setOutputVariables*

```
public void setOutputVariables( java.util.Map outputVariables )
```

- **Usage**
 - * Setter for property outputVariables.
- **Parameters**
 - * outputVariables - New value of property outputVariables.

- *toString*

```
public String toString( )
```

- **Usage**
 - * Returns the complete rule in a human readable form.
- **Returns** -

13.1.11 CLASS RuleLabelIllegalException

Illegal Rule label

DECLARATION

```
public class RuleLabelIllegalException
extends java.lang.Exception
```

 CONSTRUCTORS

- *RuleLabelIllegalException*
 public **RuleLabelIllegalException**()
 - **Usage**
 - * Creates a new instance of RuleLabelIllegalException without detail message.
-
- *RuleLabelIllegalException*
 public **RuleLabelIllegalException**(java.lang.String msg)
 - **Usage**
 - * Constructs an instance of RuleLabelIllegalException with the specified detail message.
 - **Parameters**
 - * msg - the detail message.

13.1.12 CLASS RuleSupport

This class contains methods that support the dynamic creation of able rules during runtime. All methods are static so there needs to be no instance of this class.

 DECLARATION

```
public class RuleSupport
extends java.lang.Object
```

 FIELDS

- public static final double DEFAULT_RULE_PRIORITY
 - default priority for a rule

 CONSTRUCTORS

- *RuleSupport*
 public **RuleSupport**()

 METHODS

- *createConditionalRule*
 public static AbleConditionalRule **createConditionalRule**(java.lang.String label, double priority, AbleFuzzyVariable antecedentVariable, AbleFuzzySet antecedentCondition, AbleVariable thenVariable, java.lang.String

– **Usage**

- * constructs a simple fuzzy if ... then rule. antecedents are of the kind "if x is y" where x is fuzzy variable the then list if of the kind "Z="NN"" where Z is a categorical variable

– **Parameters**

- * **label** - label of the new rule.
- * **priority** - priority of the new rule (the higher the value the higher the priority).
- * **antecedentVariable** - the antecedent variable.
- * **antecedentCondition** - the antecedent condition (A fuzzyset instance).
- * **thenVariable** - the consequent variable
- * **thenValue** - the consequent value
- * **thenVariableFired** -

– **Returns** - AbleConditionalRule instance

• *createConditionalRule*

```
public static AbleConditionalRule createConditionalRule( java.lang.String
label, double priority, java.util.Vector theAntExprList, java.util.Vector
theAntecedents, AbleVariable thenVariable, java.lang.Object thenValue,
AbleVariable thenVariableFired )
```

– **Usage**

- * constructs a simple fuzzy if ... then rule. antecedents are of the kind "if x is y" where x is fuzzy variable the then list if of the kind "Z="NN"" where Z is a categorical variable

– **Parameters**

- * **label** - label of the new rule.
- * **priority** - priority of the new rule (the higher the value the higher the priority).
- * **theAntExprList** - List of antecedent expressions.
- * **theAntecedents** - list of antecedents
- * **thenVariable** - the consequent variable
- * **thenValue** - the consequent value
- * **thenVariableFired** - additonal consequent variable.

– **Returns** - AbleConditionalRule instance

• *getDefinedVariable*

```
public static AbleVariable getDefinedVariable( AbleRuleSet ruleSet,
java.lang.String variableName )
```

– **Usage**

- * extracts and returns a AbleVariable from a ruleset. the returned variable is read-only.

this method returns null if the variable wasnt found.

– **Parameters**

- * **ruleSet** - from which ruleset
- * **variableName** - name of the variable

– **Returns** - AbleVariable instance if found.

• *getFuzzySetOfFuzzyVariable*

```
public static AbleFuzzySet getFuzzySetOfFuzzyVariable( AbleFuzzyVariable
```

- **Usage**
 - * Extracts a single set of a fuzzy variable.
 - **Parameters**
 - * `variable` - Name of the fuzzy variable.
 - * `fuzzysetName` - Name of the fuzzy set.
 - **Returns** - A `AbleFuzzySet` instance (or a subclass).
-

- *getPossibleLabels*

```
public static SortedSet getPossibleLabels( AbleFuzzyVariable variable )
```

- **Usage**
 - * returns a sorted Collection of all possible labels (membership functions) of a fuzzy variable. Requests are cached internally to make this faster.
- **Parameters**
 - * `variable` - Name of a fuzzy variable.
- **Returns** - Collection of String.

Chapter 14

Package `aha.sim.persons`

<i>Package Contents</i>	<i>Page</i>
<hr/>	
Classes	
Person	160
<i>Models a virtual person with all it behaviour</i>	
PersonLocator	161
<i>This agent is responsible for simulating virtual persons in the simulation.</i>	

14.1 Classes

14.1.1 CLASS Person

Models a virtual person with all it behaviour

DECLARATION

```
public class Person
extends java.lang.Object
```

CONSTRUCTORS

- *Person*
 public **Person**(int id, java.util.Date start, java.util.Date end)

METHODS

- *addBehaviour*
 public void **addBehaviour**(aha.sim.persons.config.Behaviour b)
 – **Usage**
 * adds a behaviour to this virtual person
 – **Parameters**
 * b - the behaviour to add

- *addRoom*
 public void **addRoom**(aha.sim.persons.config.Room room)
 – **Usage**
 * adds a room to the rouletteWheel selection collection
 – **Parameters**
 * room - the room to add

- *changeRoom*
 public int **changeRoom**(java.util.Date time)
 – **Usage**
 * check if the person left or entered a room
 – **Parameters**
 * time - the actual time (simulation time)
 – **Returns** - int actual clusterId

- *getActualCluster*
 public int **getActualCluster**()

- * returns the id of the cluster the person is presently in
 - **Returns** - int

- *getDestinations*
 - public HashMap **getDestinations**(java.util.HashMap **deviceTypes**,
aha.sim.persons.config.Behaviour **b**)
 - **Usage**
 - * returns a HashMap with all the variables affected by a behaviour
 - **Parameters**
 - * **deviceTypes** - the type of variables affected
 - * **b** - the Behaviour
 - **Returns** - HashMap the variables affected

- *getId*
 - public int **getId**()
 - **Usage**
 - * returns the personId
 - **Returns** - int the personId

- *updateSwitches*
 - public HashMap **updateSwitches**(java.util.HashMap **variables**,
java.util.HashMap **varValues**)
 - **Usage**
 - * Checks if any of the switches in the current room should be changed.
 - **Parameters**
 - * **variables** - list of all variables and their type per cluster
 - * **varValues** - all the current variable values
 - **Returns** - HashMap with the decisions for the switches

14.1.2 CLASS PersonLocator

This agent is responsible for simulating virtual persons in the simulation. It sends the same messages as the Bluetooth PersonLocator.

DECLARATION

```
public class PersonLocator
extends aha.framework.AHAAgent
```

CONSTRUCTORS

- *PersonLocator*
- public **PersonLocator**()

METHODS

- *init*
 public void **init**()
 - **Usage**
 - * Initializes the AbleAgent and loads the virtual persons from the configuration file.
 - **See Also**
 - * com.ibm.able.AbleBean.init()

- *processGetElements*
 public void **processGetElements**(aha.framework.AHAMessage msg)
 - **Usage**
 - * Processes the message GET_ELEMENTS_RESULT
 - **Parameters**
 - * msg -

- *processSetProperty*
 public void **processSetProperty**(java.util.Map param)
 - **Usage**
 - * processes a recieved VARIABLE_UPDATE
 - **Parameters**
 - * param - the params of the message

- *processUpdatePersons*
 public void **processUpdatePersons**(aha.framework.AHAMessage msg)
 - **Usage**
 - * Updates the persons locations whenever the message UPDATE_PERSONS is recieved
 - **Parameters**
 - * msg - the AHAMessage

- *receiveMessage*
 public void **receiveMessage**(javax.agent.TransportMessage theTransportMessage)
 - **Usage**
 - * recieves the messages UPDATE_PERSONS, GET_ELEMENTS_RESULT and VARIABLE_UPDATE
 - **See Also**
 - *
 javax.agent.service.transport.MessageListener.receiveMessage(TransportMessage)