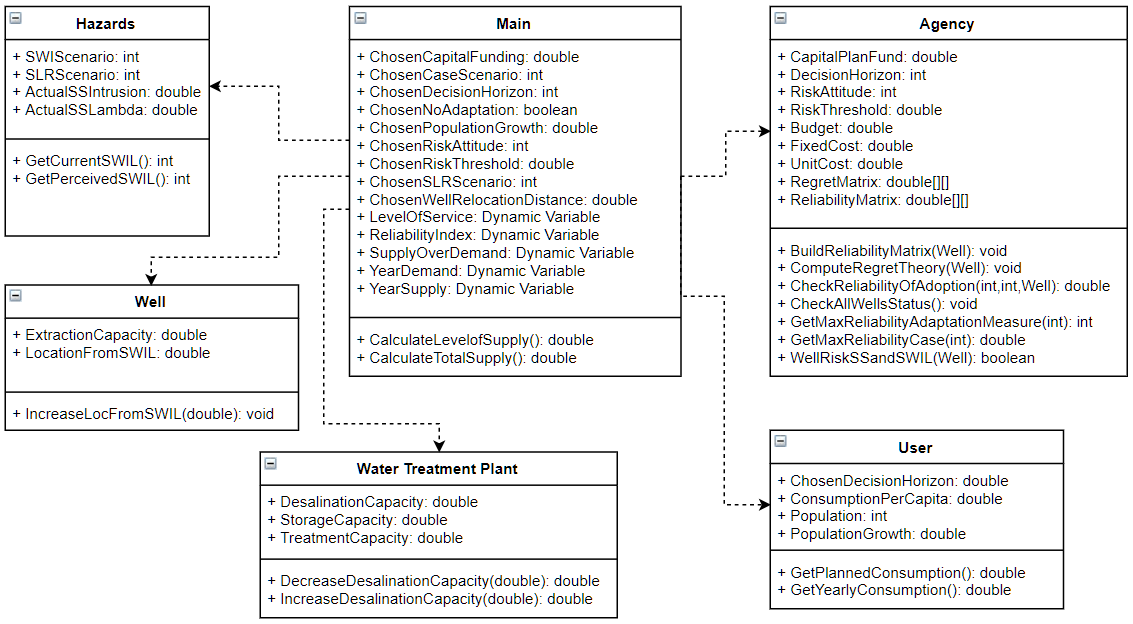
The simulation model has been uploaded in an open repository (AnyLogic Cloud), where it can be implemented online. Please use the link below to access to the model:

[https://cloud.anylogic.com/model/f050d71d-fe4c-426f-b6e1-d7f58ed75213?mode=SETTINGS](https://cloud.anylogic.com/model/f050d71d-fe4c-426f-b6e1-d7f58ed75213?mode=SETTINGS%20)

The figure below shows the Unified Modeling Language of the agents’ classes created in the simulation model:



The source codes (in Java) of the simulation model agents are as following:

**Agency Agent:**

public class Agency extends Agent

{

// Parameters

public

double CapitalPlanFund;

/\*\*

\* Returns default value for parameter <code>CapitalPlanFund</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public double \_CapitalPlanFund\_DefaultValue\_xjal() {

final Agency self = this;

return

main.ChosenCapitalFunding

;

}

public void set\_CapitalPlanFund( double CapitalPlanFund ) {

if (CapitalPlanFund == this.CapitalPlanFund) {

return;

}

double \_oldValue\_xjal = this.CapitalPlanFund;

this.CapitalPlanFund = CapitalPlanFund;

onChange\_CapitalPlanFund\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter CapitalPlanFund.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_CapitalPlanFund()</code> method instead.

\*/

protected void onChange\_CapitalPlanFund() {

onChange\_CapitalPlanFund\_xjal( CapitalPlanFund );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_CapitalPlanFund\_xjal( double oldValue ) {

}

public

int RiskAttitude;

/\*\*

\* Returns default value for parameter <code>RiskAttitude</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public int \_RiskAttitude\_DefaultValue\_xjal() {

final Agency self = this;

return 0;

}

public void set\_RiskAttitude( int RiskAttitude ) {

if (RiskAttitude == this.RiskAttitude) {

return;

}

int \_oldValue\_xjal = this.RiskAttitude;

this.RiskAttitude = RiskAttitude;

onChange\_RiskAttitude\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter RiskAttitude.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_RiskAttitude()</code> method instead.

\*/

protected void onChange\_RiskAttitude() {

onChange\_RiskAttitude\_xjal( RiskAttitude );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_RiskAttitude\_xjal( int oldValue ) {

}

public

int DecisionHorizon;

/\*\*

\* Returns default value for parameter <code>DecisionHorizon</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public int \_DecisionHorizon\_DefaultValue\_xjal() {

final Agency self = this;

return 0;

}

public void set\_DecisionHorizon( int DecisionHorizon ) {

if (DecisionHorizon == this.DecisionHorizon) {

return;

}

int \_oldValue\_xjal = this.DecisionHorizon;

this.DecisionHorizon = DecisionHorizon;

onChange\_DecisionHorizon\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter DecisionHorizon.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_DecisionHorizon()</code> method instead.

\*/

protected void onChange\_DecisionHorizon() {

onChange\_DecisionHorizon\_xjal( DecisionHorizon );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_DecisionHorizon\_xjal( int oldValue ) {

}

public

double RiskThreshold;

/\*\*

\* Returns default value for parameter <code>RiskThreshold</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public double \_RiskThreshold\_DefaultValue\_xjal() {

final Agency self = this;

return 0.0;

}

public void set\_RiskThreshold( double RiskThreshold ) {

if (RiskThreshold == this.RiskThreshold) {

return;

}

double \_oldValue\_xjal = this.RiskThreshold;

this.RiskThreshold = RiskThreshold;

onChange\_RiskThreshold\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter RiskThreshold.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_RiskThreshold()</code> method instead.

\*/

protected void onChange\_RiskThreshold() {

onChange\_RiskThreshold\_xjal( RiskThreshold );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_RiskThreshold\_xjal( double oldValue ) {

}

public

int RiskUpdateLimit;

/\*\*

\* Returns default value for parameter <code>RiskUpdateLimit</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public int \_RiskUpdateLimit\_DefaultValue\_xjal() {

final Agency self = this;

return

100

;

}

public void set\_RiskUpdateLimit( int RiskUpdateLimit ) {

if (RiskUpdateLimit == this.RiskUpdateLimit) {

return;

}

int \_oldValue\_xjal = this.RiskUpdateLimit;

this.RiskUpdateLimit = RiskUpdateLimit;

onChange\_RiskUpdateLimit\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter RiskUpdateLimit.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_RiskUpdateLimit()</code> method instead.

\*/

protected void onChange\_RiskUpdateLimit() {

onChange\_RiskUpdateLimit\_xjal( RiskUpdateLimit );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_RiskUpdateLimit\_xjal( int oldValue ) {

}

@Override

public void setParametersToDefaultValues() {

super.setParametersToDefaultValues();

CapitalPlanFund = \_CapitalPlanFund\_DefaultValue\_xjal();

RiskAttitude = \_RiskAttitude\_DefaultValue\_xjal();

DecisionHorizon = \_DecisionHorizon\_DefaultValue\_xjal();

RiskThreshold = \_RiskThreshold\_DefaultValue\_xjal();

RiskUpdateLimit = \_RiskUpdateLimit\_DefaultValue\_xjal();

}

@Override

public boolean setParameter(String \_name\_xjal, Object \_value\_xjal, boolean \_callOnChange\_xjal) {

switch ( \_name\_xjal ) {

case "CapitalPlanFund":

if ( \_callOnChange\_xjal ) {

set\_CapitalPlanFund( ((Number) \_value\_xjal).doubleValue() );

} else {

CapitalPlanFund = ((Number) \_value\_xjal).doubleValue();

}

return true;

case "RiskAttitude":

if ( \_callOnChange\_xjal ) {

set\_RiskAttitude( ((Number) \_value\_xjal).intValue() );

} else {

RiskAttitude = ((Number) \_value\_xjal).intValue();

}

return true;

case "DecisionHorizon":

if ( \_callOnChange\_xjal ) {

set\_DecisionHorizon( ((Number) \_value\_xjal).intValue() );

} else {

DecisionHorizon = ((Number) \_value\_xjal).intValue();

}

return true;

case "RiskThreshold":

if ( \_callOnChange\_xjal ) {

set\_RiskThreshold( ((Number) \_value\_xjal).doubleValue() );

} else {

RiskThreshold = ((Number) \_value\_xjal).doubleValue();

}

return true;

case "RiskUpdateLimit":

if ( \_callOnChange\_xjal ) {

set\_RiskUpdateLimit( ((Number) \_value\_xjal).intValue() );

} else {

RiskUpdateLimit = ((Number) \_value\_xjal).intValue();

}

return true;

default:

return super.setParameter( \_name\_xjal, \_value\_xjal, \_callOnChange\_xjal );

}

}

@Override

public <T> T getParameter(String \_name\_xjal) {

Object \_result\_xjal;

switch ( \_name\_xjal ) {

case "CapitalPlanFund": \_result\_xjal = CapitalPlanFund; break;

case "RiskAttitude": \_result\_xjal = RiskAttitude; break;

case "DecisionHorizon": \_result\_xjal = DecisionHorizon; break;

case "RiskThreshold": \_result\_xjal = RiskThreshold; break;

case "RiskUpdateLimit": \_result\_xjal = RiskUpdateLimit; break;

default: \_result\_xjal = super.getParameter( \_name\_xjal ); break;

}

return (T) \_result\_xjal;

}

@AnyLogicInternalCodegenAPI

private static String[] \_parameterNames\_xjal;

@Override

public String[] getParameterNames() {

String[] result = \_parameterNames\_xjal;

if (result == null) {

List<String> list = new ArrayList<>( Arrays.asList( super.getParameterNames() ) );

list.add( "CapitalPlanFund" );

list.add( "RiskAttitude" );

list.add( "DecisionHorizon" );

list.add( "RiskThreshold" );

list.add( "RiskUpdateLimit" );

result = list.toArray( new String[ list.size() ] );

\_parameterNames\_xjal = result;

}

return result;

}

// Plain Variables

public

double

Budget;

public

double[][]

ReliabilityMatrix;

public final

int

RISK\_SEEKER =

0

;

public final

int

RISK\_AVERSE =

2

;

public final

int

RISK\_NEUTRAL =

1

;

public final

int

OPTIMISTIC =

0

;

public final

int

PESSIMISTIC =

2

;

public

double[][]

RegretMatrix;

public final

int

MOST\_LIKELY =

1

;

public final

int

SLR\_SLOW =

2

;

public final

int

SLR\_MEDIUM =

3

;

public final

int

SLR\_FAST =

4

;

public

double

PerceivedSSIntrusionByDH;

public

double[]

SSIntrusionDH;

public

int

RiskUpdateCount;

public

double

UnitCost;

public

double

FixedCost;

public

boolean

DesalinationApplied;

@AnyLogicInternalCodegenAPI

private static Map<String, IElementDescriptor> elementDesciptors\_xjal = createElementDescriptors( Agency.class );

@AnyLogicInternalCodegenAPI

@Override

public Map<String, IElementDescriptor> getElementDesciptors() {

return elementDesciptors\_xjal;

}

@AnyLogicCustomProposalPriority(type = AnyLogicCustomProposalPriority.Type.STATIC\_ELEMENT)

public static final Scale scale = new Scale( 10.0 );

@Override

public Scale getScale() {

return scale;

}

// Events

public EventTimeout CheckWellStatus = new EventTimeout(this);

public EventTimeout IncrementBudget = new EventTimeout(this);

public EventTimeout YearlyPerceivedSSIntrusion = new EventTimeout(this);

public EventTimeout RecoveryAdaptEvent = new EventTimeout(this);

@Override

@AnyLogicInternalCodegenAPI

public String getNameOf( EventTimeout \_e ) {

if( \_e == CheckWellStatus ) return "CheckWellStatus";

if( \_e == IncrementBudget ) return "IncrementBudget";

if( \_e == YearlyPerceivedSSIntrusion ) return "YearlyPerceivedSSIntrusion";

if( \_e == RecoveryAdaptEvent ) return "RecoveryAdaptEvent";

return super.getNameOf( \_e );

}

@Override

@AnyLogicInternalCodegenAPI

public EventTimeout.Mode getModeOf( EventTimeout \_e ) {

if ( \_e == CheckWellStatus ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

if ( \_e == IncrementBudget ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

if ( \_e == YearlyPerceivedSSIntrusion ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

if ( \_e == RecoveryAdaptEvent ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

return super.getModeOf( \_e );

}

@Override

@AnyLogicInternalCodegenAPI

public double getFirstOccurrenceTime( EventTimeout \_e ) {

double \_t;

if ( \_e == CheckWellStatus ) {

\_t =

0

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if ( \_e == IncrementBudget ) {

\_t =

0

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if ( \_e == YearlyPerceivedSSIntrusion ) {

\_t =

0

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if ( \_e == RecoveryAdaptEvent ) {

\_t =

0

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

return super.getFirstOccurrenceTime( \_e );

}

@Override

@AnyLogicInternalCodegenAPI

public double evaluateTimeoutOf( EventTimeout \_e ) {

double \_t;

if( \_e == CheckWellStatus) {

\_t =

DecisionHorizon

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if( \_e == IncrementBudget) {

\_t =

DecisionHorizon

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if( \_e == YearlyPerceivedSSIntrusion) {

\_t =

1

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if( \_e == RecoveryAdaptEvent) {

\_t =

1

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

return super.evaluateTimeoutOf( \_e );

}

@Override

@AnyLogicInternalCodegenAPI

public void executeActionOf( EventTimeout \_e ) {

if ( \_e == CheckWellStatus ) {

EventTimeout self = \_e;

//Update the SS Intrusion for the adaptation measure function

// and the risk attitude updating

for(int i=0; i < DecisionHorizon; i++){

SSIntrusionDH[i] = SSIntrusionPoisonFormula(getPerceivedLambda());

}

CheckAllWells();

//Risk updating starts at the first decision horizon

if((main.current\_model\_year > 0) && (RiskUpdateCount < RiskUpdateLimit)){

if(RiskAttitudeUpdateBoundedRationality()){

//RiskUpdateCount++;

}

}

;

return;

}

if ( \_e == IncrementBudget ) {

EventTimeout self = \_e;

Budget += CapitalPlanFund/(100/DecisionHorizon);

;

return;

}

if ( \_e == YearlyPerceivedSSIntrusion ) {

EventTimeout self = \_e;

PerceivedSSIntrusionByDH += SSIntrusionPoisonFormula(getPerceivedLambda());

;

return;

}

if ( \_e == RecoveryAdaptEvent ) {

EventTimeout self = \_e;

for (Agent a : main.agents()){

//Check agent is a Well

if (a instanceof Well) {

//Make a copy of the agent

Well well = (Well) a;

//if there is a well contaminated

if(well.isContaminated() && !RelocationAdapt(well)){

DesalinationAdapt(well);

}

}

}

;

return;

}

super.executeActionOf( \_e );

}

/\*\* Internal constant, shouldn't be accessed by user \*/

@AnyLogicInternalCodegenAPI

protected static final int \_STATECHART\_COUNT\_xjal = 1;

// Statecharts

public Statechart<RiskAttitudeStateChart\_state> RiskAttitudeStateChart = new Statechart<>( this, (short)2 );

@Override

@AnyLogicInternalCodegenAPI

public String getNameOf( Statechart \_s ) {

if(\_s == this.RiskAttitudeStateChart) return "RiskAttitudeStateChart";

return super.getNameOf( \_s );

}

@Override

@AnyLogicInternalCodegenAPI

public int getIdOf( Statechart \_s ) {

if(\_s == this.RiskAttitudeStateChart) return 0;

return super.getIdOf( \_s );

}

@Override

@AnyLogicInternalCodegenAPI

public void executeActionOf( Statechart \_s ) {

if( \_s == this.RiskAttitudeStateChart ) {

enterState( branch, true );

return;

}

super.executeActionOf( \_s );

}

// States of all statecharts

public enum RiskAttitudeStateChart\_state implements IStatechartState<Agency, RiskAttitudeStateChart\_state> {

RiskSeeker,

RiskNeutral,

RiskAverse,

branch;

@AnyLogicInternalCodegenAPI

private Collection<RiskAttitudeStateChart\_state> \_simpleStatesDeep\_xjal;

@AnyLogicInternalCodegenAPI

private Set<RiskAttitudeStateChart\_state> \_fullState\_xjal;

@AnyLogicInternalCodegenAPI

private Set<RiskAttitudeStateChart\_state> \_statesInside\_xjal;

@Override

@AnyLogicInternalCodegenAPI

public Collection<RiskAttitudeStateChart\_state> getSimpleStatesDeep() {

Collection<RiskAttitudeStateChart\_state> result = \_simpleStatesDeep\_xjal;

if (result == null) {

\_simpleStatesDeep\_xjal = result = calculateAllSimpleStatesDeep();

}

return result;

}

@Override

public Set<RiskAttitudeStateChart\_state> getFullState() {

Set<RiskAttitudeStateChart\_state> result = \_fullState\_xjal;

if (result == null) {

\_fullState\_xjal = result = calculateFullState();

}

return result;

}

@Override

@AnyLogicInternalCodegenAPI

public Set<RiskAttitudeStateChart\_state> getStatesInside() {

Set<RiskAttitudeStateChart\_state> result = \_statesInside\_xjal;

if (result == null) {

\_statesInside\_xjal = result = calculateStatesInside();

}

return result;

}

@Override

@AnyLogicInternalCodegenAPI

public Statechart<RiskAttitudeStateChart\_state> getStatechart( Agency \_a ) {

return \_a.RiskAttitudeStateChart;

}

}

@AnyLogicCustomProposalPriority(type = AnyLogicCustomProposalPriority.Type.STATIC\_ELEMENT)

public static final RiskAttitudeStateChart\_state RiskSeeker = RiskAttitudeStateChart\_state.RiskSeeker;

@AnyLogicCustomProposalPriority(type = AnyLogicCustomProposalPriority.Type.STATIC\_ELEMENT)

public static final RiskAttitudeStateChart\_state RiskNeutral = RiskAttitudeStateChart\_state.RiskNeutral;

@AnyLogicCustomProposalPriority(type = AnyLogicCustomProposalPriority.Type.STATIC\_ELEMENT)

public static final RiskAttitudeStateChart\_state RiskAverse = RiskAttitudeStateChart\_state.RiskAverse;

@AnyLogicCustomProposalPriority(type = AnyLogicCustomProposalPriority.Type.STATIC\_ELEMENT)

public static final RiskAttitudeStateChart\_state branch = RiskAttitudeStateChart\_state.branch;

@AnyLogicInternalCodegenAPI

private void enterState( RiskAttitudeStateChart\_state self, boolean \_destination ) {

switch( self ) {

case RiskSeeker:

logToDBEnterState(RiskAttitudeStateChart, self);

// (Simple state (not composite))

RiskAttitudeStateChart.setActiveState\_xjal( RiskSeeker );

transition3.start();

return;

case RiskNeutral:

logToDBEnterState(RiskAttitudeStateChart, self);

// (Simple state (not composite))

RiskAttitudeStateChart.setActiveState\_xjal( RiskNeutral );

transition4.start();

transition6.start();

return;

case RiskAverse:

logToDBEnterState(RiskAttitudeStateChart, self);

// (Simple state (not composite))

RiskAttitudeStateChart.setActiveState\_xjal( RiskAverse );

transition5.start();

return;

case branch:

logToDBEnterState(RiskAttitudeStateChart, self);

// (Branch)

if (

RiskAttitude == RISK\_SEEKER

) { // transition

enterState( RiskSeeker, true );

return;

}

if (

RiskAttitude == RISK\_NEUTRAL

) { // transition1

enterState( RiskNeutral, true );

return;

}

if (

RiskAttitude == RISK\_AVERSE

) { // transition2

enterState( RiskAverse, true );

return;

}

throw new RuntimeException( "All exits are closed at dynamic choice point 'branch'" );

default:

return;

}

}

@AnyLogicInternalCodegenAPI

private void exitState( RiskAttitudeStateChart\_state self, Transition \_t, boolean \_source ) {

switch( self ) {

case RiskSeeker:

logToDBExitState(RiskAttitudeStateChart, self);

logToDB(RiskAttitudeStateChart, \_t, self);

// (Simple state (not composite))

if ( !\_source || \_t != transition3) transition3.cancel();

return;

case RiskNeutral:

logToDBExitState(RiskAttitudeStateChart, self);

logToDB(RiskAttitudeStateChart, \_t, self);

// (Simple state (not composite))

if ( !\_source || \_t != transition4) transition4.cancel();

if ( !\_source || \_t != transition6) transition6.cancel();

return;

case RiskAverse:

logToDBExitState(RiskAttitudeStateChart, self);

logToDB(RiskAttitudeStateChart, \_t, self);

// (Simple state (not composite))

if ( !\_source || \_t != transition5) transition5.cancel();

return;

default:

return;

}

}

@AnyLogicInternalCodegenAPI

private void exitInnerStates( RiskAttitudeStateChart\_state \_destination ) {

RiskAttitudeStateChart\_state \_state = RiskAttitudeStateChart.getActiveSimpleState();

while( \_state != \_destination ) {

exitState( \_state, null, false );

\_state = \_state.getContainerState();

}

}

public TransitionMessage transition3 = new TransitionMessage( this );

public TransitionMessage transition4 = new TransitionMessage( this );

public TransitionMessage transition5 = new TransitionMessage( this );

public TransitionMessage transition6 = new TransitionMessage( this );

@Override

@AnyLogicInternalCodegenAPI

public String getNameOf( TransitionMessage \_t ) {

if ( \_t == transition3 ) return "transition3";

if ( \_t == transition4 ) return "transition4";

if ( \_t == transition5 ) return "transition5";

if ( \_t == transition6 ) return "transition6";

return super.getNameOf( \_t );

}

@Override

@AnyLogicInternalCodegenAPI

public Statechart getStatechartOf( TransitionMessage \_t ) {

if ( \_t == transition3 ) return RiskAttitudeStateChart;

if ( \_t == transition4 ) return RiskAttitudeStateChart;

if ( \_t == transition5 ) return RiskAttitudeStateChart;

if ( \_t == transition6 ) return RiskAttitudeStateChart;

return super.getStatechartOf( \_t );

}

@Override

@AnyLogicInternalCodegenAPI

public void executeActionOf( TransitionMessage self, Object \_msg ) {

if ( self == transition3 ) {

exitState( RiskSeeker, self, true );

{

Object msg = (Object) \_msg;

RiskUpdateCount++;

;}

enterState( RiskNeutral, true );

return;

}

if ( self == transition4 ) {

exitState( RiskNeutral, self, true );

{

Object msg = (Object) \_msg;

RiskUpdateCount++;

;}

enterState( RiskAverse, true );

return;

}

if ( self == transition5 ) {

exitState( RiskAverse, self, true );

{

Object msg = (Object) \_msg;

RiskUpdateCount++;

;}

enterState( RiskNeutral, true );

return;

}

if ( self == transition6 ) {

exitState( RiskNeutral, self, true );

{

Object msg = (Object) \_msg;

RiskUpdateCount++;

;}

enterState( RiskSeeker, true );

return;

}

super.executeActionOf( self, \_msg );

}

@Override

@AnyLogicInternalCodegenAPI

public boolean testMessageOf( TransitionMessage \_t, Object \_msg ) {

if ( \_t == transition3 ) {

Object

msg = (Object) \_msg;

Object \_g =

"averse"

;

return msg.equals( \_g );

}

if ( \_t == transition4 ) {

Object

msg = (Object) \_msg;

Object \_g =

"averse"

;

return msg.equals( \_g );

}

if ( \_t == transition5 ) {

Object

msg = (Object) \_msg;

Object \_g =

"seeker"

;

return msg.equals( \_g );

}

if ( \_t == transition6 ) {

Object

msg = (Object) \_msg;

Object \_g =

"seeker"

;

return msg.equals( \_g );

}

return super.testMessageOf( \_t, \_msg );

}

// Functions

void CheckAllWells( ) {

//If the 'no adaptation' strategy is not selected

if(main.ChosenNoAdaptation == 0) {

//Go through all the agents in Main

for (Agent a : main.agents()){

//Check agent is a Well

if (a instanceof Well) {

//Make a copy of the agent

Well well = (Well) a;

//Check well water is still potable at risk of both SS and SWIL

//And the budget is sufficient

if(well.isPotable() && WellRiskSSandSWIL(well) && (Budget > 0))

{ System.out.println("Well at risk: "+well.Name);

RegretTheory(well);

int adapt\_measure = -1;

if(RiskAttitude == RISK\_AVERSE){

adapt\_measure = RiskAverseAM();

}

else if(RiskAttitude == RISK\_SEEKER){

adapt\_measure = RiskSeekerAM();

}

else if(RiskAttitude == RISK\_NEUTRAL){

adapt\_measure = RiskNeutralAM();

}

//Apply adaptation measure if chosen

double cost = getAdaptationMeasureCost(adapt\_measure);

if(cost <= Budget){

if(adapt\_measure == 3){

well.increaseRelocationSWIL(getAdaptationMeasureSwil(adapt\_measure));

} else if(adapt\_measure == 4){

well.increaseDesCapacity(DesalinatedWater());

System.out.println("Desalination adaptation cost: "+cost);

if(!DesalinationApplied){

FixedCost=0;

}

DesalinationApplied=true;

}

else{

well.increaseLocFromSWIL(getAdaptationMeasureSwil(adapt\_measure));

}

main.AdaptationYear.add(adapt\_measure);

Budget -= cost;

well.setRecoveryDelay(adapt\_measure);

}

//If chosen adaptation is desalination...

}

}

}

}

}

/\*\*

\* Check if well is at risk of both SWIL and Storm Surge

\*/

boolean

WellRiskSSandSWIL( Well well ) {

/\* Check if well is at risk of both storm surge and the

\* salt water intrusion line

\*/

double perceived\_swil = main.Stressor.getPerceivedSWIL(perceivedSLR(),

perceivedCase(), (int) main.current\_model\_year);

perceived\_swil = perceived\_swil + PerceivedSSIntrusionByDH;

if((well.CurrentLocFromSWIL - perceived\_swil) <= getSWILThreshold()){

System.out.println("WellRiskSSandSWIL perceived swil: "+perceived\_swil);

PerceivedSSIntrusionByDH = 0;

return true;

}

else{

PerceivedSSIntrusionByDH = 0;

return false;

}

}

int

perceivedCase( ) {

if(CurrentRiskAttitude() == RISK\_AVERSE){

return PESSIMISTIC;

}

else if(CurrentRiskAttitude() == RISK\_NEUTRAL){

return MOST\_LIKELY;

}

else{ //RISK\_SEEKER

return OPTIMISTIC;

}

}

double

CheckReliabilityOfAdoption( int case\_scenario, int adaptation\_id, Well well ) {

int adapt\_intrusion\_line = getAdaptationMeasureSwil(adaptation\_id);

int slr\_scenario = getSLRfromCase(case\_scenario);

//\*\*Set planned adaptation

well.setPlanAdaptIntrusionLine(adapt\_intrusion\_line);

well.setPlannedDelay(adaptation\_id);

//\*\*

if(adaptation\_id == 4){ // Desalination adaptation

//main.WTPSouthMiamiHeights.setPlannedDC(well.ExtractionCapacity);

well.setPlannedDesalinationCapacity(DesalinatedWater());

}

double accu\_extraction = PlannedSupplyAllWells(slr\_scenario, case\_scenario, adaptation\_id);

//\*\*Unset planned adaptation

well.setPlanAdaptIntrusionLine(0);

//main.WTPSouthMiamiHeights.setPlannedDC(0);

well.setPlannedDesalinationCapacity(0);

well.setPlannedDelay2Zero();

//\*\*

double planned\_demand = main.UserSouthMiamiDade.getPlannedConsumption();

planned\_demand += main.total\_demand; //add the current demand

accu\_extraction += main.total\_level\_of\_supply; //add the current total level of supply

if(planned\_demand == 0)

return 0.0;

return accu\_extraction/planned\_demand;

}

double

getAdaptationMeasureCost( int am\_id ) {

if(am\_id == 4){

return DesalinatedCost();

}

return (double)selectFrom( adaptation ).

where( adaptation.id.eq( am\_id ) ).

firstResult( adaptation.cost\_dollars );

/\*

return (double) selectFirstValue(int.class,

"SELECT cost\_dollars FROM adaptation WHERE " +

"id = ?; ",

am\_id

);\*/

}

int

getAdaptationMeasureSwil( int am\_id ) {

return (int) selectFirstValue(int.class,

"SELECT adaptation\_intrusion\_line\_ft FROM adaptation WHERE " +

"id = ?; ",

am\_id

);

}

String

getAdaptationMeasureName( int am\_id ) {

return (String) selectFirstValue(String.class,

"SELECT name FROM adaptation WHERE " +

"id = ?; ",

am\_id

);

}

void BuildReliabilityMatrix( Well well ) {

//System.out.println(i+") "+CheckReliabilityOfAdoption(perceivedCase(), i, well));

int total\_adaptations = TotalAdaptations();

for(int i = 0; i < total\_adaptations; i++){

for(int j = 0; j < 3; j++){

ReliabilityMatrix[i][j] = CheckReliabilityOfAdoption(j, i+1, well);

}

}

}

int

TotalAdaptations( ) {

return (int) selectFirstValue(true, int.class,

"SELECT COUNT(id) FROM adaptation;"

);

}

void RegretTheory( Well well ) {

BuildReliabilityMatrix(well);

//int result[] = getMaxPositionFromMatrix(ReliabilityMatrix);

int total\_adaptations = TotalAdaptations()-1;

//Calculate the regret

for(int i = 0; i < total\_adaptations; i++){

for(int j = 0; j < 3; j++){

RegretMatrix[i][j] = getMaxReliabilityCase(j) - ReliabilityMatrix[i][j];

}

}

}

double

PlannedSupplyAllWells( int slr\_scenario, int case\_scenario, int adap\_id ) {

double accu\_extraction=0;

for(int i = 1; i <= DecisionHorizon; i++){

double perceived\_ss\_intrusion = SSIntrusionDH[i-1];

double planned\_demmand=main.UserSouthMiamiDade.getPlannedConsumptionYear(i);

double well\_supply=0;

//Loop through all the agents

for (Agent a : main.agents()){

//Check agent is a Well

if (a instanceof Well) {

Well well = ((Well) a);

double well\_planned\_extraction = well.getPlannedExtractionCapacity(slr\_scenario, case\_scenario, i,

perceived\_ss\_intrusion);

//Save values to debug reliability index computation

//main.SaveRow(adap\_id, case\_scenario, slr\_scenario,

//main.Stressor.getPerceivedSWIL(slr\_scenario, case\_scenario,

//(int) main.current\_model\_year+i),

//perceived\_ss\_intrusion, (int) main.current\_model\_year+i,

//well.Name, well.CurrentLocFromSWIL, well\_planned\_extraction,

//well.PlannedAdaptIntrusionLine);

well\_supply += well\_planned\_extraction;

}

}

if(well\_supply > main.WTPSouthMiamiHeights.TreatmentCapacity)

well\_supply = main.WTPSouthMiamiHeights.TreatmentCapacity\*365;

else

well\_supply = well\_supply\*365;

if(well\_supply > planned\_demmand)

well\_supply = planned\_demmand;

accu\_extraction+=well\_supply;

}

return accu\_extraction;

}

double

getMaxReliabilityCase( int case\_scenario ) {

double maxValue = ReliabilityMatrix[0][case\_scenario];

for (int i = 0; i < ReliabilityMatrix.length; i++) {

if (ReliabilityMatrix[i][case\_scenario] > maxValue) {

maxValue = ReliabilityMatrix[i][case\_scenario];

}

}

return maxValue;

}

int

CurrentRiskAttitude( ) {

if(RiskAttitudeStateChart.getActiveSimpleState() == RiskAverse){

return RISK\_AVERSE;

}

else if(RiskAttitudeStateChart.getActiveSimpleState() == RiskNeutral){

return RISK\_NEUTRAL;

}

else{

return RISK\_SEEKER;

}

}

int

perceivedSLR( ) {

if(CurrentRiskAttitude() == RISK\_AVERSE){

return SLR\_FAST;

}

else if(CurrentRiskAttitude() == RISK\_NEUTRAL){

return SLR\_MEDIUM;

}

else{

return SLR\_SLOW;

}

}

double

getPerceivedLambda( ) {

return (double) selectFrom(ss\_impact)

.where(ss\_impact.slr\_scenario.eq(perceivedSLR()))

.uniqueResult(ss\_impact.lambda);

}

double

getSWILThreshold( ) {

return RiskThreshold;

//(double) selectFrom(risk\_attitudes\_thresholds)

//.where(risk\_attitudes\_thresholds.id.eq(CurrentRiskAttitude()))

//.uniqueResult(risk\_attitudes\_thresholds.swil\_threshold);

}

int

getSLRfromCase( int case\_scenario ) {

if(case\_scenario == OPTIMISTIC){

return SLR\_SLOW;

}

else if(case\_scenario == MOST\_LIKELY){

return SLR\_MEDIUM;

}

else{

return SLR\_FAST;

}

}

int

RiskSeekerAM( ) {

int maxPSRAM = getMaxReliabilityAM(OPTIMISTIC);

double maxPSRValue = ReliabilityMatrix[maxPSRAM][OPTIMISTIC];

double minCost = getAdaptationMeasureCost(maxPSRAM+1);

for (int i = 0; i < TotalAdaptations(); i++) {

if (ReliabilityMatrix[i][OPTIMISTIC] == maxPSRValue) {

if (minCost > getAdaptationMeasureCost(i+1)){

minCost = getAdaptationMeasureCost(i+1);

maxPSRAM = i;

}

}

}

return maxPSRAM+1;

}

int

RiskAverseAM( ) {

int maxPSRAM = getMaxReliabilityAM(PESSIMISTIC);

double maxPSRValue = ReliabilityMatrix[maxPSRAM][PESSIMISTIC];

double minCost = getAdaptationMeasureCost(maxPSRAM+1);

for (int i = 0; i < TotalAdaptations(); i++) {

if (ReliabilityMatrix[i][PESSIMISTIC] == maxPSRValue) {

if (minCost > getAdaptationMeasureCost(i+1)){

minCost = getAdaptationMeasureCost(i+1);

maxPSRAM = i;

}

}

}

return maxPSRAM+1;

}

int

RiskNeutralAM( ) {

int total\_adaptations = TotalAdaptations();

double max\_reg[] = new double[total\_adaptations];

//Calculate the max regret

for(int i = 0; i < total\_adaptations; i++){

max\_reg[i] = RegretMatrix[i][0];

for(int j = 0; j < 3; j++){

if(RegretMatrix[i][j] > max\_reg[i]){

max\_reg[i] = RegretMatrix[i][j];

}

}

}

double min\_max\_reg = max\_reg[0];

int min\_max\_reg\_i = 0;

for(int i = 0; i < total\_adaptations; i++){

if(max\_reg[i] < min\_max\_reg){

min\_max\_reg = max\_reg[i];

min\_max\_reg\_i = i;

}

}

//Get the adaptation measure with minimum cost

//and the similar reliability index

double minCost = getAdaptationMeasureCost(min\_max\_reg\_i+1);

for (int i = 0; i < total\_adaptations; i++) {

if (max\_reg[i] == min\_max\_reg) {

if (minCost > getAdaptationMeasureCost(i+1)){

minCost = getAdaptationMeasureCost(i+1);

min\_max\_reg\_i = i;

}

}

}

return min\_max\_reg\_i+1;

}

int

getMaxReliabilityAM( int case\_scenario ) {

double maxValue = ReliabilityMatrix[0][case\_scenario];

int Maxi = 0;

for (int i = 0; i < ReliabilityMatrix.length; i++) {

if (ReliabilityMatrix[i][case\_scenario] > maxValue) {

maxValue = ReliabilityMatrix[i][case\_scenario];

Maxi = i;

}

}

return Maxi;

}

double

SSIntrusionPoisonFormula( double lambda ) {

//return uniform(100, 300);

Random rand = new Random();

int result = poisson(lambda, rand);

if(result == 1){

return uniform(100, 300);

} else{

return 0.0;

}

}

boolean

contaminatedWell( ) {

for (Agent a : main.agents()){

//Check agent is a Well

if (a instanceof Well) {

//If one of the wells is contaminated

//return true

Well well = (Well) a;

if(well.isContaminated()){

return true;

}

}

}

return false;

}

void updateAdapCostDB( ) {

//Update the adaptation cost in the database

//using the uniform pert formula

for(int i=1; i<=TotalAdaptations(); i++){

double min = (double)selectFrom( adaptation ).

where( adaptation.id.eq( i ) ).

firstResult( adaptation.pert\_min );

double max = (double)selectFrom( adaptation ).

where( adaptation.id.eq( i ) ).

firstResult( adaptation.pert\_max );

double mode = (double)selectFrom( adaptation ).

where( adaptation.id.eq( i ) ).

firstResult( adaptation.pert\_mode );

update(adaptation)

.where(adaptation.id.eq( i ))

.set(adaptation.cost\_dollars, (double) pert(min, max, mode))

.execute();

}

//Update the adaptation intrusion line for the

//first two adaptation measures

for(int i=1; i<=2; i++){

double min\_line = (double)selectFrom( adaptation ).

where( adaptation.id.eq( i ) ).

firstResult( adaptation.min\_intrusion\_line\_ft );

double max\_line = (double)selectFrom( adaptation ).

where( adaptation.id.eq( i ) ).

firstResult( adaptation.max\_intrusion\_line\_ft );

update(adaptation)

.where(adaptation.id.eq( i ))

.set(adaptation.adaptation\_intrusion\_line\_ft, (double) uniform(min\_line, max\_line))

.execute();

}

}

String

getRiskAttitudeName( int id ) {

return (String) selectFirstValue(String.class,

"SELECT risk\_attitude FROM risk\_attitudes\_thresholds WHERE " +

"id = ?; ",

id

);

}

void DesalinationAdapt( Well well ) {

double desalination\_cap = DesalinatedWater();

double desalination\_cost = DesalinatedCost();

if((desalination\_cost <= Budget)){

//well.increaseWellRecoveryCount();

well.increaseDesCapacity(desalination\_cap);

main.AdaptationYear.add(4);

Budget -= desalination\_cost;

if(!DesalinationApplied){

FixedCost=0;

}

DesalinationApplied=true;

//System.out.println("Desalination recovery cost: "+desalination\_cost);

}

}

boolean

RelocationAdapt( Well well ) {

double rel\_cost = getAdaptationMeasureCost(3);

double rel\_swil = getAdaptationMeasureSwil(3);

if((rel\_cost <= Budget) && well.wellRecoveryLimit()){

well.increaseWellRecoveryCount();

well.increaseRelocationSWIL(rel\_swil);

main.AdaptationYear.add(3);

Budget -= rel\_cost;

return true;

}

return false;

}

boolean

RiskAttitudeUpdateBoundedRationality( ) {

int year1 = yearBase10((int)main.current\_model\_year - DecisionHorizon);

int year2 = yearBase10((int)main.current\_model\_year - DecisionHorizon);

if(DecisionHorizon > 10){

year2 = yearBase10((int)main.current\_model\_year - 10);

}

int aswi\_min = (int) selectFrom(slr\_impact)

.where(slr\_impact.slr\_id.eq(main.ChosenSLRScenario))

.where(slr\_impact.year.goe(year1))

.where(slr\_impact.year.loe(year2))

.orderBy(slr\_impact.line\_value.asc())

.firstResult(slr\_impact.line\_value);

int aswi\_max = (int) selectFrom(slr\_impact)

.where(slr\_impact.slr\_id.eq(main.ChosenSLRScenario))

.where(slr\_impact.year.goe(year1))

.where(slr\_impact.year.loe(year2))

.orderBy(slr\_impact.line\_value.desc())

.firstResult(slr\_impact.line\_value);

int pswi = main.Stressor.getPerceivedSWIL(perceivedSLR(),

perceivedCase(),

(int) main.current\_model\_year);

main.pswi\_dataset.add(pswi);

main.aswi\_max\_dataset.add(aswi\_max);

main.aswi\_min\_dataset.add(aswi\_min);

if(pswi >= aswi\_min && pswi <= aswi\_max){

return false; //no change

}

else if(pswi < aswi\_min){

RiskAttitudeStateChart.fireEvent("averse");

return true;

}

else if(pswi > aswi\_max){

RiskAttitudeStateChart.fireEvent("seeker");

return true;

}

else{

return false;

}

}

double

DesalinatedWater( ) {

double demand\_minus\_ncw = min(max(main.year\_demand - main.year\_supply, 0), 20000000);

//System.out.println("Demand minus Supply: "+(main.year\_demand - main.year\_supply));

return min(demand\_minus\_ncw, Budget/UnitCost);

}

double

DesalinatedCost( ) {

double dcost = FixedCost + (UnitCost \* DesalinatedWater());

return dcost;

}

int

yearBase10( int year ) {

return main.Stressor.getYearBase10(year);

}

// View areas

public ViewArea \_origin\_VA = new ViewArea( this, "[Origin]", 0, 0, 1020.0, 640.0 );

@Override

@AnyLogicInternalCodegenAPI

public int getViewAreas(Map<String, ViewArea> \_output) {

if ( \_output != null ) {

\_output.put( "\_origin\_VA", this.\_origin\_VA );

}

return 1 + super.getViewAreas( \_output );

}

@AnyLogicInternalCodegenAPI

protected static final Font \_text\_Font = new Font("SansSerif", 0, 12 );

@AnyLogicInternalCodegenAPI

protected static final Font \_text1\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text2\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text3\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text4\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text5\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text6\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text7\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text8\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text9\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text10\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text11\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text12\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text13\_Font = new Font("SansSerif", 0, 14 );

@AnyLogicInternalCodegenAPI

protected static final Font \_text14\_Font = \_text13\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text15\_Font = \_text13\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text16\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text17\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text18\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text19\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text20\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text21\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text22\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text23\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text24\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text25\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text26\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text27\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text28\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text29\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text30\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text31\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text32\_Font = \_text13\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text33\_Font = \_text13\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text34\_Font = \_text13\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text35\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text36\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text37\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text38\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text39\_Font = new Font("SansSerif", 0, 10 );

@AnyLogicInternalCodegenAPI

protected static final Font \_text40\_Font = \_text39\_Font;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle = 1;

@AnyLogicInternalCodegenAPI

protected static final int \_text = 2;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle1 = 3;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle4 = 4;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle7 = 5;

@AnyLogicInternalCodegenAPI

protected static final int \_text1 = 6;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle8 = 7;

@AnyLogicInternalCodegenAPI

protected static final int \_text2 = 8;

@AnyLogicInternalCodegenAPI

protected static final int \_text3 = 9;

@AnyLogicInternalCodegenAPI

protected static final int \_text4 = 10;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle2 = 11;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle5 = 12;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle9 = 13;

@AnyLogicInternalCodegenAPI

protected static final int \_text5 = 14;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle10 = 15;

@AnyLogicInternalCodegenAPI

protected static final int \_text6 = 16;

@AnyLogicInternalCodegenAPI

protected static final int \_text7 = 17;

@AnyLogicInternalCodegenAPI

protected static final int \_text8 = 18;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle3 = 19;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle6 = 20;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle11 = 21;

@AnyLogicInternalCodegenAPI

protected static final int \_text9 = 22;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle12 = 23;

@AnyLogicInternalCodegenAPI

protected static final int \_text10 = 24;

@AnyLogicInternalCodegenAPI

protected static final int \_text11 = 25;

@AnyLogicInternalCodegenAPI

protected static final int \_text12 = 26;

@AnyLogicInternalCodegenAPI

protected static final int \_text13 = 27;

@AnyLogicInternalCodegenAPI

protected static final int \_text14 = 28;

@AnyLogicInternalCodegenAPI

protected static final int \_text15 = 29;

@AnyLogicInternalCodegenAPI

protected static final int \_text16 = 30;

@AnyLogicInternalCodegenAPI

protected static final int \_text17 = 31;

@AnyLogicInternalCodegenAPI

protected static final int \_text18 = 32;

@AnyLogicInternalCodegenAPI

protected static final int \_text19 = 33;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle13 = 34;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle14 = 35;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle15 = 36;

@AnyLogicInternalCodegenAPI

protected static final int \_text20 = 37;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle16 = 38;

@AnyLogicInternalCodegenAPI

protected static final int \_text21 = 39;

@AnyLogicInternalCodegenAPI

protected static final int \_text22 = 40;

@AnyLogicInternalCodegenAPI

protected static final int \_text23 = 41;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle17 = 42;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle18 = 43;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle19 = 44;

@AnyLogicInternalCodegenAPI

protected static final int \_text24 = 45;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle20 = 46;

@AnyLogicInternalCodegenAPI

protected static final int \_text25 = 47;

@AnyLogicInternalCodegenAPI

protected static final int \_text26 = 48;

@AnyLogicInternalCodegenAPI

protected static final int \_text27 = 49;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle21 = 50;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle22 = 51;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle23 = 52;

@AnyLogicInternalCodegenAPI

protected static final int \_text28 = 53;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle24 = 54;

@AnyLogicInternalCodegenAPI

protected static final int \_text29 = 55;

@AnyLogicInternalCodegenAPI

protected static final int \_text30 = 56;

@AnyLogicInternalCodegenAPI

protected static final int \_text31 = 57;

@AnyLogicInternalCodegenAPI

protected static final int \_text32 = 58;

@AnyLogicInternalCodegenAPI

protected static final int \_text33 = 59;

@AnyLogicInternalCodegenAPI

protected static final int \_text34 = 60;

@AnyLogicInternalCodegenAPI

protected static final int \_text35 = 61;

@AnyLogicInternalCodegenAPI

protected static final int \_text36 = 62;

@AnyLogicInternalCodegenAPI

protected static final int \_text37 = 63;

@AnyLogicInternalCodegenAPI

protected static final int \_text38 = 64;

@AnyLogicInternalCodegenAPI

protected static final int \_text39 = 65;

@AnyLogicInternalCodegenAPI

protected static final int \_text40 = 66;

/\*\* Internal constant, shouldn't be accessed by user \*/

@AnyLogicInternalCodegenAPI

protected static final int \_SHAPE\_NEXT\_ID\_xjal = 67;

@AnyLogicInternalCodegenAPI

public boolean isPublicPresentationDefined() {

return true;

}

@AnyLogicInternalCodegenAPI

public boolean isEmbeddedAgentPresentationVisible( Agent \_a ) {

return super.isEmbeddedAgentPresentationVisible( \_a );

}

@AnyLogicInternalCodegenAPI

private void \_initialize\_level\_xjal() {

level.addAll(rectangle, text, rectangle1, rectangle4, rectangle7, text1, rectangle8, text2, text3, text4, rectangle2, rectangle5, rectangle9, text5, rectangle10, text6, text7, text8, rectangle3, rectangle6, rectangle11, text9, rectangle12, text10, text11, text12, text13, text14, text15, text16, text17, text18, text19, rectangle13, rectangle14, rectangle15, text20, rectangle16, text21, text22, text23, rectangle17, rectangle18, rectangle19, text24, rectangle20, text25, text26, text27, rectangle21, rectangle22, rectangle23, text28, rectangle24, text29, text30, text31, text32, text33, text34, text35, text36, text37, text38, text39, text40);

}

protected ShapeRectangle rectangle;

protected ShapeText text;

protected ShapeRectangle rectangle1;

protected ShapeRectangle rectangle4;

protected ShapeRectangle rectangle7;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text1\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("%.4f", ReliabilityMatrix[0][0]\*100)

);

}

protected ShapeText text1;

protected ShapeRectangle rectangle8;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text2\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("%.4f", ReliabilityMatrix[1][0]\*100)

);

}

protected ShapeText text2;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text3\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("%.4f", ReliabilityMatrix[2][0]\*100)

);

}

protected ShapeText text3;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text4\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("%.4f", ReliabilityMatrix[3][0]\*100)

);

}

protected ShapeText text4;

protected ShapeRectangle rectangle2;

protected ShapeRectangle rectangle5;

protected ShapeRectangle rectangle9;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text5\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("%.4f", ReliabilityMatrix[0][1]\*100)

);

}

protected ShapeText text5;

protected ShapeRectangle rectangle10;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text6\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("%.4f", ReliabilityMatrix[1][1]\*100)

);

}

protected ShapeText text6;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text7\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("%.4f", ReliabilityMatrix[2][1]\*100)

);

}

protected ShapeText text7;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text8\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("%.4f", ReliabilityMatrix[3][1]\*100)

);

}

protected ShapeText text8;

protected ShapeRectangle rectangle3;

protected ShapeRectangle rectangle6;

protected ShapeRectangle rectangle11;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text9\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("%.4f", ReliabilityMatrix[0][2]\*100)

);

}

protected ShapeText text9;

protected ShapeRectangle rectangle12;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text10\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("%.4f", ReliabilityMatrix[1][2]\*100)

);

}

protected ShapeText text10;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text11\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("%.4f", ReliabilityMatrix[2][2]\*100)

);

}

protected ShapeText text11;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text12\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("%.4f", ReliabilityMatrix[3][2]\*100)

);

}

protected ShapeText text12;

protected ShapeText text13;

protected ShapeText text14;

protected ShapeText text15;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text16\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

getAdaptationMeasureName(4)

);

}

protected ShapeText text16;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text17\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

getAdaptationMeasureName(3)

);

}

protected ShapeText text17;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text18\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

getAdaptationMeasureName(2)

);

}

protected ShapeText text18;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text19\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

getAdaptationMeasureName(1)

);

}

protected ShapeText text19;

protected ShapeRectangle rectangle13;

protected ShapeRectangle rectangle14;

protected ShapeRectangle rectangle15;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text20\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("%.4f", RegretMatrix[0][0]\*100)

);

}

protected ShapeText text20;

protected ShapeRectangle rectangle16;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text21\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("%.4f", RegretMatrix[1][0]\*100)

);

}

protected ShapeText text21;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text22\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("%.4f", RegretMatrix[2][0]\*100)

);

}

protected ShapeText text22;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text23\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("%.4f", RegretMatrix[3][0]\*100)

);

}

protected ShapeText text23;

protected ShapeRectangle rectangle17;

protected ShapeRectangle rectangle18;

protected ShapeRectangle rectangle19;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text24\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("%.4f", RegretMatrix[0][1]\*100)

);

}

protected ShapeText text24;

protected ShapeRectangle rectangle20;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text25\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("%.4f", RegretMatrix[1][1]\*100)

);

}

protected ShapeText text25;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text26\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("%.4f", RegretMatrix[2][1]\*100)

);

}

protected ShapeText text26;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text27\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("%.4f", RegretMatrix[3][1]\*100)

);

}

protected ShapeText text27;

protected ShapeRectangle rectangle21;

protected ShapeRectangle rectangle22;

protected ShapeRectangle rectangle23;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text28\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("%.4f", RegretMatrix[0][2]\*100)

);

}

protected ShapeText text28;

protected ShapeRectangle rectangle24;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text29\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("%.4f", RegretMatrix[1][2]\*100)

);

}

protected ShapeText text29;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text30\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("%.4f", RegretMatrix[2][2]\*100)

);

}

protected ShapeText text30;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text31\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("%.4f", RegretMatrix[3][2]\*100)

);

}

protected ShapeText text31;

protected ShapeText text32;

protected ShapeText text33;

protected ShapeText text34;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text35\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

getAdaptationMeasureName(4)

);

}

protected ShapeText text35;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text36\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

getAdaptationMeasureName(3)

);

}

protected ShapeText text36;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text37\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

getAdaptationMeasureName(2)

);

}

protected ShapeText text37;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text38\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

getAdaptationMeasureName(1)

);

}

protected ShapeText text38;

protected ShapeText text39;

protected ShapeText text40;

protected com.anylogic.engine.markup.Level level;

private com.anylogic.engine.markup.Level[] \_getLevels\_xjal;

@Override

public com.anylogic.engine.markup.Level[] getLevels() {

return \_getLevels\_xjal;

}

@AnyLogicInternalCodegenAPI

private void \_createPersistentElementsBP0\_xjal() {

rectangle = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,290.0, 40.0, 0.0, 0.0,

black, null,

700.0, 320.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

text = new ShapeText(

SHAPE\_DRAW\_2D, true,790.0, 50.0, 0.0, 0.0,

black,"Reliability Matrix and Regret Theory",

\_text\_Font, ALIGNMENT\_LEFT );

rectangle1 = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,410.0, 190.0, 0.0, 0.0,

black, white,

60.0, 40.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

rectangle4 = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,410.0, 230.0, 0.0, 0.0,

black, white,

60.0, 40.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

rectangle7 = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,410.0, 270.0, 0.0, 0.0,

black, white,

60.0, 40.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

text1 = new ShapeText(

SHAPE\_DRAW\_2D, true,420.0, 200.0, 0.0, 0.0,

black,"text",

\_text1\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text1\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

rectangle8 = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,410.0, 310.0, 0.0, 0.0,

black, white,

60.0, 40.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

text2 = new ShapeText(

SHAPE\_DRAW\_2D, true,420.0, 240.0, 0.0, 0.0,

black,"text",

\_text2\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text2\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text3 = new ShapeText(

SHAPE\_DRAW\_2D, true,420.0, 280.0, 0.0, 0.0,

black,"text",

\_text3\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text3\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text4 = new ShapeText(

SHAPE\_DRAW\_2D, true,420.0, 320.0, 0.0, 0.0,

black,"text",

\_text4\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text4\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

rectangle2 = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,470.0, 190.0, 0.0, 0.0,

black, white,

60.0, 40.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

rectangle5 = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,470.0, 230.0, 0.0, 0.0,

black, white,

60.0, 40.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

rectangle9 = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,470.0, 270.0, 0.0, 0.0,

black, white,

60.0, 40.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

text5 = new ShapeText(

SHAPE\_DRAW\_2D, true,480.0, 200.0, 0.0, 0.0,

black,"text",

\_text5\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text5\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

rectangle10 = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,470.0, 310.0, 0.0, 0.0,

black, white,

60.0, 40.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

text6 = new ShapeText(

SHAPE\_DRAW\_2D, true,480.0, 240.0, 0.0, 0.0,

black,"text",

\_text6\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text6\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text7 = new ShapeText(

SHAPE\_DRAW\_2D, true,480.0, 280.0, 0.0, 0.0,

black,"text",

\_text7\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text7\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text8 = new ShapeText(

SHAPE\_DRAW\_2D, true,480.0, 320.0, 0.0, 0.0,

black,"text",

\_text8\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text8\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

rectangle3 = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,530.0, 190.0, 0.0, 0.0,

black, white,

60.0, 40.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

rectangle6 = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,530.0, 230.0, 0.0, 0.0,

black, white,

60.0, 40.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

rectangle11 = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,530.0, 270.0, 0.0, 0.0,

black, white,

60.0, 40.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

text9 = new ShapeText(

SHAPE\_DRAW\_2D, true,540.0, 200.0, 0.0, 0.0,

black,"text",

\_text9\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text9\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

rectangle12 = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,530.0, 310.0, 0.0, 0.0,

black, white,

60.0, 40.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

text10 = new ShapeText(

SHAPE\_DRAW\_2D, true,540.0, 240.0, 0.0, 0.0,

black,"text",

\_text10\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text10\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text11 = new ShapeText(

SHAPE\_DRAW\_2D, true,540.0, 280.0, 0.0, 0.0,

black,"text",

\_text11\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text11\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text12 = new ShapeText(

SHAPE\_DRAW\_2D, true,540.0, 320.0, 0.0, 0.0,

black,"text",

\_text12\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text12\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text13 = new ShapeText(

SHAPE\_DRAW\_2D, true,430.0, 170.0, 0.0, 0.0,

black,"O",

\_text13\_Font, ALIGNMENT\_LEFT );

text14 = new ShapeText(

SHAPE\_DRAW\_2D, true,490.0, 170.0, 0.0, 0.0,

black,"M",

\_text14\_Font, ALIGNMENT\_LEFT );

text15 = new ShapeText(

SHAPE\_DRAW\_2D, true,550.0, 170.0, 0.0, 0.0,

black,"P",

\_text15\_Font, ALIGNMENT\_LEFT );

text16 = new ShapeText(

SHAPE\_DRAW\_2D, true,390.0, 320.0, 0.0, 0.0,

black,"text",

\_text16\_Font, ALIGNMENT\_RIGHT ) {

@Override

public void updateDynamicProperties() {

\_text16\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text17 = new ShapeText(

SHAPE\_DRAW\_2D, true,390.0, 280.0, 0.0, 0.0,

black,"text",

\_text17\_Font, ALIGNMENT\_RIGHT ) {

@Override

public void updateDynamicProperties() {

\_text17\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text18 = new ShapeText(

SHAPE\_DRAW\_2D, true,390.0, 240.0, 0.0, 0.0,

black,"text",

\_text18\_Font, ALIGNMENT\_RIGHT ) {

@Override

public void updateDynamicProperties() {

\_text18\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text19 = new ShapeText(

SHAPE\_DRAW\_2D, true,390.0, 200.0, 0.0, 0.0,

black,"text",

\_text19\_Font, ALIGNMENT\_RIGHT ) {

@Override

public void updateDynamicProperties() {

\_text19\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

rectangle13 = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,800.0, 190.0, 0.0, 0.0,

black, white,

60.0, 40.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

rectangle14 = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,800.0, 230.0, 0.0, 0.0,

black, white,

60.0, 40.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

rectangle15 = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,800.0, 270.0, 0.0, 0.0,

black, white,

60.0, 40.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

text20 = new ShapeText(

SHAPE\_DRAW\_2D, true,810.0, 200.0, 0.0, 0.0,

black,"text",

\_text20\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text20\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

rectangle16 = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,800.0, 310.0, 0.0, 0.0,

black, white,

60.0, 40.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

text21 = new ShapeText(

SHAPE\_DRAW\_2D, true,810.0, 240.0, 0.0, 0.0,

black,"text",

\_text21\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text21\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text22 = new ShapeText(

SHAPE\_DRAW\_2D, true,810.0, 280.0, 0.0, 0.0,

black,"text",

\_text22\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text22\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text23 = new ShapeText(

SHAPE\_DRAW\_2D, true,810.0, 320.0, 0.0, 0.0,

black,"text",

\_text23\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text23\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

rectangle17 = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,860.0, 190.0, 0.0, 0.0,

black, white,

60.0, 40.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

rectangle18 = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,860.0, 230.0, 0.0, 0.0,

black, white,

60.0, 40.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

rectangle19 = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,860.0, 270.0, 0.0, 0.0,

black, white,

60.0, 40.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

text24 = new ShapeText(

SHAPE\_DRAW\_2D, true,870.0, 200.0, 0.0, 0.0,

black,"text",

\_text24\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text24\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

rectangle20 = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,860.0, 310.0, 0.0, 0.0,

black, white,

60.0, 40.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

text25 = new ShapeText(

SHAPE\_DRAW\_2D, true,870.0, 240.0, 0.0, 0.0,

black,"text",

\_text25\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text25\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text26 = new ShapeText(

SHAPE\_DRAW\_2D, true,870.0, 280.0, 0.0, 0.0,

black,"text",

\_text26\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text26\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text27 = new ShapeText(

SHAPE\_DRAW\_2D, true,870.0, 320.0, 0.0, 0.0,

black,"text",

\_text27\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text27\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

rectangle21 = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,920.0, 190.0, 0.0, 0.0,

black, white,

60.0, 40.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

rectangle22 = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,920.0, 230.0, 0.0, 0.0,

black, white,

60.0, 40.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

rectangle23 = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,920.0, 270.0, 0.0, 0.0,

black, white,

60.0, 40.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

text28 = new ShapeText(

SHAPE\_DRAW\_2D, true,930.0, 200.0, 0.0, 0.0,

black,"text",

\_text28\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text28\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

rectangle24 = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,920.0, 310.0, 0.0, 0.0,

black, white,

60.0, 40.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

text29 = new ShapeText(

SHAPE\_DRAW\_2D, true,930.0, 240.0, 0.0, 0.0,

black,"text",

\_text29\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text29\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text30 = new ShapeText(

SHAPE\_DRAW\_2D, true,930.0, 280.0, 0.0, 0.0,

black,"text",

\_text30\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text30\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text31 = new ShapeText(

SHAPE\_DRAW\_2D, true,930.0, 320.0, 0.0, 0.0,

black,"text",

\_text31\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text31\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text32 = new ShapeText(

SHAPE\_DRAW\_2D, true,820.0, 170.0, 0.0, 0.0,

black,"O",

\_text32\_Font, ALIGNMENT\_LEFT );

text33 = new ShapeText(

SHAPE\_DRAW\_2D, true,880.0, 170.0, 0.0, 0.0,

black,"M",

\_text33\_Font, ALIGNMENT\_LEFT );

text34 = new ShapeText(

SHAPE\_DRAW\_2D, true,940.0, 170.0, 0.0, 0.0,

black,"P",

\_text34\_Font, ALIGNMENT\_LEFT );

text35 = new ShapeText(

SHAPE\_DRAW\_2D, true,780.0, 320.0, 0.0, 0.0,

black,"text",

\_text35\_Font, ALIGNMENT\_RIGHT ) {

@Override

public void updateDynamicProperties() {

\_text35\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text36 = new ShapeText(

SHAPE\_DRAW\_2D, true,780.0, 280.0, 0.0, 0.0,

black,"text",

\_text36\_Font, ALIGNMENT\_RIGHT ) {

@Override

public void updateDynamicProperties() {

\_text36\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text37 = new ShapeText(

SHAPE\_DRAW\_2D, true,780.0, 240.0, 0.0, 0.0,

black,"text",

\_text37\_Font, ALIGNMENT\_RIGHT ) {

@Override

public void updateDynamicProperties() {

\_text37\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text38 = new ShapeText(

SHAPE\_DRAW\_2D, true,780.0, 200.0, 0.0, 0.0,

black,"text",

\_text38\_Font, ALIGNMENT\_RIGHT ) {

@Override

public void updateDynamicProperties() {

\_text38\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text39 = new ShapeText(

SHAPE\_DRAW\_2D, true,460.0, 150.0, 0.0, 0.0,

black,"Reliability Matrix",

\_text39\_Font, ALIGNMENT\_LEFT );

text40 = new ShapeText(

SHAPE\_DRAW\_2D, true,850.0, 150.0, 0.0, 0.0,

black,"Regret Matrix",

\_text40\_Font, ALIGNMENT\_LEFT );

}

@AnyLogicInternalCodegenAPI

private void \_createPersistentElementsAP0\_xjal() {

}

@AnyLogicInternalCodegenAPI

private void \_createPersistentElementsBS0\_xjal() {

}

// Static initialization of persistent elements

{

level = new com.anylogic.engine.markup.Level(this, "level", SHAPE\_DRAW\_2D3D, 0.0, true, true);

\_getLevels\_xjal = new com.anylogic.engine.markup.Level[] {

level };

\_createPersistentElementsBP0\_xjal();

}

protected ShapeTopLevelPresentationGroup presentation;

protected ShapeModelElementsGroup icon;

@Override

@AnyLogicInternalCodegenAPI

public ShapeTopLevelPresentationGroup getPresentationShape() {

return presentation;

}

@Override

@AnyLogicInternalCodegenAPI

public ShapeModelElementsGroup getModelElementsShape() {

return icon;

}

/\*\*

\* Constructor

\*/

public Agency( Engine engine, Agent owner, AgentList<? extends Agency> ownerPopulation ) {

super( engine, owner, ownerPopulation );

instantiateBaseStructureThis\_xjal();

}

@AnyLogicInternalCodegenAPI

public void onOwnerChanged\_xjal() {

super.onOwnerChanged\_xjal();

setupReferences\_xjal();

}

@AnyLogicInternalCodegenAPI

public void instantiateBaseStructure\_xjal() {

super.instantiateBaseStructure\_xjal();

instantiateBaseStructureThis\_xjal();

}

@AnyLogicInternalCodegenAPI

private void instantiateBaseStructureThis\_xjal() {

setupReferences\_xjal();

}

@AnyLogicInternalCodegenAPI

private void setupReferences\_xjal() {

main = get\_Main();

}

/\*\*

\* Simple constructor. Please add created agent to some population by calling goToPopulation() function

\*/

public Agency() {

}

/\*\*

\* Simple constructor. Please add created agent to some population by calling goToPopulation() function

\*/

public Agency( double CapitalPlanFund, int RiskAttitude, int DecisionHorizon, double RiskThreshold, int RiskUpdateLimit ) {

markParametersAreSet();

this.CapitalPlanFund = CapitalPlanFund;

this.RiskAttitude = RiskAttitude;

this.DecisionHorizon = DecisionHorizon;

this.RiskThreshold = RiskThreshold;

this.RiskUpdateLimit = RiskUpdateLimit;

}

@Override

@AnyLogicInternalCodegenAPI

public void doCreate() {

super.doCreate();

// Assigning initial values for plain variables

setupPlainVariables\_Agency\_xjal();

// Dynamic initialization of persistent elements

\_createPersistentElementsAP0\_xjal();

\_initialize\_level\_xjal();

level.initialize();

presentation = new ShapeTopLevelPresentationGroup( Agency.this, true, 0, 0, 0, 0 , level );

icon = new ShapeModelElementsGroup( Agency.this, getElementProperty( "wateradaptationmodelv8.Agency.icon", IElementDescriptor.MODEL\_ELEMENT\_DESCRIPTORS ) );

icon.setIconOffsets( 0.0, 0.0 );

// Port connectors with non-replicated objects

// Creating replicated embedded objects

setupInitialConditions\_xjal( Agency.class );

// Dynamic initialization of persistent elements

\_createPersistentElementsBS0\_xjal();

}

@Override

@AnyLogicInternalCodegenAPI

public void doStart() {

super.doStart();

CheckWellStatus.start();

IncrementBudget.start();

YearlyPerceivedSSIntrusion.start();

RecoveryAdaptEvent.start();

RiskAttitudeStateChart.start();

}

@AnyLogicInternalCodegenAPI

public void onStartup() {

super.onStartup();

updateAdapCostDB();

}

/\*\*

\* Assigning initial values for plain variables<br>

\* <em>This method isn't designed to be called by user and may be removed in future releases.</em>

\*/

@AnyLogicInternalCodegenAPI

public void setupPlainVariables\_xjal() {

setupPlainVariables\_Agency\_xjal();

}

/\*\*

\* Assigning initial values for plain variables<br>

\* <em>This method isn't designed to be called by user and may be removed in future releases.</em>

\*/

@AnyLogicInternalCodegenAPI

private void setupPlainVariables\_Agency\_xjal() {

Budget =

0

;

ReliabilityMatrix =

new double[TotalAdaptations()][3]

;

RegretMatrix =

new double[TotalAdaptations()][3]

;

PerceivedSSIntrusionByDH =

0

;

SSIntrusionDH =

new double[DecisionHorizon]

;

RiskUpdateCount =

0

;

UnitCost =

(double) selectUniqueValue(double.class,

"SELECT cost\_dollars FROM adaptation WHERE " +

"id = ? LIMIT 2;",

4

)

;

FixedCost =

(double) pert(500,700,600)

;

DesalinationApplied =

false

;

}

// User API -----------------------------------------------------

public Main get\_Main() {

{

Agent owner = getOwner();

if ( owner instanceof Main ) return (Main) owner;

}

return null;

}

/\*\*

\* Read-only variable. <em>Shouldn't be modified by user.</em>

\*/

@AnyLogicCustomSerialization(AnyLogicCustomSerializationMode.REFERENCE)

public transient wateradaptationmodelv8.Main main;

@AnyLogicInternalCodegenAPI

public static LinkToAgentAnimationSettings \_connections\_commonAnimationSettings\_xjal = new LinkToAgentAnimationSettingsImpl( false, black, 1.0, LINE\_STYLE\_SOLID, ARROW\_NONE, 0.0 );

public LinkToAgentCollection<Agent, Agent> connections = new LinkToAgentStandardImpl<Agent, Agent>(this, \_connections\_commonAnimationSettings\_xjal);

@Override

public LinkToAgentCollection<? extends Agent, ? extends Agent> getLinkToAgentStandard\_xjal() {

return connections;

}

@Override

@AnyLogicInternalCodegenAPI

public void onReceive( Object \_msg\_xjal, Agent \_sender\_xjal ) {

super.onReceive( \_msg\_xjal, \_sender\_xjal );

RiskAttitudeStateChart.fireEvent( \_msg\_xjal );

}

@AnyLogicInternalCodegenAPI

public void drawLinksToAgents(boolean \_underAgents\_xjal, LinkToAgentAnimator \_animator\_xjal) {

super.drawLinksToAgents(\_underAgents\_xjal, \_animator\_xjal);

if ( \_underAgents\_xjal ) {

\_animator\_xjal.drawLink( this, connections, true, true );

}

}

public AgentList<? extends Agency> getPopulation() {

return (AgentList<? extends Agency>) super.getPopulation();

}

public List<? extends Agency> agentsInRange( double distance ) {

return (List<? extends Agency>) super.agentsInRange( distance );

}

@Override

@AnyLogicInternalCodegenAPI

public boolean isLoggingToDB(EventOriginator \_e) {

if ( \_e == IncrementBudget ) return false;

return super.isLoggingToDB( \_e );

}

// Reaction on changes -------------------------------------

public void onChange() {

super.onChange();

RiskAttitudeStateChart.onChange();

}

@AnyLogicInternalCodegenAPI

public void onDestroy() {

CheckWellStatus.onDestroy();

IncrementBudget.onDestroy();

YearlyPerceivedSSIntrusion.onDestroy();

RecoveryAdaptEvent.onDestroy();

RiskAttitudeStateChart.onDestroy();

super.onDestroy();

}

}

**Hazards (Stressor) Agent:**

public class Stressor extends Agent

{

// Parameters

public

double SLRScenario;

/\*\*

\* Returns default value for parameter <code>SLRScenario</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public double \_SLRScenario\_DefaultValue\_xjal() {

final Stressor self = this;

return

main.ChosenSLRScenario

;

}

public void set\_SLRScenario( double SLRScenario ) {

if (SLRScenario == this.SLRScenario) {

return;

}

double \_oldValue\_xjal = this.SLRScenario;

this.SLRScenario = SLRScenario;

onChange\_SLRScenario\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter SLRScenario.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_SLRScenario()</code> method instead.

\*/

protected void onChange\_SLRScenario() {

onChange\_SLRScenario\_xjal( SLRScenario );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_SLRScenario\_xjal( double oldValue ) {

}

public

double CaseScenario;

/\*\*

\* Returns default value for parameter <code>CaseScenario</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public double \_CaseScenario\_DefaultValue\_xjal() {

final Stressor self = this;

return 0.0;

}

public void set\_CaseScenario( double CaseScenario ) {

if (CaseScenario == this.CaseScenario) {

return;

}

double \_oldValue\_xjal = this.CaseScenario;

this.CaseScenario = CaseScenario;

onChange\_CaseScenario\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter CaseScenario.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_CaseScenario()</code> method instead.

\*/

protected void onChange\_CaseScenario() {

onChange\_CaseScenario\_xjal( CaseScenario );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_CaseScenario\_xjal( double oldValue ) {

}

@Override

public void setParametersToDefaultValues() {

super.setParametersToDefaultValues();

SLRScenario = \_SLRScenario\_DefaultValue\_xjal();

CaseScenario = \_CaseScenario\_DefaultValue\_xjal();

}

@Override

public boolean setParameter(String \_name\_xjal, Object \_value\_xjal, boolean \_callOnChange\_xjal) {

switch ( \_name\_xjal ) {

case "SLRScenario":

if ( \_callOnChange\_xjal ) {

set\_SLRScenario( ((Number) \_value\_xjal).doubleValue() );

} else {

SLRScenario = ((Number) \_value\_xjal).doubleValue();

}

return true;

case "CaseScenario":

if ( \_callOnChange\_xjal ) {

set\_CaseScenario( ((Number) \_value\_xjal).doubleValue() );

} else {

CaseScenario = ((Number) \_value\_xjal).doubleValue();

}

return true;

default:

return super.setParameter( \_name\_xjal, \_value\_xjal, \_callOnChange\_xjal );

}

}

@Override

public <T> T getParameter(String \_name\_xjal) {

Object \_result\_xjal;

switch ( \_name\_xjal ) {

case "SLRScenario": \_result\_xjal = SLRScenario; break;

case "CaseScenario": \_result\_xjal = CaseScenario; break;

default: \_result\_xjal = super.getParameter( \_name\_xjal ); break;

}

return (T) \_result\_xjal;

}

@AnyLogicInternalCodegenAPI

private static String[] \_parameterNames\_xjal;

@Override

public String[] getParameterNames() {

String[] result = \_parameterNames\_xjal;

if (result == null) {

List<String> list = new ArrayList<>( Arrays.asList( super.getParameterNames() ) );

list.add( "SLRScenario" );

list.add( "CaseScenario" );

result = list.toArray( new String[ list.size() ] );

\_parameterNames\_xjal = result;

}

return result;

}

// Plain Variables

public

double

StormSurgeCounter;

public

double

ActualSSLambda;

// Dynamic (Flow/Auxiliary/Stock) Variables

public double year\_base\_10;

@AnyLogicInternalCodegenAPI

public void assignInitialConditions\_xjal() {

super.assignInitialConditions\_xjal();

\_assign\_year\_base\_10\_Formula\_xjal();

}

@AnyLogicInternalCodegenAPI

public void setupInitialConditions\_xjal(Class<?> callerClass) {

if (callerClass != Stressor.class) {

return;

}

if (getInitialAlgebraicFlatEquationsCount\_xjal() > 0) {

SDIntegrationManager integrationManagerForInitialConditions = new SDIntegrationManager( 0, getInitialAlgebraicFlatEquationsCount\_xjal(), getInitialFormulaFlatEquationsCount\_xjal() );

integrationManagerForInitialConditions.doStep( this, 0, 0.1, true );

} else {

assignInitialConditions\_xjal();

}

}

@AnyLogicInternalCodegenAPI

public void \_assign\_year\_base\_10\_Formula\_xjal() {

year\_base\_10 =

getYearBase10((int)main.current\_model\_year)

;

}

@AnyLogicInternalCodegenAPI

public void formulasExecute\_xjal() {

super.formulasExecute\_xjal();

\_assign\_year\_base\_10\_Formula\_xjal();

}

@AnyLogicInternalCodegenAPI

protected SDIntegrationManager integrationManager\_xjal = null;

@AnyLogicInternalCodegenAPI

public SDIntegrationManager getIntegrationManager\_xjal() {

if (integrationManager\_xjal == null) {

integrationManager\_xjal = new SDIntegrationManager( getDifferentialFlatEquationsCount\_xjal(), getRuntimeAlgebraicFlatEquationsCount\_xjal(), getRuntimeFormulaFlatEquationsCount\_xjal() );

}

return integrationManager\_xjal;

}

@Override

@AnyLogicInternalCodegenAPI

public int getRuntimeFormulaFlatEquationsCount\_xjal() {

return super.getRuntimeFormulaFlatEquationsCount\_xjal() + 1;

}

@Override

@AnyLogicInternalCodegenAPI

public int getInitialFormulaFlatEquationsCount\_xjal() {

return super.getInitialFormulaFlatEquationsCount\_xjal() + 1;

}

@AnyLogicInternalCodegenAPI

private static Map<String, IElementDescriptor> elementDesciptors\_xjal = createElementDescriptors( Stressor.class );

@AnyLogicInternalCodegenAPI

@Override

public Map<String, IElementDescriptor> getElementDesciptors() {

return elementDesciptors\_xjal;

}

@AnyLogicCustomProposalPriority(type = AnyLogicCustomProposalPriority.Type.STATIC\_ELEMENT)

public static final Scale scale = new Scale( 10.0 );

@Override

public Scale getScale() {

return scale;

}

// Events

public EventRate YearlyStormSurgeEvent = new EventRate(this);

@AnyLogicInternalCodegenAPI

public EventTimeout \_autoCreatedDS\_xjal = new EventTimeout(this);

@Override

@AnyLogicInternalCodegenAPI

public String getNameOf( EventTimeout \_e ) {

if( \_e == \_autoCreatedDS\_xjal ) return "Auto-created DataSets auto update event";

return super.getNameOf( \_e );

}

@Override

@AnyLogicInternalCodegenAPI

public EventTimeout.Mode getModeOf( EventTimeout \_e ) {

if ( \_e == \_autoCreatedDS\_xjal ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

return super.getModeOf( \_e );

}

@Override

@AnyLogicInternalCodegenAPI

public double getFirstOccurrenceTime( EventTimeout \_e ) {

double \_t;

if ( \_e == \_autoCreatedDS\_xjal ) {

\_t =

0

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

return super.getFirstOccurrenceTime( \_e );

}

@Override

@AnyLogicInternalCodegenAPI

public double evaluateTimeoutOf( EventTimeout \_e ) {

double \_t;

if( \_e == \_autoCreatedDS\_xjal) {

\_t =

1

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

return super.evaluateTimeoutOf( \_e );

}

@Override

@AnyLogicInternalCodegenAPI

public void executeActionOf( EventTimeout \_e ) {

if ( \_e == \_autoCreatedDS\_xjal ) {

\_ds\_year\_base\_10.update();

return;

}

super.executeActionOf( \_e );

}

@Override

@AnyLogicInternalCodegenAPI

public String getNameOf( EventRate \_e ) {

if ( \_e == YearlyStormSurgeEvent) return "YearlyStormSurgeEvent";

return super.getNameOf( \_e );

}

@Override

@AnyLogicInternalCodegenAPI

public double evaluateRateOf( EventRate \_e ) {

double \_value;

if ( \_e == YearlyStormSurgeEvent) {

\_value =

5

;

\_value = toModelRate( \_value, PER\_YEAR );

return \_value;

}

return super.evaluateRateOf( \_e );

}

@Override

@AnyLogicInternalCodegenAPI

public void executeActionOf( EventRate self ) {

if ( self == YearlyStormSurgeEvent) {

Random rand = new Random();

int poissonval = poisson(ActualSSLambda, rand);

if(poissonval==1)

{

StormSurgeStateChart.fireEvent(true);

}

;

return ;

}

super.executeActionOf( self );

}

/\*\* Internal constant, shouldn't be accessed by user \*/

@AnyLogicInternalCodegenAPI

protected static final int \_STATECHART\_COUNT\_xjal = 1;

// Statecharts

public Statechart<StormSurgeStateChart\_state> StormSurgeStateChart = new Statechart<>( this, (short)1 );

@Override

@AnyLogicInternalCodegenAPI

public String getNameOf( Statechart \_s ) {

if(\_s == this.StormSurgeStateChart) return "StormSurgeStateChart";

return super.getNameOf( \_s );

}

@Override

@AnyLogicInternalCodegenAPI

public int getIdOf( Statechart \_s ) {

if(\_s == this.StormSurgeStateChart) return 0;

return super.getIdOf( \_s );

}

@Override

@AnyLogicInternalCodegenAPI

public void executeActionOf( Statechart \_s ) {

if( \_s == this.StormSurgeStateChart ) {

enterState( NoStormSurge, true );

return;

}

super.executeActionOf( \_s );

}

// States of all statecharts

public enum StormSurgeStateChart\_state implements IStatechartState<Stressor, StormSurgeStateChart\_state> {

NoStormSurge,

StormSurge;

@AnyLogicInternalCodegenAPI

private Collection<StormSurgeStateChart\_state> \_simpleStatesDeep\_xjal;

@AnyLogicInternalCodegenAPI

private Set<StormSurgeStateChart\_state> \_fullState\_xjal;

@AnyLogicInternalCodegenAPI

private Set<StormSurgeStateChart\_state> \_statesInside\_xjal;

@Override

@AnyLogicInternalCodegenAPI

public Collection<StormSurgeStateChart\_state> getSimpleStatesDeep() {

Collection<StormSurgeStateChart\_state> result = \_simpleStatesDeep\_xjal;

if (result == null) {

\_simpleStatesDeep\_xjal = result = calculateAllSimpleStatesDeep();

}

return result;

}

@Override

public Set<StormSurgeStateChart\_state> getFullState() {

Set<StormSurgeStateChart\_state> result = \_fullState\_xjal;

if (result == null) {

\_fullState\_xjal = result = calculateFullState();

}

return result;

}

@Override

@AnyLogicInternalCodegenAPI

public Set<StormSurgeStateChart\_state> getStatesInside() {

Set<StormSurgeStateChart\_state> result = \_statesInside\_xjal;

if (result == null) {

\_statesInside\_xjal = result = calculateStatesInside();

}

return result;

}

@Override

@AnyLogicInternalCodegenAPI

public Statechart<StormSurgeStateChart\_state> getStatechart( Stressor \_a ) {

return \_a.StormSurgeStateChart;

}

}

@AnyLogicCustomProposalPriority(type = AnyLogicCustomProposalPriority.Type.STATIC\_ELEMENT)

public static final StormSurgeStateChart\_state NoStormSurge = StormSurgeStateChart\_state.NoStormSurge;

@AnyLogicCustomProposalPriority(type = AnyLogicCustomProposalPriority.Type.STATIC\_ELEMENT)

public static final StormSurgeStateChart\_state StormSurge = StormSurgeStateChart\_state.StormSurge;

@AnyLogicInternalCodegenAPI

private void enterState( StormSurgeStateChart\_state self, boolean \_destination ) {

switch( self ) {

case NoStormSurge:

logToDBEnterState(StormSurgeStateChart, self);

// (Simple state (not composite))

StormSurgeStateChart.setActiveState\_xjal( NoStormSurge );

transition.start();

return;

case StormSurge:

logToDBEnterState(StormSurgeStateChart, self);

// (Simple state (not composite))

StormSurgeStateChart.setActiveState\_xjal( StormSurge );

{

StormSurgeCounter++;

int yb\_10 = (int)year\_base\_10;

if(yb\_10 > 90){

yb\_10 = 90;

}

if(main.StormSurgeBCData.containsKey(yb\_10)){

int ss\_val = main.StormSurgeBCData.get(yb\_10);

main.StormSurgeBCData.replace(yb\_10, ss\_val+1);

} else{

main.StormSurgeBCData.put(yb\_10, 1);

}

;}

transition1.start();

return;

default:

return;

}

}

@AnyLogicInternalCodegenAPI

private void exitState( StormSurgeStateChart\_state self, Transition \_t, boolean \_source ) {

switch( self ) {

case NoStormSurge:

logToDBExitState(StormSurgeStateChart, self);

logToDB(StormSurgeStateChart, \_t, self);

// (Simple state (not composite))

if ( !\_source || \_t != transition) transition.cancel();

return;

case StormSurge:

logToDBExitState(StormSurgeStateChart, self);

logToDB(StormSurgeStateChart, \_t, self);

// (Simple state (not composite))

if ( !\_source || \_t != transition1) transition1.cancel();

return;

default:

return;

}

}

@AnyLogicInternalCodegenAPI

private void exitInnerStates( StormSurgeStateChart\_state \_destination ) {

StormSurgeStateChart\_state \_state = StormSurgeStateChart.getActiveSimpleState();

while( \_state != \_destination ) {

exitState( \_state, null, false );

\_state = \_state.getContainerState();

}

}

public TransitionTimeout transition1 = new TransitionTimeout( this );

@Override

@AnyLogicInternalCodegenAPI

public String getNameOf( TransitionTimeout \_t ) {

if ( \_t == transition1 ) return "transition1";

return super.getNameOf( \_t );

}

@Override

@AnyLogicInternalCodegenAPI

public Statechart getStatechartOf( TransitionTimeout \_t ) {

if ( \_t == transition1 ) return StormSurgeStateChart;

return super.getStatechartOf( \_t );

}

@Override

@AnyLogicInternalCodegenAPI

public void executeActionOf( TransitionTimeout self ) {

if ( self == transition1 ) {

exitState( StormSurge, self, true );

enterState( NoStormSurge, true );

return;

}

super.executeActionOf( self );

}

@Override

@AnyLogicInternalCodegenAPI

public double evaluateTimeoutOf( TransitionTimeout \_t ) {

double \_value;

if ( \_t == transition1 ) {

\_value =

1

;

\_value = toModelTime( \_value, YEAR );

return \_value;

}

return super.evaluateTimeoutOf( \_t );

}

public TransitionMessage transition = new TransitionMessage( this );

@Override

@AnyLogicInternalCodegenAPI

public String getNameOf( TransitionMessage \_t ) {

if ( \_t == transition ) return "transition";

return super.getNameOf( \_t );

}

@Override

@AnyLogicInternalCodegenAPI

public Statechart getStatechartOf( TransitionMessage \_t ) {

if ( \_t == transition ) return StormSurgeStateChart;

return super.getStatechartOf( \_t );

}

@Override

@AnyLogicInternalCodegenAPI

public void executeActionOf( TransitionMessage self, Object \_msg ) {

if ( self == transition ) {

exitState( NoStormSurge, self, true );

enterState( StormSurge, true );

return;

}

super.executeActionOf( self, \_msg );

}

@Override

@AnyLogicInternalCodegenAPI

public boolean testMessageOf( TransitionMessage \_t, Object \_msg ) {

if ( \_t == transition ) {

Object

msg = (Object) \_msg;

return true;

}

return super.testMessageOf( \_t, \_msg );

}

// Functions

/\*\*

\* Get the current saltwater intrusion line according to year, chosen case scenario and sea level rise scenario.

\*/

int

getCurrentSWIL( ) {

//System.out.println(SLRScenario);

//System.out.println(CaseScenario);

//System.out.println(year\_base\_10);

int impact = (int) selectFrom(slr\_impact)

.where(slr\_impact.slr\_id.eq((int) SLRScenario))

.where(slr\_impact.case\_id.eq((int) CaseScenario))

.where(slr\_impact.year.eq((int) year\_base\_10))

.uniqueResult(slr\_impact.line\_value);

return (StormSurgeStateChart.getActiveSimpleState() == StormSurge) ? impact + (int) ActualSSIntrusion() : impact;

}

double

getStormSurgeEvent( ) {

if(StormSurgeStateChart.getActiveSimpleState() == StormSurge)

return 1.0;

else

return 0.0;

}

int

getPerceivedSWIL( int slr\_scenario, int case\_scenario, int year ) {

//System.out.println("getPerceivedSWIL");

//System.out.println(slr\_scenario);

//System.out.println(case\_scenario);

//System.out.println(getYearBase10(year));

int impact = (int) selectFrom(slr\_impact)

.where(slr\_impact.slr\_id.eq((int) slr\_scenario))

.where(slr\_impact.case\_id.eq((int) case\_scenario))

.where(slr\_impact.year.eq((int) getYearBase10(year)))

.uniqueResult(slr\_impact.line\_value);

return impact;

//return impact + (int) perceived\_ssintrusion;

}

int

getYearBase10( int year ) {

if(year > 9 && year < 110){

return (int) ((Math.ceil(year/10) \* 10));

} else if(year > 109){

return 100;

} else{

return 0;

}

}

double

ActualSSIntrusion( ) {

if(SLRScenario == main.Agency.SLR\_SLOW){

return uniform(100, 200);

}

else if(SLRScenario == main.Agency.SLR\_MEDIUM){

return uniform(200, 300);

}

else {

return uniform(300, 400);

}

}

/\*\*

\* Auto-created data set(s) for year\_base\_10

\*/

@AnyLogicInternalCodegenAPI

public DataSet \_ds\_year\_base\_10 = new DataSet( 100, new DataUpdater\_xjal() {

double \_lastUpdateTime = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateTime ) { return; }

\_d.add( time(), Stressor.this.year\_base\_10 );

\_lastUpdateTime = time();

}

} );

// View areas

public ViewArea \_origin\_VA = new ViewArea( this, "[Origin]", 0, 0, 1020.0, 640.0 );

@Override

@AnyLogicInternalCodegenAPI

public int getViewAreas(Map<String, ViewArea> \_output) {

if ( \_output != null ) {

\_output.put( "\_origin\_VA", this.\_origin\_VA );

}

return 1 + super.getViewAreas( \_output );

}

/\*\* Internal constant, shouldn't be accessed by user \*/

@AnyLogicInternalCodegenAPI

protected static final int \_SHAPE\_NEXT\_ID\_xjal = 1;

@AnyLogicInternalCodegenAPI

public boolean isPublicPresentationDefined() {

return false;

}

@AnyLogicInternalCodegenAPI

public boolean isEmbeddedAgentPresentationVisible( Agent \_a ) {

return super.isEmbeddedAgentPresentationVisible( \_a );

}

@AnyLogicInternalCodegenAPI

private void \_initialize\_level\_xjal() {

level.addAll();

}

protected com.anylogic.engine.markup.Level level;

private com.anylogic.engine.markup.Level[] \_getLevels\_xjal;

@Override

public com.anylogic.engine.markup.Level[] getLevels() {

return \_getLevels\_xjal;

}

@AnyLogicInternalCodegenAPI

private void \_createPersistentElementsBP0\_xjal() {

}

@AnyLogicInternalCodegenAPI

private void \_createPersistentElementsAP0\_xjal() {

}

@AnyLogicInternalCodegenAPI

private void \_createPersistentElementsBS0\_xjal() {

}

// Static initialization of persistent elements

{

level = new com.anylogic.engine.markup.Level(this, "level", SHAPE\_DRAW\_2D3D, 0.0, true, true);

\_getLevels\_xjal = new com.anylogic.engine.markup.Level[] {

level };

\_createPersistentElementsBP0\_xjal();

}

protected ShapeTopLevelPresentationGroup presentation;

protected ShapeModelElementsGroup icon;

@Override

@AnyLogicInternalCodegenAPI

public ShapeTopLevelPresentationGroup getPresentationShape() {

return presentation;

}

@Override

@AnyLogicInternalCodegenAPI

public ShapeModelElementsGroup getModelElementsShape() {

return icon;

}

/\*\*

\* Constructor

\*/

public Stressor( Engine engine, Agent owner, AgentList<? extends Stressor> ownerPopulation ) {

super( engine, owner, ownerPopulation );

instantiateBaseStructureThis\_xjal();

}

@AnyLogicInternalCodegenAPI

public void onOwnerChanged\_xjal() {

super.onOwnerChanged\_xjal();

setupReferences\_xjal();

}

@AnyLogicInternalCodegenAPI

public void instantiateBaseStructure\_xjal() {

super.instantiateBaseStructure\_xjal();

instantiateBaseStructureThis\_xjal();

}

@AnyLogicInternalCodegenAPI

private void instantiateBaseStructureThis\_xjal() {

setupReferences\_xjal();

// Registering in Engine continuous part

getEngine().registerAgentWithEquations( this );

}

@AnyLogicInternalCodegenAPI

private void setupReferences\_xjal() {

main = get\_Main();

}

/\*\*

\* Simple constructor. Please add created agent to some population by calling goToPopulation() function

\*/

public Stressor() {

}

/\*\*

\* Simple constructor. Please add created agent to some population by calling goToPopulation() function

\*/

public Stressor( double SLRScenario, double CaseScenario ) {

markParametersAreSet();

this.SLRScenario = SLRScenario;

this.CaseScenario = CaseScenario;

}

@Override

@AnyLogicInternalCodegenAPI

public void doCreate() {

super.doCreate();

// Assigning initial values for plain variables

setupPlainVariables\_Stressor\_xjal();

// Dynamic initialization of persistent elements

\_createPersistentElementsAP0\_xjal();

\_initialize\_level\_xjal();

level.initialize();

presentation = new ShapeTopLevelPresentationGroup( Stressor.this, true, 0, 0, 0, 0 , level );

icon = new ShapeModelElementsGroup( Stressor.this, getElementProperty( "wateradaptationmodelv8.Stressor.icon", IElementDescriptor.MODEL\_ELEMENT\_DESCRIPTORS ) );

icon.setIconOffsets( 0.0, 0.0 );

// Port connectors with non-replicated objects

// Creating replicated embedded objects

setupInitialConditions\_xjal( Stressor.class );

// Dynamic initialization of persistent elements

\_createPersistentElementsBS0\_xjal();

}

@Override

@AnyLogicInternalCodegenAPI

public void doStart() {

super.doStart();

\_autoCreatedDS\_xjal.start();

YearlyStormSurgeEvent.start();

StormSurgeStateChart.start();

}

/\*\*

\* Assigning initial values for plain variables<br>

\* <em>This method isn't designed to be called by user and may be removed in future releases.</em>

\*/

@AnyLogicInternalCodegenAPI

public void setupPlainVariables\_xjal() {

setupPlainVariables\_Stressor\_xjal();

}

/\*\*

\* Assigning initial values for plain variables<br>

\* <em>This method isn't designed to be called by user and may be removed in future releases.</em>

\*/

@AnyLogicInternalCodegenAPI

private void setupPlainVariables\_Stressor\_xjal() {

StormSurgeCounter =

0

;

ActualSSLambda =

(double) selectUniqueValue(double.class,

"SELECT lambda FROM ss\_impact WHERE " +

"slr\_scenario = ? LIMIT 2;",

SLRScenario

)

;

}

// User API -----------------------------------------------------

public Main get\_Main() {

{

Agent owner = getOwner();

if ( owner instanceof Main ) return (Main) owner;

}

return null;

}

/\*\*

\* Read-only variable. <em>Shouldn't be modified by user.</em>

\*/

@AnyLogicCustomSerialization(AnyLogicCustomSerializationMode.REFERENCE)

public transient wateradaptationmodelv8.Main main;

@AnyLogicInternalCodegenAPI

public static LinkToAgentAnimationSettings \_connections\_commonAnimationSettings\_xjal = new LinkToAgentAnimationSettingsImpl( false, black, 1.0, LINE\_STYLE\_SOLID, ARROW\_NONE, 0.0 );

public LinkToAgentCollection<Agent, Agent> connections = new LinkToAgentStandardImpl<Agent, Agent>(this, \_connections\_commonAnimationSettings\_xjal);

@Override

public LinkToAgentCollection<? extends Agent, ? extends Agent> getLinkToAgentStandard\_xjal() {

return connections;

}

@Override

@AnyLogicInternalCodegenAPI

public void onReceive( Object \_msg\_xjal, Agent \_sender\_xjal ) {

super.onReceive( \_msg\_xjal, \_sender\_xjal );

StormSurgeStateChart.fireEvent( \_msg\_xjal );

}

@AnyLogicInternalCodegenAPI

public void drawLinksToAgents(boolean \_underAgents\_xjal, LinkToAgentAnimator \_animator\_xjal) {

super.drawLinksToAgents(\_underAgents\_xjal, \_animator\_xjal);

if ( \_underAgents\_xjal ) {

\_animator\_xjal.drawLink( this, connections, true, true );

}

}

public AgentList<? extends Stressor> getPopulation() {

return (AgentList<? extends Stressor>) super.getPopulation();

}

public List<? extends Stressor> agentsInRange( double distance ) {

return (List<? extends Stressor>) super.agentsInRange( distance );

}

@Override

@AnyLogicInternalCodegenAPI

public boolean isLoggingToDB(EventOriginator \_e) {

if ( \_e == \_autoCreatedDS\_xjal ) return false;

return super.isLoggingToDB( \_e );

}

// Reaction on changes -------------------------------------

public void onChange() {

super.onChange();

YearlyStormSurgeEvent.onChange();

StormSurgeStateChart.onChange();

}

@AnyLogicInternalCodegenAPI

public void onDestroy() {

\_autoCreatedDS\_xjal.onDestroy();

YearlyStormSurgeEvent.onDestroy();

StormSurgeStateChart.onDestroy();

// Unregistering in Engine continuous part

getEngine().unregisterAgentWithEquations( this );

\_ds\_year\_base\_10.destroyUpdater\_xjal();

super.onDestroy();

}

}

**User Agent:**

public class User extends Agent

{

// Parameters

public

double ConsumptionPerCapita;

/\*\*

\* Returns default value for parameter <code>ConsumptionPerCapita</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public double \_ConsumptionPerCapita\_DefaultValue\_xjal() {

final User self = this;

return 0.0;

}

public void set\_ConsumptionPerCapita( double ConsumptionPerCapita ) {

if (ConsumptionPerCapita == this.ConsumptionPerCapita) {

return;

}

double \_oldValue\_xjal = this.ConsumptionPerCapita;

this.ConsumptionPerCapita = ConsumptionPerCapita;

onChange\_ConsumptionPerCapita\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter ConsumptionPerCapita.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_ConsumptionPerCapita()</code> method instead.

\*/

protected void onChange\_ConsumptionPerCapita() {

onChange\_ConsumptionPerCapita\_xjal( ConsumptionPerCapita );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_ConsumptionPerCapita\_xjal( double oldValue ) {

}

public

String Name;

/\*\*

\* Returns default value for parameter <code>Name</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public String \_Name\_DefaultValue\_xjal() {

final User self = this;

return null;

}

public void set\_Name( String Name ) {

if (Name == this.Name) {

return;

}

String \_oldValue\_xjal = this.Name;

this.Name = Name;

onChange\_Name\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter Name.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_Name()</code> method instead.

\*/

protected void onChange\_Name() {

onChange\_Name\_xjal( Name );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_Name\_xjal( String oldValue ) {

}

public

int Population;

/\*\*

\* Returns default value for parameter <code>Population</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public int \_Population\_DefaultValue\_xjal() {

final User self = this;

return 0;

}

public void set\_Population( int Population ) {

if (Population == this.Population) {

return;

}

int \_oldValue\_xjal = this.Population;

this.Population = Population;

onChange\_Population\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter Population.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_Population()</code> method instead.

\*/

protected void onChange\_Population() {

onChange\_Population\_xjal( Population );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_Population\_xjal( int oldValue ) {

}

public

double PopulationGrowth;

/\*\*

\* Returns default value for parameter <code>PopulationGrowth</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public double \_PopulationGrowth\_DefaultValue\_xjal() {

final User self = this;

return 0.0;

}

public void set\_PopulationGrowth( double PopulationGrowth ) {

if (PopulationGrowth == this.PopulationGrowth) {

return;

}

double \_oldValue\_xjal = this.PopulationGrowth;

this.PopulationGrowth = PopulationGrowth;

onChange\_PopulationGrowth\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter PopulationGrowth.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_PopulationGrowth()</code> method instead.

\*/

protected void onChange\_PopulationGrowth() {

onChange\_PopulationGrowth\_xjal( PopulationGrowth );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_PopulationGrowth\_xjal( double oldValue ) {

}

public

double ChosenDecisionHorizon;

/\*\*

\* Returns default value for parameter <code>ChosenDecisionHorizon</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public double \_ChosenDecisionHorizon\_DefaultValue\_xjal() {

final User self = this;

return 0.0;

}

public void set\_ChosenDecisionHorizon( double ChosenDecisionHorizon ) {

if (ChosenDecisionHorizon == this.ChosenDecisionHorizon) {

return;

}

double \_oldValue\_xjal = this.ChosenDecisionHorizon;

this.ChosenDecisionHorizon = ChosenDecisionHorizon;

onChange\_ChosenDecisionHorizon\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter ChosenDecisionHorizon.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_ChosenDecisionHorizon()</code> method instead.

\*/

protected void onChange\_ChosenDecisionHorizon() {

onChange\_ChosenDecisionHorizon\_xjal( ChosenDecisionHorizon );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_ChosenDecisionHorizon\_xjal( double oldValue ) {

}

@Override

public void setParametersToDefaultValues() {

super.setParametersToDefaultValues();

ConsumptionPerCapita = \_ConsumptionPerCapita\_DefaultValue\_xjal();

Name = \_Name\_DefaultValue\_xjal();

Population = \_Population\_DefaultValue\_xjal();

PopulationGrowth = \_PopulationGrowth\_DefaultValue\_xjal();

ChosenDecisionHorizon = \_ChosenDecisionHorizon\_DefaultValue\_xjal();

}

@Override

public boolean setParameter(String \_name\_xjal, Object \_value\_xjal, boolean \_callOnChange\_xjal) {

switch ( \_name\_xjal ) {

case "ConsumptionPerCapita":

if ( \_callOnChange\_xjal ) {

set\_ConsumptionPerCapita( ((Number) \_value\_xjal).doubleValue() );

} else {

ConsumptionPerCapita = ((Number) \_value\_xjal).doubleValue();

}

return true;

case "Name":

if ( \_callOnChange\_xjal ) {

set\_Name( (String) \_value\_xjal );

} else {

Name = (String) \_value\_xjal;

}

return true;

case "Population":

if ( \_callOnChange\_xjal ) {

set\_Population( ((Number) \_value\_xjal).intValue() );

} else {

Population = ((Number) \_value\_xjal).intValue();

}

return true;

case "PopulationGrowth":

if ( \_callOnChange\_xjal ) {

set\_PopulationGrowth( ((Number) \_value\_xjal).doubleValue() );

} else {

PopulationGrowth = ((Number) \_value\_xjal).doubleValue();

}

return true;

case "ChosenDecisionHorizon":

if ( \_callOnChange\_xjal ) {

set\_ChosenDecisionHorizon( ((Number) \_value\_xjal).doubleValue() );

} else {

ChosenDecisionHorizon = ((Number) \_value\_xjal).doubleValue();

}

return true;

default:

return super.setParameter( \_name\_xjal, \_value\_xjal, \_callOnChange\_xjal );

}

}

@Override

public <T> T getParameter(String \_name\_xjal) {

Object \_result\_xjal;

switch ( \_name\_xjal ) {

case "ConsumptionPerCapita": \_result\_xjal = ConsumptionPerCapita; break;

case "Name": \_result\_xjal = Name; break;

case "Population": \_result\_xjal = Population; break;

case "PopulationGrowth": \_result\_xjal = PopulationGrowth; break;

case "ChosenDecisionHorizon": \_result\_xjal = ChosenDecisionHorizon; break;

default: \_result\_xjal = super.getParameter( \_name\_xjal ); break;

}

return (T) \_result\_xjal;

}

@AnyLogicInternalCodegenAPI

private static String[] \_parameterNames\_xjal;

@Override

public String[] getParameterNames() {

String[] result = \_parameterNames\_xjal;

if (result == null) {

List<String> list = new ArrayList<>( Arrays.asList( super.getParameterNames() ) );

list.add( "ConsumptionPerCapita" );

list.add( "Name" );

list.add( "Population" );

list.add( "PopulationGrowth" );

list.add( "ChosenDecisionHorizon" );

result = list.toArray( new String[ list.size() ] );

\_parameterNames\_xjal = result;

}

return result;

}

@AnyLogicInternalCodegenAPI

private static Map<String, IElementDescriptor> elementDesciptors\_xjal = createElementDescriptors( User.class );

@AnyLogicInternalCodegenAPI

@Override

public Map<String, IElementDescriptor> getElementDesciptors() {

return elementDesciptors\_xjal;

}

@AnyLogicCustomProposalPriority(type = AnyLogicCustomProposalPriority.Type.STATIC\_ELEMENT)

public static final Scale scale = new Scale( 10.0 );

@Override

public Scale getScale() {

return scale;

}

// Events

@AnyLogicInternalCodegenAPI

public EventTimeout \_plot\_autoUpdateEvent\_xjal = new EventTimeout(this);

@Override

@AnyLogicInternalCodegenAPI

public String getNameOf( EventTimeout \_e ) {

if( \_e == \_plot\_autoUpdateEvent\_xjal ) return "plot auto update event";

return super.getNameOf( \_e );

}

@Override

@AnyLogicInternalCodegenAPI

public EventTimeout.Mode getModeOf( EventTimeout \_e ) {

if ( \_e == \_plot\_autoUpdateEvent\_xjal ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

return super.getModeOf( \_e );

}

@Override

@AnyLogicInternalCodegenAPI

public double getFirstOccurrenceTime( EventTimeout \_e ) {

double \_t;

if ( \_e == \_plot\_autoUpdateEvent\_xjal ) {

\_t =

0

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

return super.getFirstOccurrenceTime( \_e );

}

@Override

@AnyLogicInternalCodegenAPI

public double evaluateTimeoutOf( EventTimeout \_e ) {

double \_t;

if( \_e == \_plot\_autoUpdateEvent\_xjal) {

\_t =

1

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

return super.evaluateTimeoutOf( \_e );

}

@Override

@AnyLogicInternalCodegenAPI

public void executeActionOf( EventTimeout \_e ) {

if ( \_e == \_plot\_autoUpdateEvent\_xjal ) {

plot.updateData();

return;

}

super.executeActionOf( \_e );

}

/\*\* Internal constant, shouldn't be accessed by user \*/

@AnyLogicInternalCodegenAPI

protected static final int \_STATECHART\_COUNT\_xjal = 0;

// Functions

double

getYearlyConsumption( ) {

return (Population\*ConsumptionPerCapita\*365\*(Math.pow(PopulationGrowth+1, main.current\_model\_year)));

}

double

getPlannedConsumption( ) {

double planned\_consumption = 0; //initialize with 0

//double planned\_consumption=0;

//calculate the demand according to each year

for(int i = 1; i <= ChosenDecisionHorizon; i++){

planned\_consumption += (Population\*ConsumptionPerCapita\*365\*(Math.pow(PopulationGrowth+1, main.current\_model\_year+i)));

}

return planned\_consumption;

}

double

getPlannedConsumptionYear( int i ) {

return (Population\*ConsumptionPerCapita\*365\*(Math.pow(PopulationGrowth+1, main.current\_model\_year+i)));

}

@AnyLogicInternalCodegenAPI

public DataSet \_plot\_expression0\_dataSet\_xjal = new DataSet( 100, new DataUpdater\_xjal() {

double \_lastUpdateX = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateX ) { return; }

\_d.add( time(), \_\_plot\_expression0\_dataSet\_xjal\_YValue() );

\_lastUpdateX = time();

}

} );

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_\_plot\_expression0\_dataSet\_xjal\_YValue() {

return

getYearlyConsumption()

;

}

// View areas

public ViewArea \_origin\_VA = new ViewArea( this, "[Origin]", 0, 0, 1020.0, 640.0 );

@Override

@AnyLogicInternalCodegenAPI

public int getViewAreas(Map<String, ViewArea> \_output) {

if ( \_output != null ) {

\_output.put( "\_origin\_VA", this.\_origin\_VA );

}

return 1 + super.getViewAreas( \_output );

}

@AnyLogicInternalCodegenAPI

protected static final int \_plot = 1;

/\*\* Internal constant, shouldn't be accessed by user \*/

@AnyLogicInternalCodegenAPI

protected static final int \_SHAPE\_NEXT\_ID\_xjal = 2;

@AnyLogicInternalCodegenAPI

public boolean isPublicPresentationDefined() {

return true;

}

@AnyLogicInternalCodegenAPI

public boolean isEmbeddedAgentPresentationVisible( Agent \_a ) {

return super.isEmbeddedAgentPresentationVisible( \_a );

}

@AnyLogicInternalCodegenAPI

private void \_initialize\_level\_xjal() {

level.addAll(plot);

}

protected TimePlot plot;

protected com.anylogic.engine.markup.Level level;

private com.anylogic.engine.markup.Level[] \_getLevels\_xjal;

@Override

public com.anylogic.engine.markup.Level[] getLevels() {

return \_getLevels\_xjal;

}

@AnyLogicInternalCodegenAPI

private void \_createPersistentElementsBP0\_xjal() {

}

@AnyLogicInternalCodegenAPI

private void \_createPersistentElementsAP0\_xjal() {

{

DataSet \_item;

List<DataSet> \_items = new ArrayList<DataSet>( 1 );

\_items.add( \_plot\_expression0\_dataSet\_xjal );

List<String> \_titles = new ArrayList<String>( 1 );

\_titles.add( "Yearly Consumption" );

List<Chart2DPlot.Appearance> \_appearances = new ArrayList<Chart2DPlot.Appearance>( 1 );

\_appearances.add( new Chart2DPlot.Appearance( lightSlateBlue, true, true, Chart.INTERPOLATION\_LINEAR, 2.0, Chart.POINT\_CIRCLE ) );

plot = new TimePlot(

User.this, true, 210.0, 80.0,

340.0, 220.0,

null, null,

50.0, 30.0,

260.0, 130.0, white, black, black,

30.0, Chart.SOUTH,

100

, Chart.WINDOW\_MOVES\_WITH\_TIME, null, Chart.SCALE\_AUTO,

0, 0, Chart.GRID\_DEFAULT, Chart.GRID\_DEFAULT,

darkGray, darkGray, \_items, \_titles, \_appearances );

}

}

@AnyLogicInternalCodegenAPI

private void \_createPersistentElementsBS0\_xjal() {

}

// Static initialization of persistent elements

{

level = new com.anylogic.engine.markup.Level(this, "level", SHAPE\_DRAW\_2D3D, 0.0, true, true);

\_getLevels\_xjal = new com.anylogic.engine.markup.Level[] {

level };

\_createPersistentElementsBP0\_xjal();

}

protected ShapeTopLevelPresentationGroup presentation;

protected ShapeModelElementsGroup icon;

@Override

@AnyLogicInternalCodegenAPI

public ShapeTopLevelPresentationGroup getPresentationShape() {

return presentation;

}

@Override

@AnyLogicInternalCodegenAPI

public ShapeModelElementsGroup getModelElementsShape() {

return icon;

}

/\*\*

\* Constructor

\*/

public User( Engine engine, Agent owner, AgentList<? extends User> ownerPopulation ) {

super( engine, owner, ownerPopulation );

instantiateBaseStructureThis\_xjal();

}

@AnyLogicInternalCodegenAPI

public void onOwnerChanged\_xjal() {

super.onOwnerChanged\_xjal();

setupReferences\_xjal();

}

@AnyLogicInternalCodegenAPI

public void instantiateBaseStructure\_xjal() {

super.instantiateBaseStructure\_xjal();

instantiateBaseStructureThis\_xjal();

}

@AnyLogicInternalCodegenAPI

private void instantiateBaseStructureThis\_xjal() {

setupReferences\_xjal();

}

@AnyLogicInternalCodegenAPI

private void setupReferences\_xjal() {

main = get\_Main();

}

/\*\*

\* Simple constructor. Please add created agent to some population by calling goToPopulation() function

\*/

public User() {

}

/\*\*

\* Simple constructor. Please add created agent to some population by calling goToPopulation() function

\*/

public User( double ConsumptionPerCapita, String Name, int Population, double PopulationGrowth, double ChosenDecisionHorizon ) {

markParametersAreSet();

this.ConsumptionPerCapita = ConsumptionPerCapita;

this.Name = Name;

this.Population = Population;

this.PopulationGrowth = PopulationGrowth;

this.ChosenDecisionHorizon = ChosenDecisionHorizon;

}

@Override

@AnyLogicInternalCodegenAPI

public void doCreate() {

super.doCreate();

// Assigning initial values for plain variables

setupPlainVariables\_User\_xjal();

// Dynamic initialization of persistent elements

\_createPersistentElementsAP0\_xjal();

\_initialize\_level\_xjal();

level.initialize();

presentation = new ShapeTopLevelPresentationGroup( User.this, true, 0, 0, 0, 0 , level );

icon = new ShapeModelElementsGroup( User.this, getElementProperty( "wateradaptationmodelv8.User.icon", IElementDescriptor.MODEL\_ELEMENT\_DESCRIPTORS ) );

icon.setIconOffsets( 0.0, 0.0 );

// Port connectors with non-replicated objects

// Creating replicated embedded objects

setupInitialConditions\_xjal( User.class );

// Dynamic initialization of persistent elements

\_createPersistentElementsBS0\_xjal();

}

@Override

@AnyLogicInternalCodegenAPI

public void doStart() {

super.doStart();

\_plot\_autoUpdateEvent\_xjal.start();

}

/\*\*

\* Assigning initial values for plain variables<br>

\* <em>This method isn't designed to be called by user and may be removed in future releases.</em>

\*/

@AnyLogicInternalCodegenAPI

public void setupPlainVariables\_xjal() {

setupPlainVariables\_User\_xjal();

}

/\*\*

\* Assigning initial values for plain variables<br>

\* <em>This method isn't designed to be called by user and may be removed in future releases.</em>

\*/

@AnyLogicInternalCodegenAPI

private void setupPlainVariables\_User\_xjal() {

}

// User API -----------------------------------------------------

public Main get\_Main() {

{

Agent owner = getOwner();

if ( owner instanceof Main ) return (Main) owner;

}

return null;

}

/\*\*

\* Read-only variable. <em>Shouldn't be modified by user.</em>

\*/

@AnyLogicCustomSerialization(AnyLogicCustomSerializationMode.REFERENCE)

public transient wateradaptationmodelv8.Main main;

@AnyLogicInternalCodegenAPI

public static LinkToAgentAnimationSettings \_connections\_commonAnimationSettings\_xjal = new LinkToAgentAnimationSettingsImpl( false, black, 1.0, LINE\_STYLE\_SOLID, ARROW\_NONE, 0.0 );

public LinkToAgentCollection<Agent, Agent> connections = new LinkToAgentStandardImpl<Agent, Agent>(this, \_connections\_commonAnimationSettings\_xjal);

@Override

public LinkToAgentCollection<? extends Agent, ? extends Agent> getLinkToAgentStandard\_xjal() {

return connections;

}

@AnyLogicInternalCodegenAPI

public void drawLinksToAgents(boolean \_underAgents\_xjal, LinkToAgentAnimator \_animator\_xjal) {

super.drawLinksToAgents(\_underAgents\_xjal, \_animator\_xjal);

if ( \_underAgents\_xjal ) {

\_animator\_xjal.drawLink( this, connections, true, true );

}

}

public AgentList<? extends User> getPopulation() {

return (AgentList<? extends User>) super.getPopulation();

}

public List<? extends User> agentsInRange( double distance ) {

return (List<? extends User>) super.agentsInRange( distance );

}

@Override

@AnyLogicInternalCodegenAPI

public boolean isLoggingToDB(EventOriginator \_e) {

if ( \_e == \_plot\_autoUpdateEvent\_xjal ) return false;

return super.isLoggingToDB( \_e );

}

@AnyLogicInternalCodegenAPI

public void onDestroy() {

\_plot\_autoUpdateEvent\_xjal.onDestroy();

\_plot\_expression0\_dataSet\_xjal.destroyUpdater\_xjal();

logToDB( \_plot\_expression0\_dataSet\_xjal, "plot : Yearly Consumption" );

super.onDestroy();

}

}

**Water Treatment Plant (Infrastructure Agent):**

public class WaterTreatmentPlant extends Agent

{

// Parameters

public

double DesalinationCapacity;

/\*\*

\* Returns default value for parameter <code>DesalinationCapacity</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public double \_DesalinationCapacity\_DefaultValue\_xjal() {

final WaterTreatmentPlant self = this;

return 0.0;

}

public void set\_DesalinationCapacity( double DesalinationCapacity ) {

if (DesalinationCapacity == this.DesalinationCapacity) {

return;

}

double \_oldValue\_xjal = this.DesalinationCapacity;

this.DesalinationCapacity = DesalinationCapacity;

onChange\_DesalinationCapacity\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter DesalinationCapacity.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_DesalinationCapacity()</code> method instead.

\*/

protected void onChange\_DesalinationCapacity() {

onChange\_DesalinationCapacity\_xjal( DesalinationCapacity );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_DesalinationCapacity\_xjal( double oldValue ) {

}

public

String Name;

/\*\*

\* Returns default value for parameter <code>Name</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public String \_Name\_DefaultValue\_xjal() {

final WaterTreatmentPlant self = this;

return null;

}

public void set\_Name( String Name ) {

if (Name == this.Name) {

return;

}

String \_oldValue\_xjal = this.Name;

this.Name = Name;

onChange\_Name\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter Name.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_Name()</code> method instead.

\*/

protected void onChange\_Name() {

onChange\_Name\_xjal( Name );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_Name\_xjal( String oldValue ) {

}

public

double TreatmentCapacity;

/\*\*

\* Returns default value for parameter <code>TreatmentCapacity</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public double \_TreatmentCapacity\_DefaultValue\_xjal() {

final WaterTreatmentPlant self = this;

return 0.0;

}

public void set\_TreatmentCapacity( double TreatmentCapacity ) {

if (TreatmentCapacity == this.TreatmentCapacity) {

return;

}

double \_oldValue\_xjal = this.TreatmentCapacity;

this.TreatmentCapacity = TreatmentCapacity;

onChange\_TreatmentCapacity\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter TreatmentCapacity.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_TreatmentCapacity()</code> method instead.

\*/

protected void onChange\_TreatmentCapacity() {

onChange\_TreatmentCapacity\_xjal( TreatmentCapacity );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_TreatmentCapacity\_xjal( double oldValue ) {

}

public

double StorageCapacity;

/\*\*

\* Returns default value for parameter <code>StorageCapacity</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public double \_StorageCapacity\_DefaultValue\_xjal() {

final WaterTreatmentPlant self = this;

return 0.0;

}

public void set\_StorageCapacity( double StorageCapacity ) {

if (StorageCapacity == this.StorageCapacity) {

return;

}

double \_oldValue\_xjal = this.StorageCapacity;

this.StorageCapacity = StorageCapacity;

onChange\_StorageCapacity\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter StorageCapacity.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_StorageCapacity()</code> method instead.

\*/

protected void onChange\_StorageCapacity() {

onChange\_StorageCapacity\_xjal( StorageCapacity );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_StorageCapacity\_xjal( double oldValue ) {

}

@Override

public void setParametersToDefaultValues() {

super.setParametersToDefaultValues();

DesalinationCapacity = \_DesalinationCapacity\_DefaultValue\_xjal();

Name = \_Name\_DefaultValue\_xjal();

TreatmentCapacity = \_TreatmentCapacity\_DefaultValue\_xjal();

StorageCapacity = \_StorageCapacity\_DefaultValue\_xjal();

}

@Override

public boolean setParameter(String \_name\_xjal, Object \_value\_xjal, boolean \_callOnChange\_xjal) {

switch ( \_name\_xjal ) {

case "DesalinationCapacity":

if ( \_callOnChange\_xjal ) {

set\_DesalinationCapacity( ((Number) \_value\_xjal).doubleValue() );

} else {

DesalinationCapacity = ((Number) \_value\_xjal).doubleValue();

}

return true;

case "Name":

if ( \_callOnChange\_xjal ) {

set\_Name( (String) \_value\_xjal );

} else {

Name = (String) \_value\_xjal;

}

return true;

case "TreatmentCapacity":

if ( \_callOnChange\_xjal ) {

set\_TreatmentCapacity( ((Number) \_value\_xjal).doubleValue() );

} else {

TreatmentCapacity = ((Number) \_value\_xjal).doubleValue();

}

return true;

case "StorageCapacity":

if ( \_callOnChange\_xjal ) {

set\_StorageCapacity( ((Number) \_value\_xjal).doubleValue() );

} else {

StorageCapacity = ((Number) \_value\_xjal).doubleValue();

}

return true;

default:

return super.setParameter( \_name\_xjal, \_value\_xjal, \_callOnChange\_xjal );

}

}

@Override

public <T> T getParameter(String \_name\_xjal) {

Object \_result\_xjal;

switch ( \_name\_xjal ) {

case "DesalinationCapacity": \_result\_xjal = DesalinationCapacity; break;

case "Name": \_result\_xjal = Name; break;

case "TreatmentCapacity": \_result\_xjal = TreatmentCapacity; break;

case "StorageCapacity": \_result\_xjal = StorageCapacity; break;

default: \_result\_xjal = super.getParameter( \_name\_xjal ); break;

}

return (T) \_result\_xjal;

}

@AnyLogicInternalCodegenAPI

private static String[] \_parameterNames\_xjal;

@Override

public String[] getParameterNames() {

String[] result = \_parameterNames\_xjal;

if (result == null) {

List<String> list = new ArrayList<>( Arrays.asList( super.getParameterNames() ) );

list.add( "DesalinationCapacity" );

list.add( "Name" );

list.add( "TreatmentCapacity" );

list.add( "StorageCapacity" );

result = list.toArray( new String[ list.size() ] );

\_parameterNames\_xjal = result;

}

return result;

}

// Plain Variables

public

double

CurrentDesalinationCapacity;

public

double

PlannedDesalinationCapacity;

@AnyLogicInternalCodegenAPI

private static Map<String, IElementDescriptor> elementDesciptors\_xjal = createElementDescriptors( WaterTreatmentPlant.class );

@AnyLogicInternalCodegenAPI

@Override

public Map<String, IElementDescriptor> getElementDesciptors() {

return elementDesciptors\_xjal;

}

@AnyLogicCustomProposalPriority(type = AnyLogicCustomProposalPriority.Type.STATIC\_ELEMENT)

public static final Scale scale = new Scale( 10.0 );

@Override

public Scale getScale() {

return scale;

}

/\*\* Internal constant, shouldn't be accessed by user \*/

@AnyLogicInternalCodegenAPI

protected static final int \_STATECHART\_COUNT\_xjal = 0;

// Functions

double

decreaseDesalinationCapacity( double capacity ) {

if(capacity >= CurrentDesalinationCapacity){

CurrentDesalinationCapacity = 0;

return CurrentDesalinationCapacity;

}

else{

CurrentDesalinationCapacity -= capacity;

return capacity;

}

}

void increaseDesalinationCapacity( double capacity ) {

CurrentDesalinationCapacity+=capacity;

}

double

decreasePlannedDesalinationCapacity( double capacity ) {

if(capacity >= PlannedDesalinationCapacity){

PlannedDesalinationCapacity = 0;

return PlannedDesalinationCapacity;

}

else{

PlannedDesalinationCapacity -= capacity;

return capacity;

}

}

void setPlannedDC( double capacity ) {

PlannedDesalinationCapacity = capacity;

}

// View areas

public ViewArea \_origin\_VA = new ViewArea( this, "[Origin]", 0, 0, 1020.0, 640.0 );

@Override

@AnyLogicInternalCodegenAPI

public int getViewAreas(Map<String, ViewArea> \_output) {

if ( \_output != null ) {

\_output.put( "\_origin\_VA", this.\_origin\_VA );

}

return 1 + super.getViewAreas( \_output );

}

/\*\* Internal constant, shouldn't be accessed by user \*/

@AnyLogicInternalCodegenAPI

protected static final int \_SHAPE\_NEXT\_ID\_xjal = 1;

@AnyLogicInternalCodegenAPI

public boolean isPublicPresentationDefined() {

return false;

}

@AnyLogicInternalCodegenAPI

public boolean isEmbeddedAgentPresentationVisible( Agent \_a ) {

return super.isEmbeddedAgentPresentationVisible( \_a );

}

@AnyLogicInternalCodegenAPI

private void \_initialize\_level\_xjal() {

level.addAll();

}

protected com.anylogic.engine.markup.Level level;

private com.anylogic.engine.markup.Level[] \_getLevels\_xjal;

@Override

public com.anylogic.engine.markup.Level[] getLevels() {

return \_getLevels\_xjal;

}

@AnyLogicInternalCodegenAPI

private void \_createPersistentElementsBP0\_xjal() {

}

@AnyLogicInternalCodegenAPI

private void \_createPersistentElementsAP0\_xjal() {

}

@AnyLogicInternalCodegenAPI

private void \_createPersistentElementsBS0\_xjal() {

}

// Static initialization of persistent elements

{

level = new com.anylogic.engine.markup.Level(this, "level", SHAPE\_DRAW\_2D3D, 0.0, true, true);

\_getLevels\_xjal = new com.anylogic.engine.markup.Level[] {

level };

\_createPersistentElementsBP0\_xjal();

}

protected ShapeTopLevelPresentationGroup presentation;

protected ShapeModelElementsGroup icon;

@Override

@AnyLogicInternalCodegenAPI

public ShapeTopLevelPresentationGroup getPresentationShape() {

return presentation;

}

@Override

@AnyLogicInternalCodegenAPI

public ShapeModelElementsGroup getModelElementsShape() {

return icon;

}

/\*\*

\* Constructor

\*/

public WaterTreatmentPlant( Engine engine, Agent owner, AgentList<? extends WaterTreatmentPlant> ownerPopulation ) {

super( engine, owner, ownerPopulation );

instantiateBaseStructureThis\_xjal();

}

@AnyLogicInternalCodegenAPI

public void onOwnerChanged\_xjal() {

super.onOwnerChanged\_xjal();

setupReferences\_xjal();

}

@AnyLogicInternalCodegenAPI

public void instantiateBaseStructure\_xjal() {

super.instantiateBaseStructure\_xjal();

instantiateBaseStructureThis\_xjal();

}

@AnyLogicInternalCodegenAPI

private void instantiateBaseStructureThis\_xjal() {

setupReferences\_xjal();

}

@AnyLogicInternalCodegenAPI

private void setupReferences\_xjal() {

main = get\_Main();

}

/\*\*

\* Simple constructor. Please add created agent to some population by calling goToPopulation() function

\*/

public WaterTreatmentPlant() {

}

/\*\*

\* Simple constructor. Please add created agent to some population by calling goToPopulation() function

\*/

public WaterTreatmentPlant( double DesalinationCapacity, String Name, double TreatmentCapacity, double StorageCapacity ) {

markParametersAreSet();

this.DesalinationCapacity = DesalinationCapacity;

this.Name = Name;

this.TreatmentCapacity = TreatmentCapacity;

this.StorageCapacity = StorageCapacity;

}

@Override

@AnyLogicInternalCodegenAPI

public void doCreate() {

super.doCreate();

// Assigning initial values for plain variables

setupPlainVariables\_WaterTreatmentPlant\_xjal();

// Dynamic initialization of persistent elements

\_createPersistentElementsAP0\_xjal();

\_initialize\_level\_xjal();

level.initialize();

presentation = new ShapeTopLevelPresentationGroup( WaterTreatmentPlant.this, true, 0, 0, 0, 0 , level );

icon = new ShapeModelElementsGroup( WaterTreatmentPlant.this, getElementProperty( "wateradaptationmodelv8.WaterTreatmentPlant.icon", IElementDescriptor.MODEL\_ELEMENT\_DESCRIPTORS ) );

icon.setIconOffsets( 0.0, 0.0 );

// Port connectors with non-replicated objects

// Creating replicated embedded objects

setupInitialConditions\_xjal( WaterTreatmentPlant.class );

// Dynamic initialization of persistent elements

\_createPersistentElementsBS0\_xjal();

}

@Override

@AnyLogicInternalCodegenAPI

public void doStart() {

super.doStart();

}

/\*\*

\* Assigning initial values for plain variables<br>

\* <em>This method isn't designed to be called by user and may be removed in future releases.</em>

\*/

@AnyLogicInternalCodegenAPI

public void setupPlainVariables\_xjal() {

setupPlainVariables\_WaterTreatmentPlant\_xjal();

}

/\*\*

\* Assigning initial values for plain variables<br>

\* <em>This method isn't designed to be called by user and may be removed in future releases.</em>

\*/

@AnyLogicInternalCodegenAPI

private void setupPlainVariables\_WaterTreatmentPlant\_xjal() {

CurrentDesalinationCapacity =

DesalinationCapacity

;

PlannedDesalinationCapacity =

0

;

}

// User API -----------------------------------------------------

public Main get\_Main() {

{

Agent owner = getOwner();

if ( owner instanceof Main ) return (Main) owner;

}

return null;

}

/\*\*

\* Read-only variable. <em>Shouldn't be modified by user.</em>

\*/

@AnyLogicCustomSerialization(AnyLogicCustomSerializationMode.REFERENCE)

public transient wateradaptationmodelv8.Main main;

@AnyLogicInternalCodegenAPI

public static LinkToAgentAnimationSettings \_connections\_commonAnimationSettings\_xjal = new LinkToAgentAnimationSettingsImpl( false, black, 1.0, LINE\_STYLE\_SOLID, ARROW\_NONE, 0.0 );

public LinkToAgentCollection<Agent, Agent> connections = new LinkToAgentStandardImpl<Agent, Agent>(this, \_connections\_commonAnimationSettings\_xjal);

@Override

public LinkToAgentCollection<? extends Agent, ? extends Agent> getLinkToAgentStandard\_xjal() {

return connections;

}

@AnyLogicInternalCodegenAPI

public void drawLinksToAgents(boolean \_underAgents\_xjal, LinkToAgentAnimator \_animator\_xjal) {

super.drawLinksToAgents(\_underAgents\_xjal, \_animator\_xjal);

if ( \_underAgents\_xjal ) {

\_animator\_xjal.drawLink( this, connections, true, true );

}

}

public AgentList<? extends WaterTreatmentPlant> getPopulation() {

return (AgentList<? extends WaterTreatmentPlant>) super.getPopulation();

}

public List<? extends WaterTreatmentPlant> agentsInRange( double distance ) {

return (List<? extends WaterTreatmentPlant>) super.agentsInRange( distance );

}

}

**Wellfields (Infrastructure Agent):**

public class Well extends Agent

{

// Parameters

public

String Name;

/\*\*

\* Returns default value for parameter <code>Name</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public String \_Name\_DefaultValue\_xjal() {

final Well self = this;

return null;

}

public void set\_Name( String Name ) {

if (Name == this.Name) {

return;

}

String \_oldValue\_xjal = this.Name;

this.Name = Name;

onChange\_Name\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter Name.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_Name()</code> method instead.

\*/

protected void onChange\_Name() {

onChange\_Name\_xjal( Name );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_Name\_xjal( String oldValue ) {

}

public

double ExtractionCapacity;

/\*\*

\* Returns default value for parameter <code>ExtractionCapacity</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public double \_ExtractionCapacity\_DefaultValue\_xjal() {

final Well self = this;

return 0.0;

}

public void set\_ExtractionCapacity( double ExtractionCapacity ) {

if (ExtractionCapacity == this.ExtractionCapacity) {

return;

}

double \_oldValue\_xjal = this.ExtractionCapacity;

this.ExtractionCapacity = ExtractionCapacity;

onChange\_ExtractionCapacity\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter ExtractionCapacity.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_ExtractionCapacity()</code> method instead.

\*/

protected void onChange\_ExtractionCapacity() {

onChange\_ExtractionCapacity\_xjal( ExtractionCapacity );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_ExtractionCapacity\_xjal( double oldValue ) {

}

public

double LocationFromSWIL;

/\*\*

\* Returns default value for parameter <code>LocationFromSWIL</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public double \_LocationFromSWIL\_DefaultValue\_xjal() {

final Well self = this;

return 0.0;

}

public void set\_LocationFromSWIL( double LocationFromSWIL ) {

if (LocationFromSWIL == this.LocationFromSWIL) {

return;

}

double \_oldValue\_xjal = this.LocationFromSWIL;

this.LocationFromSWIL = LocationFromSWIL;

onChange\_LocationFromSWIL\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter LocationFromSWIL.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_LocationFromSWIL()</code> method instead.

\*/

protected void onChange\_LocationFromSWIL() {

onChange\_LocationFromSWIL\_xjal( LocationFromSWIL );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_LocationFromSWIL\_xjal( double oldValue ) {

}

public

double SalinatedProb;

/\*\*

\* Returns default value for parameter <code>SalinatedProb</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public double \_SalinatedProb\_DefaultValue\_xjal() {

final Well self = this;

return 0.0;

}

public void set\_SalinatedProb( double SalinatedProb ) {

if (SalinatedProb == this.SalinatedProb) {

return;

}

double \_oldValue\_xjal = this.SalinatedProb;

this.SalinatedProb = SalinatedProb;

onChange\_SalinatedProb\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter SalinatedProb.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_SalinatedProb()</code> method instead.

\*/

protected void onChange\_SalinatedProb() {

onChange\_SalinatedProb\_xjal( SalinatedProb );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_SalinatedProb\_xjal( double oldValue ) {

}

@Override

public void setParametersToDefaultValues() {

super.setParametersToDefaultValues();

Name = \_Name\_DefaultValue\_xjal();

ExtractionCapacity = \_ExtractionCapacity\_DefaultValue\_xjal();

LocationFromSWIL = \_LocationFromSWIL\_DefaultValue\_xjal();

SalinatedProb = \_SalinatedProb\_DefaultValue\_xjal();

}

@Override

public boolean setParameter(String \_name\_xjal, Object \_value\_xjal, boolean \_callOnChange\_xjal) {

switch ( \_name\_xjal ) {

case "Name":

if ( \_callOnChange\_xjal ) {

set\_Name( (String) \_value\_xjal );

} else {

Name = (String) \_value\_xjal;

}

return true;

case "ExtractionCapacity":

if ( \_callOnChange\_xjal ) {

set\_ExtractionCapacity( ((Number) \_value\_xjal).doubleValue() );

} else {

ExtractionCapacity = ((Number) \_value\_xjal).doubleValue();

}

return true;

case "LocationFromSWIL":

if ( \_callOnChange\_xjal ) {

set\_LocationFromSWIL( ((Number) \_value\_xjal).doubleValue() );

} else {

LocationFromSWIL = ((Number) \_value\_xjal).doubleValue();

}

return true;

case "SalinatedProb":

if ( \_callOnChange\_xjal ) {

set\_SalinatedProb( ((Number) \_value\_xjal).doubleValue() );

} else {

SalinatedProb = ((Number) \_value\_xjal).doubleValue();

}

return true;

default:

return super.setParameter( \_name\_xjal, \_value\_xjal, \_callOnChange\_xjal );

}

}

@Override

public <T> T getParameter(String \_name\_xjal) {

Object \_result\_xjal;

switch ( \_name\_xjal ) {

case "Name": \_result\_xjal = Name; break;

case "ExtractionCapacity": \_result\_xjal = ExtractionCapacity; break;

case "LocationFromSWIL": \_result\_xjal = LocationFromSWIL; break;

case "SalinatedProb": \_result\_xjal = SalinatedProb; break;

default: \_result\_xjal = super.getParameter( \_name\_xjal ); break;

}

return (T) \_result\_xjal;

}

@AnyLogicInternalCodegenAPI

private static String[] \_parameterNames\_xjal;

@Override

public String[] getParameterNames() {

String[] result = \_parameterNames\_xjal;

if (result == null) {

List<String> list = new ArrayList<>( Arrays.asList( super.getParameterNames() ) );

list.add( "Name" );

list.add( "ExtractionCapacity" );

list.add( "LocationFromSWIL" );

list.add( "SalinatedProb" );

result = list.toArray( new String[ list.size() ] );

\_parameterNames\_xjal = result;

}

return result;

}

// Plain Variables

public

double

CurrentExtractionCapacity;

public

double

CurrentLocFromSWIL;

/\*\*

\* this variable is set by the agency

\*/

public

int

PlannedAdaptIntrusionLine;

public

double

DesalinationCap;

public

double[]

YearExtractionCapacity;

public

int

WellRecoveryCount;

public

double

RelocationSWIL;

public

double

PlannedDesalinationCapacity;

public

int

PlannedDelay;

public

int

RecoveryDelay;

// Dynamic (Flow/Auxiliary/Stock) Variables

public double ClosingDistanceFromSWIL;

@AnyLogicInternalCodegenAPI

public void assignInitialConditions\_xjal() {

super.assignInitialConditions\_xjal();

\_assign\_ClosingDistanceFromSWIL\_Formula\_xjal();

}

@AnyLogicInternalCodegenAPI

public void setupInitialConditions\_xjal(Class<?> callerClass) {

if (callerClass != Well.class) {

return;

}

if (getInitialAlgebraicFlatEquationsCount\_xjal() > 0) {

SDIntegrationManager integrationManagerForInitialConditions = new SDIntegrationManager( 0, getInitialAlgebraicFlatEquationsCount\_xjal(), getInitialFormulaFlatEquationsCount\_xjal() );

integrationManagerForInitialConditions.doStep( this, 0, 0.1, true );

} else {

assignInitialConditions\_xjal();

}

}

@AnyLogicInternalCodegenAPI

public void \_assign\_ClosingDistanceFromSWIL\_Formula\_xjal() {

ClosingDistanceFromSWIL =

CurrentLocFromSWIL - main.Stressor.getCurrentSWIL()

;

}

@AnyLogicInternalCodegenAPI

public void formulasExecute\_xjal() {

super.formulasExecute\_xjal();

\_assign\_ClosingDistanceFromSWIL\_Formula\_xjal();

}

@AnyLogicInternalCodegenAPI

protected SDIntegrationManager integrationManager\_xjal = null;

@AnyLogicInternalCodegenAPI

public SDIntegrationManager getIntegrationManager\_xjal() {

if (integrationManager\_xjal == null) {

integrationManager\_xjal = new SDIntegrationManager( getDifferentialFlatEquationsCount\_xjal(), getRuntimeAlgebraicFlatEquationsCount\_xjal(), getRuntimeFormulaFlatEquationsCount\_xjal() );

}

return integrationManager\_xjal;

}

@Override

@AnyLogicInternalCodegenAPI

public int getRuntimeFormulaFlatEquationsCount\_xjal() {

return super.getRuntimeFormulaFlatEquationsCount\_xjal() + 1;

}

@Override

@AnyLogicInternalCodegenAPI

public int getInitialFormulaFlatEquationsCount\_xjal() {

return super.getInitialFormulaFlatEquationsCount\_xjal() + 1;

}

@AnyLogicInternalCodegenAPI

private static Map<String, IElementDescriptor> elementDesciptors\_xjal = createElementDescriptors( Well.class );

@AnyLogicInternalCodegenAPI

@Override

public Map<String, IElementDescriptor> getElementDesciptors() {

return elementDesciptors\_xjal;

}

@AnyLogicCustomProposalPriority(type = AnyLogicCustomProposalPriority.Type.STATIC\_ELEMENT)

public static final Scale scale = new Scale( 10.0 );

@Override

public Scale getScale() {

return scale;

}

// Events

public EventTimeout AccuExtractionCapacity = new EventTimeout(this);

@AnyLogicInternalCodegenAPI

public EventTimeout \_autoCreatedDS\_xjal = new EventTimeout(this);

@Override

@AnyLogicInternalCodegenAPI

public String getNameOf( EventTimeout \_e ) {

if( \_e == AccuExtractionCapacity ) return "AccuExtractionCapacity";

if( \_e == \_autoCreatedDS\_xjal ) return "Auto-created DataSets auto update event";

return super.getNameOf( \_e );

}

@Override

@AnyLogicInternalCodegenAPI

public EventTimeout.Mode getModeOf( EventTimeout \_e ) {

if ( \_e == AccuExtractionCapacity ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

if ( \_e == \_autoCreatedDS\_xjal ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

return super.getModeOf( \_e );

}

@Override

@AnyLogicInternalCodegenAPI

public double getFirstOccurrenceTime( EventTimeout \_e ) {

double \_t;

if ( \_e == AccuExtractionCapacity ) {

\_t =

0

;

\_t = toModelTime( \_t, DAY );

return \_t;

}

if ( \_e == \_autoCreatedDS\_xjal ) {

\_t =

0

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

return super.getFirstOccurrenceTime( \_e );

}

@Override

@AnyLogicInternalCodegenAPI

public double evaluateTimeoutOf( EventTimeout \_e ) {

double \_t;

if( \_e == AccuExtractionCapacity) {

\_t =

1

;

\_t = toModelTime( \_t, DAY );

return \_t;

}

if( \_e == \_autoCreatedDS\_xjal) {

\_t =

1

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

return super.evaluateTimeoutOf( \_e );

}

@Override

@AnyLogicInternalCodegenAPI

public void executeActionOf( EventTimeout \_e ) {

if ( \_e == AccuExtractionCapacity ) {

EventTimeout self = \_e;

int year = (int) main.current\_model\_year;

YearExtractionCapacity[year] += CurrentExtractionCapacity;

;

return;

}

if ( \_e == \_autoCreatedDS\_xjal ) {

\_ds\_ClosingDistanceFromSWIL.update();

return;

}

super.executeActionOf( \_e );

}

/\*\* Internal constant, shouldn't be accessed by user \*/

@AnyLogicInternalCodegenAPI

protected static final int \_STATECHART\_COUNT\_xjal = 1;

// Statecharts

public Statechart<WaterQuality\_state> WaterQuality = new Statechart<>( this, (short)2 );

@Override

@AnyLogicInternalCodegenAPI

public String getNameOf( Statechart \_s ) {

if(\_s == this.WaterQuality) return "WaterQuality";

return super.getNameOf( \_s );

}

@Override

@AnyLogicInternalCodegenAPI

public int getIdOf( Statechart \_s ) {

if(\_s == this.WaterQuality) return 0;

return super.getIdOf( \_s );

}

@Override

@AnyLogicInternalCodegenAPI

public void executeActionOf( Statechart \_s ) {

if( \_s == this.WaterQuality ) {

enterState( Potable, true );

return;

}

super.executeActionOf( \_s );

}

// States of all statecharts

public enum WaterQuality\_state implements IStatechartState<Well, WaterQuality\_state> {

Potable,

Contaminated,

AdaptationDelay,

DesalinationCapacity,

branch;

@AnyLogicInternalCodegenAPI

private Collection<WaterQuality\_state> \_simpleStatesDeep\_xjal;

@AnyLogicInternalCodegenAPI

private Set<WaterQuality\_state> \_fullState\_xjal;

@AnyLogicInternalCodegenAPI

private Set<WaterQuality\_state> \_statesInside\_xjal;

@Override

@AnyLogicInternalCodegenAPI

public Collection<WaterQuality\_state> getSimpleStatesDeep() {

Collection<WaterQuality\_state> result = \_simpleStatesDeep\_xjal;

if (result == null) {

\_simpleStatesDeep\_xjal = result = calculateAllSimpleStatesDeep();

}

return result;

}

@Override

public Set<WaterQuality\_state> getFullState() {

Set<WaterQuality\_state> result = \_fullState\_xjal;

if (result == null) {

\_fullState\_xjal = result = calculateFullState();

}

return result;

}

@Override

@AnyLogicInternalCodegenAPI

public Set<WaterQuality\_state> getStatesInside() {

Set<WaterQuality\_state> result = \_statesInside\_xjal;

if (result == null) {

\_statesInside\_xjal = result = calculateStatesInside();

}

return result;

}

@Override

@AnyLogicInternalCodegenAPI

public Statechart<WaterQuality\_state> getStatechart( Well \_a ) {

return \_a.WaterQuality;

}

}

@AnyLogicCustomProposalPriority(type = AnyLogicCustomProposalPriority.Type.STATIC\_ELEMENT)

public static final WaterQuality\_state Potable = WaterQuality\_state.Potable;

@AnyLogicCustomProposalPriority(type = AnyLogicCustomProposalPriority.Type.STATIC\_ELEMENT)

public static final WaterQuality\_state Contaminated = WaterQuality\_state.Contaminated;

@AnyLogicCustomProposalPriority(type = AnyLogicCustomProposalPriority.Type.STATIC\_ELEMENT)

public static final WaterQuality\_state AdaptationDelay = WaterQuality\_state.AdaptationDelay;

@AnyLogicCustomProposalPriority(type = AnyLogicCustomProposalPriority.Type.STATIC\_ELEMENT)

public static final WaterQuality\_state DesalinationCapacity = WaterQuality\_state.DesalinationCapacity;

@AnyLogicCustomProposalPriority(type = AnyLogicCustomProposalPriority.Type.STATIC\_ELEMENT)

public static final WaterQuality\_state branch = WaterQuality\_state.branch;

@AnyLogicInternalCodegenAPI

private void enterState( WaterQuality\_state self, boolean \_destination ) {

switch( self ) {

case Potable:

logToDBEnterState(WaterQuality, self);

// (Simple state (not composite))

WaterQuality.setActiveState\_xjal( Potable );

{

CurrentExtractionCapacity = ExtractionCapacity;

if(RelocationSWIL != 0){

//System.out.println("Applied relocation: "+RelocationSWIL);

CurrentLocFromSWIL += RelocationSWIL;

RelocationSWIL=0;

}

;}

transition1.start();

transition7.start();

return;

case Contaminated:

logToDBEnterState(WaterQuality, self);

// (Simple state (not composite))

WaterQuality.setActiveState\_xjal( Contaminated );

{

CurrentExtractionCapacity = 0;

DesalinationCap = 0;

;}

transition.start();

transition4.start();

return;

case AdaptationDelay:

logToDBEnterState(WaterQuality, self);

// (Simple state (not composite))

WaterQuality.setActiveState\_xjal( AdaptationDelay );

transition8.start();

return;

case DesalinationCapacity:

logToDBEnterState(WaterQuality, self);

// (Simple state (not composite))

WaterQuality.setActiveState\_xjal( DesalinationCapacity );

{

CurrentExtractionCapacity=DesalinationCap;

;}

desalination\_to\_contaminated.start();

return;

case branch:

logToDBEnterState(WaterQuality, self);

// (Branch)

if (

DesalinationCap > 0

) { // transition3

enterState( DesalinationCapacity, true );

return;

}

// transition2 (default)

enterState( Contaminated, true );

return;

default:

return;

}

}

@AnyLogicInternalCodegenAPI

private void exitState( WaterQuality\_state self, Transition \_t, boolean \_source ) {

switch( self ) {

case Potable:

logToDBExitState(WaterQuality, self);

logToDB(WaterQuality, \_t, self);

// (Simple state (not composite))

if ( !\_source || \_t != transition1) transition1.cancel();

if ( !\_source || \_t != transition7) transition7.cancel();

return;

case Contaminated:

logToDBExitState(WaterQuality, self);

logToDB(WaterQuality, \_t, self);

// (Simple state (not composite))

if ( !\_source || \_t != transition) transition.cancel();

if ( !\_source || \_t != transition4) transition4.cancel();

return;

case AdaptationDelay:

logToDBExitState(WaterQuality, self);

logToDB(WaterQuality, \_t, self);

// (Simple state (not composite))

if ( !\_source || \_t != transition8) transition8.cancel();

return;

case DesalinationCapacity:

logToDBExitState(WaterQuality, self);

logToDB(WaterQuality, \_t, self);

// (Simple state (not composite))

if ( !\_source || \_t != desalination\_to\_contaminated) desalination\_to\_contaminated.cancel();

return;

default:

return;

}

}

@AnyLogicInternalCodegenAPI

private void exitInnerStates( WaterQuality\_state \_destination ) {

WaterQuality\_state \_state = WaterQuality.getActiveSimpleState();

while( \_state != \_destination ) {

exitState( \_state, null, false );

\_state = \_state.getContainerState();

}

}

public TransitionTimeout transition8 = new TransitionTimeout( this );

public TransitionTimeout desalination\_to\_contaminated = new TransitionTimeout( this );

@Override

@AnyLogicInternalCodegenAPI

public String getNameOf( TransitionTimeout \_t ) {

if ( \_t == transition8 ) return "transition8";

if ( \_t == desalination\_to\_contaminated ) return "desalination\_to\_contaminated";

return super.getNameOf( \_t );

}

@Override

@AnyLogicInternalCodegenAPI

public Statechart getStatechartOf( TransitionTimeout \_t ) {

if ( \_t == transition8 ) return WaterQuality;

if ( \_t == desalination\_to\_contaminated ) return WaterQuality;

return super.getStatechartOf( \_t );

}

@Override

@AnyLogicInternalCodegenAPI

public void executeActionOf( TransitionTimeout self ) {

if ( self == transition8 ) {

exitState( AdaptationDelay, self, true );

enterState( Potable, true );

return;

}

if ( self == desalination\_to\_contaminated ) {

exitState( DesalinationCapacity, self, true );

enterState( Contaminated, true );

return;

}

super.executeActionOf( self );

}

@Override

@AnyLogicInternalCodegenAPI

public double evaluateTimeoutOf( TransitionTimeout \_t ) {

double \_value;

if ( \_t == transition8 ) {

\_value =

RecoveryDelay

;

\_value = toModelTime( \_value, YEAR );

return \_value;

}

if ( \_t == desalination\_to\_contaminated ) {

\_value =

1

;

\_value = toModelTime( \_value, YEAR );

return \_value;

}

return super.evaluateTimeoutOf( \_t );

}

public TransitionCondition transition1 = new TransitionCondition( this );

@Override

@AnyLogicInternalCodegenAPI

public String getNameOf( TransitionCondition \_t ) {

if ( \_t == transition1 ) return "transition1";

return super.getNameOf( \_t );

}

@Override

@AnyLogicInternalCodegenAPI

public Statechart getStatechartOf( TransitionCondition \_t ) {

if ( \_t == transition1 ) return WaterQuality;

return super.getStatechartOf( \_t );

}

@Override

@AnyLogicInternalCodegenAPI

public void executeActionOf( TransitionCondition self ) {

if ( self == transition1 ) {

exitState( Potable, self, true );

enterState( branch, true );

return;

}

super.executeActionOf( self );

}

@Override

@AnyLogicInternalCodegenAPI

public boolean testConditionOf( TransitionCondition \_t ) {

if ( \_t == transition1 ) return

ClosingDistanceFromSWIL < 0

;

return super.testConditionOf( \_t );

}

public TransitionMessage transition = new TransitionMessage( this );

public TransitionMessage transition4 = new TransitionMessage( this );

public TransitionMessage transition7 = new TransitionMessage( this );

@Override

@AnyLogicInternalCodegenAPI

public String getNameOf( TransitionMessage \_t ) {

if ( \_t == transition ) return "transition";

if ( \_t == transition4 ) return "transition4";

if ( \_t == transition7 ) return "transition7";

return super.getNameOf( \_t );

}

@Override

@AnyLogicInternalCodegenAPI

public Statechart getStatechartOf( TransitionMessage \_t ) {

if ( \_t == transition ) return WaterQuality;

if ( \_t == transition4 ) return WaterQuality;

if ( \_t == transition7 ) return WaterQuality;

return super.getStatechartOf( \_t );

}

@Override

@AnyLogicInternalCodegenAPI

public void executeActionOf( TransitionMessage self, Object \_msg ) {

if ( self == transition ) {

exitState( Contaminated, self, true );

enterState( AdaptationDelay, true );

return;

}

if ( self == transition4 ) {

exitState( Contaminated, self, true );

enterState( DesalinationCapacity, true );

return;

}

if ( self == transition7 ) {

exitState( Potable, self, true );

enterState( AdaptationDelay, true );

return;

}

super.executeActionOf( self, \_msg );

}

@Override

@AnyLogicInternalCodegenAPI

public boolean testMessageOf( TransitionMessage \_t, Object \_msg ) {

if ( \_t == transition ) {

Object

msg = (Object) \_msg;

Object \_g =

"recovery"

;

return msg.equals( \_g );

}

if ( \_t == transition4 ) {

Object

msg = (Object) \_msg;

Object \_g =

"desalination"

;

return msg.equals( \_g );

}

if ( \_t == transition7 ) {

Object

msg = (Object) \_msg;

Object \_g =

"recovery"

;

return msg.equals( \_g );

}

return super.testMessageOf( \_t, \_msg );

}

// Functions

boolean

isContaminated( ) {

return WaterQuality.getActiveSimpleState() == Contaminated;

}

boolean

isDesalinated( ) {

return WaterQuality.getActiveSimpleState() == DesalinationCapacity;

}

boolean

isPotable( ) {

return WaterQuality.getActiveSimpleState() == Potable;

}

void increaseLocFromSWIL( double relocation\_distance ) {

CurrentLocFromSWIL += relocation\_distance;

}

double

getPlannedExtractionCapacity( int slr\_scenario, int case\_scenario, int year\_i, double perceived\_ss\_intrusion ) {

int horizon = main.ChosenDecisionHorizon;

double accuExtraction = 0;

//calculate the supply for the well according to each year and the case scenario

int perceived\_swil = main.Stressor.getPerceivedSWIL(slr\_scenario, case\_scenario,

(int) main.current\_model\_year+year\_i);

double PlannedLocFromSWIL=CurrentLocFromSWIL + PlannedAdaptIntrusionLine;

double PlannedSWIL = perceived\_swil + perceived\_ss\_intrusion;

if((PlannedLocFromSWIL > PlannedSWIL) && (year\_i > PlannedDelay)){

accuExtraction += CurrentExtractionCapacity;

} else if ((PlannedDesalinationCapacity > 0) && (year\_i > PlannedDelay)){

//\*

accuExtraction += PlannedDesalinationCapacity;

}

return accuExtraction;

}

void setPlanAdaptIntrusionLine( int line ) {

PlannedAdaptIntrusionLine = line;

}

double

getPlannedExtractionRAU( int slr\_scenario, int case\_scenario, int year\_i, double perceived\_ss\_intrusion ) {

int horizon = main.ChosenDecisionHorizon;

double accuExtraction = 0;

//calculate the supply for the well according to each year and the case scenario

int perceived\_swil = main.Stressor.getPerceivedSWIL(slr\_scenario, case\_scenario,

(int) main.current\_model\_year+year\_i);

if((CurrentLocFromSWIL) > (perceived\_swil + perceived\_ss\_intrusion)){

accuExtraction = CurrentExtractionCapacity;

}

return accuExtraction;

}

void increaseDesCapacity( double capacity ) {

DesalinationCap = capacity;

WaterQuality.fireEvent("recovery");

//CurrentExtractionCapacity = DesalinationCap;

}

boolean

wellRecoveryLimit( ) {

//Did well recovery reach the limit?

return (WellRecoveryCount < main.WellRecoveryLimit);

}

void increaseRelocationSWIL( double swil ) {

RelocationSWIL = swil;

WaterQuality.fireEvent("recovery");

}

void setPlannedDesalinationCapacity( double cap ) {

PlannedDesalinationCapacity = cap;

}

void setPlannedDelay( int adapt\_id ) {

int min = (int) selectFrom(adaptation)

.where(adaptation.id.eq(adapt\_id))

.uniqueResult(adaptation.delay\_min);

int max = (int) selectFrom(adaptation)

.where(adaptation.id.eq(adapt\_id))

.uniqueResult(adaptation.delay\_max);

PlannedDelay = (int) uniform(min, max);

}

void setPlannedDelay2Zero( ) {

PlannedDelay = 0;

}

void increaseWellRecoveryCount( ) {

//Did well recovery reach the limit?

WellRecoveryCount++;

}

void setRecoveryDelay( int adapt\_id ) {

int min = (int) selectFrom(adaptation)

.where(adaptation.id.eq(adapt\_id))

.uniqueResult(adaptation.delay\_min);

int max = (int) selectFrom(adaptation)

.where(adaptation.id.eq(adapt\_id))

.uniqueResult(adaptation.delay\_max);

RecoveryDelay = (int) uniform(min, max);

}

/\*\*

\* Auto-created data set(s) for ClosingDistanceFromSWIL

\*/

@AnyLogicInternalCodegenAPI

public DataSet \_ds\_ClosingDistanceFromSWIL = new DataSet( 100, new DataUpdater\_xjal() {

double \_lastUpdateTime = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateTime ) { return; }

\_d.add( time(), Well.this.ClosingDistanceFromSWIL );

\_lastUpdateTime = time();

}

} );

// View areas

public ViewArea \_origin\_VA = new ViewArea( this, "[Origin]", 0, 0, 1020.0, 640.0 );

@Override

@AnyLogicInternalCodegenAPI

public int getViewAreas(Map<String, ViewArea> \_output) {

if ( \_output != null ) {

\_output.put( "\_origin\_VA", this.\_origin\_VA );

}

return 1 + super.getViewAreas( \_output );

}

/\*\* Internal constant, shouldn't be accessed by user \*/

@AnyLogicInternalCodegenAPI

protected static final int \_SHAPE\_NEXT\_ID\_xjal = 1;

@AnyLogicInternalCodegenAPI

public boolean isPublicPresentationDefined() {

return false;

}

@AnyLogicInternalCodegenAPI

public boolean isEmbeddedAgentPresentationVisible( Agent \_a ) {

return super.isEmbeddedAgentPresentationVisible( \_a );

}

@AnyLogicInternalCodegenAPI

private void \_initialize\_level\_xjal() {

level.addAll();

}

protected com.anylogic.engine.markup.Level level;

private com.anylogic.engine.markup.Level[] \_getLevels\_xjal;

@Override

public com.anylogic.engine.markup.Level[] getLevels() {

return \_getLevels\_xjal;

}

@AnyLogicInternalCodegenAPI

private void \_createPersistentElementsBP0\_xjal() {

}

@AnyLogicInternalCodegenAPI

private void \_createPersistentElementsAP0\_xjal() {

}

@AnyLogicInternalCodegenAPI

private void \_createPersistentElementsBS0\_xjal() {

}

// Static initialization of persistent elements

{

level = new com.anylogic.engine.markup.Level(this, "level", SHAPE\_DRAW\_2D3D, 0.0, true, true);

\_getLevels\_xjal = new com.anylogic.engine.markup.Level[] {

level };

\_createPersistentElementsBP0\_xjal();

}

protected ShapeTopLevelPresentationGroup presentation;

protected ShapeModelElementsGroup icon;

@Override

@AnyLogicInternalCodegenAPI

public ShapeTopLevelPresentationGroup getPresentationShape() {

return presentation;

}

@Override

@AnyLogicInternalCodegenAPI

public ShapeModelElementsGroup getModelElementsShape() {

return icon;

}

/\*\*

\* Constructor

\*/

public Well( Engine engine, Agent owner, AgentList<? extends Well> ownerPopulation ) {

super( engine, owner, ownerPopulation );

instantiateBaseStructureThis\_xjal();

}

@AnyLogicInternalCodegenAPI

public void onOwnerChanged\_xjal() {

super.onOwnerChanged\_xjal();

setupReferences\_xjal();

}

@AnyLogicInternalCodegenAPI

public void instantiateBaseStructure\_xjal() {

super.instantiateBaseStructure\_xjal();

instantiateBaseStructureThis\_xjal();

}

@AnyLogicInternalCodegenAPI

private void instantiateBaseStructureThis\_xjal() {

setupReferences\_xjal();

// Registering in Engine continuous part

getEngine().registerAgentWithEquations( this );

}

@AnyLogicInternalCodegenAPI

private void setupReferences\_xjal() {

main = get\_Main();

}

/\*\*

\* Simple constructor. Please add created agent to some population by calling goToPopulation() function

\*/

public Well() {

}

/\*\*

\* Simple constructor. Please add created agent to some population by calling goToPopulation() function

\*/

public Well( String Name, double ExtractionCapacity, double LocationFromSWIL, double SalinatedProb ) {

markParametersAreSet();

this.Name = Name;

this.ExtractionCapacity = ExtractionCapacity;

this.LocationFromSWIL = LocationFromSWIL;

this.SalinatedProb = SalinatedProb;

}

@Override

@AnyLogicInternalCodegenAPI

public void doCreate() {

super.doCreate();

// Assigning initial values for plain variables

setupPlainVariables\_Well\_xjal();

// Dynamic initialization of persistent elements

\_createPersistentElementsAP0\_xjal();

\_initialize\_level\_xjal();

level.initialize();

presentation = new ShapeTopLevelPresentationGroup( Well.this, true, 0, 0, 0, 0 , level );

icon = new ShapeModelElementsGroup( Well.this, getElementProperty( "wateradaptationmodelv8.Well.icon", IElementDescriptor.MODEL\_ELEMENT\_DESCRIPTORS ) );

icon.setIconOffsets( 0.0, 0.0 );

// Port connectors with non-replicated objects

// Creating replicated embedded objects

setupInitialConditions\_xjal( Well.class );

// Dynamic initialization of persistent elements

\_createPersistentElementsBS0\_xjal();

}

@Override

@AnyLogicInternalCodegenAPI

public void doStart() {

super.doStart();

AccuExtractionCapacity.start();

\_autoCreatedDS\_xjal.start();

WaterQuality.start();

}

/\*\*

\* Assigning initial values for plain variables<br>

\* <em>This method isn't designed to be called by user and may be removed in future releases.</em>

\*/

@AnyLogicInternalCodegenAPI

public void setupPlainVariables\_xjal() {

setupPlainVariables\_Well\_xjal();

}

/\*\*

\* Assigning initial values for plain variables<br>

\* <em>This method isn't designed to be called by user and may be removed in future releases.</em>

\*/

@AnyLogicInternalCodegenAPI

private void setupPlainVariables\_Well\_xjal() {

CurrentExtractionCapacity =

ExtractionCapacity

;

CurrentLocFromSWIL =

LocationFromSWIL

;

PlannedAdaptIntrusionLine =

0

;

DesalinationCap =

0

;

YearExtractionCapacity =

new double [101]

;

WellRecoveryCount =

0

;

RelocationSWIL =

0

;

PlannedDesalinationCapacity =

0

;

PlannedDelay =

0

;

RecoveryDelay =

0

;

}

// User API -----------------------------------------------------

public Main get\_Main() {

{

Agent owner = getOwner();

if ( owner instanceof Main ) return (Main) owner;

}

return null;

}

/\*\*

\* Read-only variable. <em>Shouldn't be modified by user.</em>

\*/

@AnyLogicCustomSerialization(AnyLogicCustomSerializationMode.REFERENCE)

public transient wateradaptationmodelv8.Main main;

@AnyLogicInternalCodegenAPI

public static LinkToAgentAnimationSettings \_connections\_commonAnimationSettings\_xjal = new LinkToAgentAnimationSettingsImpl( false, black, 1.0, LINE\_STYLE\_SOLID, ARROW\_NONE, 0.0 );

public LinkToAgentCollection<Agent, Agent> connections = new LinkToAgentStandardImpl<Agent, Agent>(this, \_connections\_commonAnimationSettings\_xjal);

@Override

public LinkToAgentCollection<? extends Agent, ? extends Agent> getLinkToAgentStandard\_xjal() {

return connections;

}

@Override

@AnyLogicInternalCodegenAPI

public void onReceive( Object \_msg\_xjal, Agent \_sender\_xjal ) {

super.onReceive( \_msg\_xjal, \_sender\_xjal );

WaterQuality.fireEvent( \_msg\_xjal );

}

@AnyLogicInternalCodegenAPI

public void drawLinksToAgents(boolean \_underAgents\_xjal, LinkToAgentAnimator \_animator\_xjal) {

super.drawLinksToAgents(\_underAgents\_xjal, \_animator\_xjal);

if ( \_underAgents\_xjal ) {

\_animator\_xjal.drawLink( this, connections, true, true );

}

}

public AgentList<? extends Well> getPopulation() {

return (AgentList<? extends Well>) super.getPopulation();

}

public List<? extends Well> agentsInRange( double distance ) {

return (List<? extends Well>) super.agentsInRange( distance );

}

@Override

@AnyLogicInternalCodegenAPI

public boolean isLoggingToDB(EventOriginator \_e) {

if ( \_e == \_autoCreatedDS\_xjal ) return false;

return super.isLoggingToDB( \_e );

}

// Reaction on changes -------------------------------------

public void onChange() {

super.onChange();

WaterQuality.onChange();

}

@AnyLogicInternalCodegenAPI

public void onDestroy() {

AccuExtractionCapacity.onDestroy();

\_autoCreatedDS\_xjal.onDestroy();

WaterQuality.onDestroy();

// Unregistering in Engine continuous part

getEngine().unregisterAgentWithEquations( this );

\_ds\_ClosingDistanceFromSWIL.destroyUpdater\_xjal();

super.onDestroy();

}

}

**Main (Agents Connection)**

public class Main extends Agent

{

// Excel Files

// Parameters

public

double ChosenCapitalFunding;

/\*\*

\* Returns default value for parameter <code>ChosenCapitalFunding</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public double \_ChosenCapitalFunding\_DefaultValue\_xjal() {

final Main self = this;

return 0.0;

}

public void set\_ChosenCapitalFunding( double ChosenCapitalFunding ) {

if (ChosenCapitalFunding == this.ChosenCapitalFunding) {

return;

}

double \_oldValue\_xjal = this.ChosenCapitalFunding;

this.ChosenCapitalFunding = ChosenCapitalFunding;

onChange\_ChosenCapitalFunding\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter ChosenCapitalFunding.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_ChosenCapitalFunding()</code> method instead.

\*/

protected void onChange\_ChosenCapitalFunding() {

onChange\_ChosenCapitalFunding\_xjal( ChosenCapitalFunding );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_ChosenCapitalFunding\_xjal( double oldValue ) {

int index;

{

Agency self = Agency;

double \_value;

\_value = ChosenCapitalFunding

;

Agency.set\_CapitalPlanFund( \_value );

}

}

public

int ChosenSLRScenario;

/\*\*

\* Returns default value for parameter <code>ChosenSLRScenario</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public int \_ChosenSLRScenario\_DefaultValue\_xjal() {

final Main self = this;

return 0;

}

public void set\_ChosenSLRScenario( int ChosenSLRScenario ) {

if (ChosenSLRScenario == this.ChosenSLRScenario) {

return;

}

int \_oldValue\_xjal = this.ChosenSLRScenario;

this.ChosenSLRScenario = ChosenSLRScenario;

onChange\_ChosenSLRScenario\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter ChosenSLRScenario.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_ChosenSLRScenario()</code> method instead.

\*/

protected void onChange\_ChosenSLRScenario() {

onChange\_ChosenSLRScenario\_xjal( ChosenSLRScenario );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_ChosenSLRScenario\_xjal( int oldValue ) {

int index;

{

Stressor self = Stressor;

double \_value;

\_value = ChosenSLRScenario

;

Stressor.set\_SLRScenario( \_value );

}

}

public

int ChosenCaseScenario;

/\*\*

\* Returns default value for parameter <code>ChosenCaseScenario</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public int \_ChosenCaseScenario\_DefaultValue\_xjal() {

final Main self = this;

return 0;

}

public void set\_ChosenCaseScenario( int ChosenCaseScenario ) {

if (ChosenCaseScenario == this.ChosenCaseScenario) {

return;

}

int \_oldValue\_xjal = this.ChosenCaseScenario;

this.ChosenCaseScenario = ChosenCaseScenario;

onChange\_ChosenCaseScenario\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter ChosenCaseScenario.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_ChosenCaseScenario()</code> method instead.

\*/

protected void onChange\_ChosenCaseScenario() {

onChange\_ChosenCaseScenario\_xjal( ChosenCaseScenario );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_ChosenCaseScenario\_xjal( int oldValue ) {

int index;

{

Stressor self = Stressor;

double \_value;

\_value = ChosenCaseScenario

;

Stressor.set\_CaseScenario( \_value );

}

}

public

int ChosenRiskAttitude;

/\*\*

\* Returns default value for parameter <code>ChosenRiskAttitude</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public int \_ChosenRiskAttitude\_DefaultValue\_xjal() {

final Main self = this;

return 0;

}

public void set\_ChosenRiskAttitude( int ChosenRiskAttitude ) {

if (ChosenRiskAttitude == this.ChosenRiskAttitude) {

return;

}

int \_oldValue\_xjal = this.ChosenRiskAttitude;

this.ChosenRiskAttitude = ChosenRiskAttitude;

onChange\_ChosenRiskAttitude\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter ChosenRiskAttitude.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_ChosenRiskAttitude()</code> method instead.

\*/

protected void onChange\_ChosenRiskAttitude() {

onChange\_ChosenRiskAttitude\_xjal( ChosenRiskAttitude );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_ChosenRiskAttitude\_xjal( int oldValue ) {

int index;

{

Agency self = Agency;

int \_value;

\_value = ChosenRiskAttitude

;

Agency.set\_RiskAttitude( \_value );

}

}

public

double ChosenWellRelocationDistance;

/\*\*

\* Returns default value for parameter <code>ChosenWellRelocationDistance</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public double \_ChosenWellRelocationDistance\_DefaultValue\_xjal() {

final Main self = this;

return 0.0;

}

public void set\_ChosenWellRelocationDistance( double ChosenWellRelocationDistance ) {

if (ChosenWellRelocationDistance == this.ChosenWellRelocationDistance) {

return;

}

double \_oldValue\_xjal = this.ChosenWellRelocationDistance;

this.ChosenWellRelocationDistance = ChosenWellRelocationDistance;

onChange\_ChosenWellRelocationDistance\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter ChosenWellRelocationDistance.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_ChosenWellRelocationDistance()</code> method instead.

\*/

protected void onChange\_ChosenWellRelocationDistance() {

onChange\_ChosenWellRelocationDistance\_xjal( ChosenWellRelocationDistance );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_ChosenWellRelocationDistance\_xjal( double oldValue ) {

}

public

int ChosenDecisionHorizon;

/\*\*

\* Returns default value for parameter <code>ChosenDecisionHorizon</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public int \_ChosenDecisionHorizon\_DefaultValue\_xjal() {

final Main self = this;

return 0;

}

public void set\_ChosenDecisionHorizon( int ChosenDecisionHorizon ) {

if (ChosenDecisionHorizon == this.ChosenDecisionHorizon) {

return;

}

int \_oldValue\_xjal = this.ChosenDecisionHorizon;

this.ChosenDecisionHorizon = ChosenDecisionHorizon;

onChange\_ChosenDecisionHorizon\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter ChosenDecisionHorizon.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_ChosenDecisionHorizon()</code> method instead.

\*/

protected void onChange\_ChosenDecisionHorizon() {

onChange\_ChosenDecisionHorizon\_xjal( ChosenDecisionHorizon );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_ChosenDecisionHorizon\_xjal( int oldValue ) {

int index;

{

Agency self = Agency;

int \_value;

\_value = ChosenDecisionHorizon

;

Agency.set\_DecisionHorizon( \_value );

}

{

User self = UserSouthMiamiDade;

double \_value;

\_value = ChosenDecisionHorizon

;

UserSouthMiamiDade.set\_ChosenDecisionHorizon( \_value );

}

}

public

double ChosenPopulationGrowth;

/\*\*

\* Returns default value for parameter <code>ChosenPopulationGrowth</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public double \_ChosenPopulationGrowth\_DefaultValue\_xjal() {

final Main self = this;

return 0.0;

}

public void set\_ChosenPopulationGrowth( double ChosenPopulationGrowth ) {

if (ChosenPopulationGrowth == this.ChosenPopulationGrowth) {

return;

}

double \_oldValue\_xjal = this.ChosenPopulationGrowth;

this.ChosenPopulationGrowth = ChosenPopulationGrowth;

onChange\_ChosenPopulationGrowth\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter ChosenPopulationGrowth.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_ChosenPopulationGrowth()</code> method instead.

\*/

protected void onChange\_ChosenPopulationGrowth() {

onChange\_ChosenPopulationGrowth\_xjal( ChosenPopulationGrowth );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_ChosenPopulationGrowth\_xjal( double oldValue ) {

int index;

{

User self = UserSouthMiamiDade;

double \_value;

\_value = ChosenPopulationGrowth/100

;

UserSouthMiamiDade.set\_PopulationGrowth( \_value );

}

}

public

int ChosenNoAdaptation;

/\*\*

\* Returns default value for parameter <code>ChosenNoAdaptation</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public int \_ChosenNoAdaptation\_DefaultValue\_xjal() {

final Main self = this;

return 0;

}

public void set\_ChosenNoAdaptation( int ChosenNoAdaptation ) {

if (ChosenNoAdaptation == this.ChosenNoAdaptation) {

return;

}

int \_oldValue\_xjal = this.ChosenNoAdaptation;

this.ChosenNoAdaptation = ChosenNoAdaptation;

onChange\_ChosenNoAdaptation\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter ChosenNoAdaptation.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_ChosenNoAdaptation()</code> method instead.

\*/

protected void onChange\_ChosenNoAdaptation() {

onChange\_ChosenNoAdaptation\_xjal( ChosenNoAdaptation );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_ChosenNoAdaptation\_xjal( int oldValue ) {

}

public

double ChosenRiskThreshold;

/\*\*

\* Returns default value for parameter <code>ChosenRiskThreshold</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public double \_ChosenRiskThreshold\_DefaultValue\_xjal() {

final Main self = this;

return 0.0;

}

public void set\_ChosenRiskThreshold( double ChosenRiskThreshold ) {

if (ChosenRiskThreshold == this.ChosenRiskThreshold) {

return;

}

double \_oldValue\_xjal = this.ChosenRiskThreshold;

this.ChosenRiskThreshold = ChosenRiskThreshold;

onChange\_ChosenRiskThreshold\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter ChosenRiskThreshold.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_ChosenRiskThreshold()</code> method instead.

\*/

protected void onChange\_ChosenRiskThreshold() {

onChange\_ChosenRiskThreshold\_xjal( ChosenRiskThreshold );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_ChosenRiskThreshold\_xjal( double oldValue ) {

int index;

{

Agency self = Agency;

double \_value;

\_value = ChosenRiskThreshold

;

Agency.set\_RiskThreshold( \_value );

}

}

public

int WellRecoveryLimit;

/\*\*

\* Returns default value for parameter <code>WellRecoveryLimit</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public int \_WellRecoveryLimit\_DefaultValue\_xjal() {

final Main self = this;

return

3

;

}

public void set\_WellRecoveryLimit( int WellRecoveryLimit ) {

if (WellRecoveryLimit == this.WellRecoveryLimit) {

return;

}

int \_oldValue\_xjal = this.WellRecoveryLimit;

this.WellRecoveryLimit = WellRecoveryLimit;

onChange\_WellRecoveryLimit\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter WellRecoveryLimit.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_WellRecoveryLimit()</code> method instead.

\*/

protected void onChange\_WellRecoveryLimit() {

onChange\_WellRecoveryLimit\_xjal( WellRecoveryLimit );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_WellRecoveryLimit\_xjal( int oldValue ) {

}

public

int RecoveryDelayMin;

/\*\*

\* Returns default value for parameter <code>RecoveryDelayMin</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public int \_RecoveryDelayMin\_DefaultValue\_xjal() {

final Main self = this;

return

1

;

}

public void set\_RecoveryDelayMin( int RecoveryDelayMin ) {

if (RecoveryDelayMin == this.RecoveryDelayMin) {

return;

}

int \_oldValue\_xjal = this.RecoveryDelayMin;

this.RecoveryDelayMin = RecoveryDelayMin;

onChange\_RecoveryDelayMin\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter RecoveryDelayMin.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_RecoveryDelayMin()</code> method instead.

\*/

protected void onChange\_RecoveryDelayMin() {

onChange\_RecoveryDelayMin\_xjal( RecoveryDelayMin );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_RecoveryDelayMin\_xjal( int oldValue ) {

}

public

int RecoveryDelayMax;

/\*\*

\* Returns default value for parameter <code>RecoveryDelayMax</code>.

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

public int \_RecoveryDelayMax\_DefaultValue\_xjal() {

final Main self = this;

return

3

;

}

public void set\_RecoveryDelayMax( int RecoveryDelayMax ) {

if (RecoveryDelayMax == this.RecoveryDelayMax) {

return;

}

int \_oldValue\_xjal = this.RecoveryDelayMax;

this.RecoveryDelayMax = RecoveryDelayMax;

onChange\_RecoveryDelayMax\_xjal( \_oldValue\_xjal );

onChange();

}

/\*\*

\* Calls "On change" action for parameter RecoveryDelayMax.<br>

\* Note that 'oldValue' in that action will be unavailable if this method is called by user

\* (current parameter value will be passed as 'oldValue').<br>

\* Please call <code>set\_RecoveryDelayMax()</code> method instead.

\*/

protected void onChange\_RecoveryDelayMax() {

onChange\_RecoveryDelayMax\_xjal( RecoveryDelayMax );

}

@AnyLogicInternalCodegenAPI

protected void onChange\_RecoveryDelayMax\_xjal( int oldValue ) {

}

@Override

public void setParametersToDefaultValues() {

super.setParametersToDefaultValues();

ChosenCapitalFunding = \_ChosenCapitalFunding\_DefaultValue\_xjal();

ChosenSLRScenario = \_ChosenSLRScenario\_DefaultValue\_xjal();

ChosenCaseScenario = \_ChosenCaseScenario\_DefaultValue\_xjal();

ChosenRiskAttitude = \_ChosenRiskAttitude\_DefaultValue\_xjal();

ChosenWellRelocationDistance = \_ChosenWellRelocationDistance\_DefaultValue\_xjal();

ChosenDecisionHorizon = \_ChosenDecisionHorizon\_DefaultValue\_xjal();

ChosenPopulationGrowth = \_ChosenPopulationGrowth\_DefaultValue\_xjal();

ChosenNoAdaptation = \_ChosenNoAdaptation\_DefaultValue\_xjal();

ChosenRiskThreshold = \_ChosenRiskThreshold\_DefaultValue\_xjal();

WellRecoveryLimit = \_WellRecoveryLimit\_DefaultValue\_xjal();

RecoveryDelayMin = \_RecoveryDelayMin\_DefaultValue\_xjal();

RecoveryDelayMax = \_RecoveryDelayMax\_DefaultValue\_xjal();

}

@Override

public boolean setParameter(String \_name\_xjal, Object \_value\_xjal, boolean \_callOnChange\_xjal) {

switch ( \_name\_xjal ) {

case "ChosenCapitalFunding":

if ( \_callOnChange\_xjal ) {

set\_ChosenCapitalFunding( ((Number) \_value\_xjal).doubleValue() );

} else {

ChosenCapitalFunding = ((Number) \_value\_xjal).doubleValue();

}

return true;

case "ChosenSLRScenario":

if ( \_callOnChange\_xjal ) {

set\_ChosenSLRScenario( ((Number) \_value\_xjal).intValue() );

} else {

ChosenSLRScenario = ((Number) \_value\_xjal).intValue();

}

return true;

case "ChosenCaseScenario":

if ( \_callOnChange\_xjal ) {

set\_ChosenCaseScenario( ((Number) \_value\_xjal).intValue() );

} else {

ChosenCaseScenario = ((Number) \_value\_xjal).intValue();

}

return true;

case "ChosenRiskAttitude":

if ( \_callOnChange\_xjal ) {

set\_ChosenRiskAttitude( ((Number) \_value\_xjal).intValue() );

} else {

ChosenRiskAttitude = ((Number) \_value\_xjal).intValue();

}

return true;

case "ChosenWellRelocationDistance":

if ( \_callOnChange\_xjal ) {

set\_ChosenWellRelocationDistance( ((Number) \_value\_xjal).doubleValue() );

} else {

ChosenWellRelocationDistance = ((Number) \_value\_xjal).doubleValue();

}

return true;

case "ChosenDecisionHorizon":

if ( \_callOnChange\_xjal ) {

set\_ChosenDecisionHorizon( ((Number) \_value\_xjal).intValue() );

} else {

ChosenDecisionHorizon = ((Number) \_value\_xjal).intValue();

}

return true;

case "ChosenPopulationGrowth":

if ( \_callOnChange\_xjal ) {

set\_ChosenPopulationGrowth( ((Number) \_value\_xjal).doubleValue() );

} else {

ChosenPopulationGrowth = ((Number) \_value\_xjal).doubleValue();

}

return true;

case "ChosenNoAdaptation":

if ( \_callOnChange\_xjal ) {

set\_ChosenNoAdaptation( ((Number) \_value\_xjal).intValue() );

} else {

ChosenNoAdaptation = ((Number) \_value\_xjal).intValue();

}

return true;

case "ChosenRiskThreshold":

if ( \_callOnChange\_xjal ) {

set\_ChosenRiskThreshold( ((Number) \_value\_xjal).doubleValue() );

} else {

ChosenRiskThreshold = ((Number) \_value\_xjal).doubleValue();

}

return true;

case "WellRecoveryLimit":

if ( \_callOnChange\_xjal ) {

set\_WellRecoveryLimit( ((Number) \_value\_xjal).intValue() );

} else {

WellRecoveryLimit = ((Number) \_value\_xjal).intValue();

}

return true;

case "RecoveryDelayMin":

if ( \_callOnChange\_xjal ) {

set\_RecoveryDelayMin( ((Number) \_value\_xjal).intValue() );

} else {

RecoveryDelayMin = ((Number) \_value\_xjal).intValue();

}

return true;

case "RecoveryDelayMax":

if ( \_callOnChange\_xjal ) {

set\_RecoveryDelayMax( ((Number) \_value\_xjal).intValue() );

} else {

RecoveryDelayMax = ((Number) \_value\_xjal).intValue();

}

return true;

default:

return super.setParameter( \_name\_xjal, \_value\_xjal, \_callOnChange\_xjal );

}

}

@Override

public <T> T getParameter(String \_name\_xjal) {

Object \_result\_xjal;

switch ( \_name\_xjal ) {

case "ChosenCapitalFunding": \_result\_xjal = ChosenCapitalFunding; break;

case "ChosenSLRScenario": \_result\_xjal = ChosenSLRScenario; break;

case "ChosenCaseScenario": \_result\_xjal = ChosenCaseScenario; break;

case "ChosenRiskAttitude": \_result\_xjal = ChosenRiskAttitude; break;

case "ChosenWellRelocationDistance": \_result\_xjal = ChosenWellRelocationDistance; break;

case "ChosenDecisionHorizon": \_result\_xjal = ChosenDecisionHorizon; break;

case "ChosenPopulationGrowth": \_result\_xjal = ChosenPopulationGrowth; break;

case "ChosenNoAdaptation": \_result\_xjal = ChosenNoAdaptation; break;

case "ChosenRiskThreshold": \_result\_xjal = ChosenRiskThreshold; break;

case "WellRecoveryLimit": \_result\_xjal = WellRecoveryLimit; break;

case "RecoveryDelayMin": \_result\_xjal = RecoveryDelayMin; break;

case "RecoveryDelayMax": \_result\_xjal = RecoveryDelayMax; break;

default: \_result\_xjal = super.getParameter( \_name\_xjal ); break;

}

return (T) \_result\_xjal;

}

@AnyLogicInternalCodegenAPI

private static String[] \_parameterNames\_xjal;

@Override

public String[] getParameterNames() {

String[] result = \_parameterNames\_xjal;

if (result == null) {

List<String> list = new ArrayList<>( Arrays.asList( super.getParameterNames() ) );

list.add( "ChosenCapitalFunding" );

list.add( "ChosenSLRScenario" );

list.add( "ChosenCaseScenario" );

list.add( "ChosenRiskAttitude" );

list.add( "ChosenWellRelocationDistance" );

list.add( "ChosenDecisionHorizon" );

list.add( "ChosenPopulationGrowth" );

list.add( "ChosenNoAdaptation" );

list.add( "ChosenRiskThreshold" );

list.add( "WellRecoveryLimit" );

list.add( "RecoveryDelayMin" );

list.add( "RecoveryDelayMax" );

result = list.toArray( new String[ list.size() ] );

\_parameterNames\_xjal = result;

}

return result;

}

// Plain Variables

public

int

SheetIndex;

public

double

total\_level\_of\_supply;

public

double

total\_demand;

public

int

Row;

public

double

sum\_los;

// Collection Variables

public

TreeMap <

Integer ,

Integer > StormSurgeBCData = new TreeMap<Integer, Integer>();

public

ArrayList <

String > RowHeader = new ArrayList<String>();

// Dynamic (Flow/Auxiliary/Stock) Variables

public double current\_model\_year;

public double reliability\_index;

public double year\_supply;

public double year\_demand;

public double level\_of\_service;

public double supply\_over\_demand;

@AnyLogicInternalCodegenAPI

public void assignInitialConditions\_xjal() {

super.assignInitialConditions\_xjal();

\_assign\_current\_model\_year\_Formula\_xjal();

\_assign\_year\_supply\_Formula\_xjal();

\_assign\_year\_demand\_Formula\_xjal();

\_assign\_reliability\_index\_Formula\_xjal();

\_assign\_level\_of\_service\_Formula\_xjal();

\_assign\_supply\_over\_demand\_Formula\_xjal();

}

@AnyLogicInternalCodegenAPI

public void setupInitialConditions\_xjal(Class<?> callerClass) {

if (callerClass != Main.class) {

return;

}

if (getInitialAlgebraicFlatEquationsCount\_xjal() > 0) {

SDIntegrationManager integrationManagerForInitialConditions = new SDIntegrationManager( 0, getInitialAlgebraicFlatEquationsCount\_xjal(), getInitialFormulaFlatEquationsCount\_xjal() );

integrationManagerForInitialConditions.doStep( this, 0, 0.1, true );

} else {

assignInitialConditions\_xjal();

}

}

@AnyLogicInternalCodegenAPI

public void \_assign\_current\_model\_year\_Formula\_xjal() {

current\_model\_year =

(int)Math.ceil(time(YEAR))

;

}

@AnyLogicInternalCodegenAPI

public void \_assign\_year\_supply\_Formula\_xjal() {

year\_supply =

calculateLevelofSupply()

;

}

@AnyLogicInternalCodegenAPI

public void \_assign\_year\_demand\_Formula\_xjal() {

year\_demand =

UserSouthMiamiDade.getYearlyConsumption()

;

}

@AnyLogicInternalCodegenAPI

public void \_assign\_reliability\_index\_Formula\_xjal() {

reliability\_index =

(current\_model\_year != 0)?(sum\_los / current\_model\_year):0

;

}

@AnyLogicInternalCodegenAPI

public void \_assign\_level\_of\_service\_Formula\_xjal() {

level\_of\_service =

(total\_demand != 0)?(total\_level\_of\_supply / total\_demand) \* 100:0

;

}

@AnyLogicInternalCodegenAPI

public void \_assign\_supply\_over\_demand\_Formula\_xjal() {

supply\_over\_demand =

(year\_demand != 0)?(year\_supply / year\_demand) \* 100:0

;

}

@AnyLogicInternalCodegenAPI

public void formulasExecute\_xjal() {

super.formulasExecute\_xjal();

\_assign\_current\_model\_year\_Formula\_xjal();

\_assign\_year\_supply\_Formula\_xjal();

\_assign\_year\_demand\_Formula\_xjal();

\_assign\_reliability\_index\_Formula\_xjal();

\_assign\_level\_of\_service\_Formula\_xjal();

\_assign\_supply\_over\_demand\_Formula\_xjal();

}

@AnyLogicInternalCodegenAPI

protected SDIntegrationManager integrationManager\_xjal = null;

@AnyLogicInternalCodegenAPI

public SDIntegrationManager getIntegrationManager\_xjal() {

if (integrationManager\_xjal == null) {

integrationManager\_xjal = new SDIntegrationManager( getDifferentialFlatEquationsCount\_xjal(), getRuntimeAlgebraicFlatEquationsCount\_xjal(), getRuntimeFormulaFlatEquationsCount\_xjal() );

}

return integrationManager\_xjal;

}

@Override

@AnyLogicInternalCodegenAPI

public int getRuntimeFormulaFlatEquationsCount\_xjal() {

return super.getRuntimeFormulaFlatEquationsCount\_xjal() + 6;

}

@Override

@AnyLogicInternalCodegenAPI

public int getInitialFormulaFlatEquationsCount\_xjal() {

return super.getInitialFormulaFlatEquationsCount\_xjal() + 6;

}

@AnyLogicInternalCodegenAPI

private static Map<String, IElementDescriptor> elementDesciptors\_xjal = createElementDescriptors( Main.class );

@AnyLogicInternalCodegenAPI

@Override

public Map<String, IElementDescriptor> getElementDesciptors() {

return elementDesciptors\_xjal;

}

@AnyLogicCustomProposalPriority(type = AnyLogicCustomProposalPriority.Type.STATIC\_ELEMENT)

public static final Scale scale = new Scale( 10.0 );

@Override

public Scale getScale() {

return scale;

}

// Events

public EventTimeout update\_demand\_los = new EventTimeout(this);

@AnyLogicInternalCodegenAPI

public EventTimeout \_SWILDataset\_autoUpdateEvent\_xjal = new EventTimeout(this);

@AnyLogicInternalCodegenAPI

public EventTimeout \_BudgetDataset\_autoUpdateEvent\_xjal = new EventTimeout(this);

@AnyLogicInternalCodegenAPI

public EventTimeout \_Supply\_over\_demand\_autoUpdateEvent\_xjal = new EventTimeout(this);

@AnyLogicInternalCodegenAPI

public EventTimeout \_RiskUpdating\_autoUpdateEvent\_xjal = new EventTimeout(this);

@AnyLogicInternalCodegenAPI

public EventTimeout \_year\_well\_supply\_autoUpdateEvent\_xjal = new EventTimeout(this);

@AnyLogicInternalCodegenAPI

public EventTimeout \_year\_demand\_dataset\_autoUpdateEvent\_xjal = new EventTimeout(this);

@AnyLogicInternalCodegenAPI

public EventTimeout \_actual\_sr\_autoUpdateEvent\_xjal = new EventTimeout(this);

@AnyLogicInternalCodegenAPI

public EventTimeout \_swil\_plot\_autoUpdateEvent\_xjal = new EventTimeout(this);

@AnyLogicInternalCodegenAPI

public EventTimeout \_budget\_plot\_autoUpdateEvent\_xjal = new EventTimeout(this);

@AnyLogicInternalCodegenAPI

public EventTimeout \_chart\_autoUpdateEvent\_xjal = new EventTimeout(this);

@AnyLogicInternalCodegenAPI

public EventTimeout \_plot\_autoUpdateEvent\_xjal = new EventTimeout(this);

@AnyLogicInternalCodegenAPI

public EventTimeout \_plot1\_autoUpdateEvent\_xjal = new EventTimeout(this);

@AnyLogicInternalCodegenAPI

public EventTimeout \_plot2\_autoUpdateEvent\_xjal = new EventTimeout(this);

@AnyLogicInternalCodegenAPI

public EventTimeout \_chart1\_autoUpdateEvent\_xjal = new EventTimeout(this);

@AnyLogicInternalCodegenAPI

public EventTimeout \_chart2\_autoUpdateEvent\_xjal = new EventTimeout(this);

@AnyLogicInternalCodegenAPI

public EventTimeout \_plot3\_autoUpdateEvent\_xjal = new EventTimeout(this);

@AnyLogicInternalCodegenAPI

public EventTimeout \_plot4\_autoUpdateEvent\_xjal = new EventTimeout(this);

@AnyLogicInternalCodegenAPI

public EventTimeout \_autoCreatedDS\_xjal = new EventTimeout(this);

@Override

@AnyLogicInternalCodegenAPI

public String getNameOf( EventTimeout \_e ) {

if( \_e == update\_demand\_los ) return "update\_demand\_los";

if( \_e == \_SWILDataset\_autoUpdateEvent\_xjal ) return "SWILDataset auto update event";

if( \_e == \_BudgetDataset\_autoUpdateEvent\_xjal ) return "BudgetDataset auto update event";

if( \_e == \_Supply\_over\_demand\_autoUpdateEvent\_xjal ) return "Supply\_over\_demand auto update event";

if( \_e == \_RiskUpdating\_autoUpdateEvent\_xjal ) return "RiskUpdating auto update event";

if( \_e == \_year\_well\_supply\_autoUpdateEvent\_xjal ) return "year\_well\_supply auto update event";

if( \_e == \_year\_demand\_dataset\_autoUpdateEvent\_xjal ) return "year\_demand\_dataset auto update event";

if( \_e == \_actual\_sr\_autoUpdateEvent\_xjal ) return "actual\_sr auto update event";

if( \_e == \_swil\_plot\_autoUpdateEvent\_xjal ) return "swil\_plot auto update event";

if( \_e == \_budget\_plot\_autoUpdateEvent\_xjal ) return "budget\_plot auto update event";

if( \_e == \_chart\_autoUpdateEvent\_xjal ) return "chart auto update event";

if( \_e == \_plot\_autoUpdateEvent\_xjal ) return "plot auto update event";

if( \_e == \_plot1\_autoUpdateEvent\_xjal ) return "plot1 auto update event";

if( \_e == \_plot2\_autoUpdateEvent\_xjal ) return "plot2 auto update event";

if( \_e == \_chart1\_autoUpdateEvent\_xjal ) return "chart1 auto update event";

if( \_e == \_chart2\_autoUpdateEvent\_xjal ) return "chart2 auto update event";

if( \_e == \_plot3\_autoUpdateEvent\_xjal ) return "plot3 auto update event";

if( \_e == \_plot4\_autoUpdateEvent\_xjal ) return "plot4 auto update event";

if( \_e == \_autoCreatedDS\_xjal ) return "Auto-created DataSets auto update event";

return super.getNameOf( \_e );

}

@Override

@AnyLogicInternalCodegenAPI

public EventTimeout.Mode getModeOf( EventTimeout \_e ) {

if ( \_e == update\_demand\_los ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

if ( \_e == \_SWILDataset\_autoUpdateEvent\_xjal ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

if ( \_e == \_BudgetDataset\_autoUpdateEvent\_xjal ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

if ( \_e == \_Supply\_over\_demand\_autoUpdateEvent\_xjal ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

if ( \_e == \_RiskUpdating\_autoUpdateEvent\_xjal ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

if ( \_e == \_year\_well\_supply\_autoUpdateEvent\_xjal ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

if ( \_e == \_year\_demand\_dataset\_autoUpdateEvent\_xjal ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

if ( \_e == \_actual\_sr\_autoUpdateEvent\_xjal ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

if ( \_e == \_swil\_plot\_autoUpdateEvent\_xjal ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

if ( \_e == \_budget\_plot\_autoUpdateEvent\_xjal ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

if ( \_e == \_chart\_autoUpdateEvent\_xjal ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

if ( \_e == \_plot\_autoUpdateEvent\_xjal ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

if ( \_e == \_plot1\_autoUpdateEvent\_xjal ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

if ( \_e == \_plot2\_autoUpdateEvent\_xjal ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

if ( \_e == \_chart1\_autoUpdateEvent\_xjal ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

if ( \_e == \_chart2\_autoUpdateEvent\_xjal ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

if ( \_e == \_plot3\_autoUpdateEvent\_xjal ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

if ( \_e == \_plot4\_autoUpdateEvent\_xjal ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

if ( \_e == \_autoCreatedDS\_xjal ) return EVENT\_TIMEOUT\_MODE\_CYCLIC;

return super.getModeOf( \_e );

}

@Override

@AnyLogicInternalCodegenAPI

public double getFirstOccurrenceTime( EventTimeout \_e ) {

double \_t;

if ( \_e == update\_demand\_los ) {

\_t =

1

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if ( \_e == \_SWILDataset\_autoUpdateEvent\_xjal ) {

\_t =

0

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if ( \_e == \_BudgetDataset\_autoUpdateEvent\_xjal ) {

\_t =

0

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if ( \_e == \_Supply\_over\_demand\_autoUpdateEvent\_xjal ) {

\_t =

0

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if ( \_e == \_RiskUpdating\_autoUpdateEvent\_xjal ) {

\_t =

0

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if ( \_e == \_year\_well\_supply\_autoUpdateEvent\_xjal ) {

\_t =

0

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if ( \_e == \_year\_demand\_dataset\_autoUpdateEvent\_xjal ) {

\_t =

0

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if ( \_e == \_actual\_sr\_autoUpdateEvent\_xjal ) {

\_t =

ChosenDecisionHorizon

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if ( \_e == \_swil\_plot\_autoUpdateEvent\_xjal ) {

\_t =

0

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if ( \_e == \_budget\_plot\_autoUpdateEvent\_xjal ) {

\_t =

0

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if ( \_e == \_chart\_autoUpdateEvent\_xjal ) {

\_t =

0

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if ( \_e == \_plot\_autoUpdateEvent\_xjal ) {

\_t =

0

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if ( \_e == \_plot1\_autoUpdateEvent\_xjal ) {

\_t =

0

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if ( \_e == \_plot2\_autoUpdateEvent\_xjal ) {

\_t =

0

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if ( \_e == \_chart1\_autoUpdateEvent\_xjal ) {

\_t =

0

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if ( \_e == \_chart2\_autoUpdateEvent\_xjal ) {

\_t =

0

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if ( \_e == \_plot3\_autoUpdateEvent\_xjal ) {

\_t =

0

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if ( \_e == \_plot4\_autoUpdateEvent\_xjal ) {

\_t =

0

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if ( \_e == \_autoCreatedDS\_xjal ) {

\_t =

0

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

return super.getFirstOccurrenceTime( \_e );

}

@Override

@AnyLogicInternalCodegenAPI

public double evaluateTimeoutOf( EventTimeout \_e ) {

double \_t;

if( \_e == update\_demand\_los) {

\_t =

1

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if( \_e == \_SWILDataset\_autoUpdateEvent\_xjal) {

\_t =

1

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if( \_e == \_BudgetDataset\_autoUpdateEvent\_xjal) {

\_t =

1

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if( \_e == \_Supply\_over\_demand\_autoUpdateEvent\_xjal) {

\_t =

1

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if( \_e == \_RiskUpdating\_autoUpdateEvent\_xjal) {

\_t =

ChosenDecisionHorizon

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if( \_e == \_year\_well\_supply\_autoUpdateEvent\_xjal) {

\_t =

1

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if( \_e == \_year\_demand\_dataset\_autoUpdateEvent\_xjal) {

\_t =

1

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if( \_e == \_actual\_sr\_autoUpdateEvent\_xjal) {

\_t =

ChosenDecisionHorizon

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if( \_e == \_swil\_plot\_autoUpdateEvent\_xjal) {

\_t =

1

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if( \_e == \_budget\_plot\_autoUpdateEvent\_xjal) {

\_t =

1

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if( \_e == \_chart\_autoUpdateEvent\_xjal) {

\_t =

1

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if( \_e == \_plot\_autoUpdateEvent\_xjal) {

\_t =

1

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if( \_e == \_plot1\_autoUpdateEvent\_xjal) {

\_t =

1

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if( \_e == \_plot2\_autoUpdateEvent\_xjal) {

\_t =

1

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if( \_e == \_chart1\_autoUpdateEvent\_xjal) {

\_t =

1

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if( \_e == \_chart2\_autoUpdateEvent\_xjal) {

\_t =

1

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if( \_e == \_plot3\_autoUpdateEvent\_xjal) {

\_t =

ChosenDecisionHorizon

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if( \_e == \_plot4\_autoUpdateEvent\_xjal) {

\_t =

1

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

if( \_e == \_autoCreatedDS\_xjal) {

\_t =

1

;

\_t = toModelTime( \_t, YEAR );

return \_t;

}

return super.evaluateTimeoutOf( \_e );

}

@Override

@AnyLogicInternalCodegenAPI

public void executeActionOf( EventTimeout \_e ) {

if ( \_e == update\_demand\_los ) {

EventTimeout self = \_e;

total\_demand += UserSouthMiamiDade.getYearlyConsumption();

total\_level\_of\_supply += calculateLevelofSupply();

sum\_los += supply\_over\_demand;

;

return;

}

if ( \_e == \_SWILDataset\_autoUpdateEvent\_xjal ) {

SWILDataset.update();

return;

}

if ( \_e == \_BudgetDataset\_autoUpdateEvent\_xjal ) {

BudgetDataset.update();

return;

}

if ( \_e == \_Supply\_over\_demand\_autoUpdateEvent\_xjal ) {

Supply\_over\_demand.update();

return;

}

if ( \_e == \_RiskUpdating\_autoUpdateEvent\_xjal ) {

RiskUpdating.update();

return;

}

if ( \_e == \_year\_well\_supply\_autoUpdateEvent\_xjal ) {

year\_well\_supply.update();

return;

}

if ( \_e == \_year\_demand\_dataset\_autoUpdateEvent\_xjal ) {

year\_demand\_dataset.update();

return;

}

if ( \_e == \_actual\_sr\_autoUpdateEvent\_xjal ) {

actual\_sr.update();

return;

}

if ( \_e == \_swil\_plot\_autoUpdateEvent\_xjal ) {

swil\_plot.updateData();

return;

}

if ( \_e == \_budget\_plot\_autoUpdateEvent\_xjal ) {

budget\_plot.updateData();

return;

}

if ( \_e == \_chart\_autoUpdateEvent\_xjal ) {

chart.updateData();

return;

}

if ( \_e == \_plot\_autoUpdateEvent\_xjal ) {

plot.updateData();

return;

}

if ( \_e == \_plot1\_autoUpdateEvent\_xjal ) {

plot1.updateData();

return;

}

if ( \_e == \_plot2\_autoUpdateEvent\_xjal ) {

plot2.updateData();

return;

}

if ( \_e == \_chart1\_autoUpdateEvent\_xjal ) {

chart1.updateData();

return;

}

if ( \_e == \_chart2\_autoUpdateEvent\_xjal ) {

chart2.updateData();

return;

}

if ( \_e == \_plot3\_autoUpdateEvent\_xjal ) {

plot3.updateData();

return;

}

if ( \_e == \_plot4\_autoUpdateEvent\_xjal ) {

plot4.updateData();

return;

}

if ( \_e == \_autoCreatedDS\_xjal ) {

\_ds\_current\_model\_year.update();

\_ds\_reliability\_index.update();

\_ds\_year\_supply.update();

\_ds\_year\_demand.update();

\_ds\_level\_of\_service.update();

\_ds\_supply\_over\_demand.update();

return;

}

super.executeActionOf( \_e );

}

/\*\* Internal constant, shouldn't be accessed by user \*/

@AnyLogicInternalCodegenAPI

protected static final int \_STATECHART\_COUNT\_xjal = 0;

// Embedded Objects

public Agency Agency;

public WaterTreatmentPlant WTPSouthMiamiHeights;

public User UserSouthMiamiDade;

public Stressor Stressor;

public Well WellSouthMiamiHeights;

public Well WellRobertaHunter;

public Well WellFormerPlant;

public String getNameOf( Agent ao ) {

if ( ao == Agency ) return "Agency";

if ( ao == WTPSouthMiamiHeights ) return "WTPSouthMiamiHeights";

if ( ao == UserSouthMiamiDade ) return "UserSouthMiamiDade";

if ( ao == Stressor ) return "Stressor";

if ( ao == WellSouthMiamiHeights ) return "WellSouthMiamiHeights";

if ( ao == WellRobertaHunter ) return "WellRobertaHunter";

if ( ao == WellFormerPlant ) return "WellFormerPlant";

return super.getNameOf( ao );

}

public AgentAnimationSettings getAnimationSettingsOf( Agent ao ) {

return super.getAnimationSettingsOf( ao );

}

public String getNameOf( AgentList<?> aolist ) {

return super.getNameOf( aolist );

}

public AgentAnimationSettings getAnimationSettingsOf( AgentList<?> aolist ) {

return super.getAnimationSettingsOf( aolist );

}

/\*\*

\* Creates an embedded object instance<br>

\* <i>This method should not be called by user</i>

\*/

protected Agency instantiate\_Agency\_xjal() {

Agency \_result\_xjal = new Agency( getEngine(), this, null );

return \_result\_xjal;

}

/\*\*

\* Setups parameters of an embedded object instance<br>

\* This method should not be called by user

\*/

private void setupParameters\_Agency\_xjal( final Agency self, TableInput \_t ) {

self.CapitalPlanFund =

ChosenCapitalFunding

;

self.RiskAttitude =

ChosenRiskAttitude

;

self.DecisionHorizon =

ChosenDecisionHorizon

;

self.RiskThreshold =

ChosenRiskThreshold

;

self.RiskUpdateLimit = self.\_RiskUpdateLimit\_DefaultValue\_xjal();

}

/\*\*

\* Setups an embedded object instance<br>

\* This method should not be called by user

\*/

@AnyLogicInternalCodegenAPI

private void doBeforeCreate\_Agency\_xjal( Agency self, TableInput \_t ) {

self.setEnvironment( this );

}

/\*\*

\* Creates an embedded object instance<br>

\* <i>This method should not be called by user</i>

\*/

protected WaterTreatmentPlant instantiate\_WTPSouthMiamiHeights\_xjal() {

WaterTreatmentPlant \_result\_xjal = new WaterTreatmentPlant( getEngine(), this, null );

return \_result\_xjal;

}

/\*\*

\* Setups parameters of an embedded object instance<br>

\* This method should not be called by user

\*/

private void setupParameters\_WTPSouthMiamiHeights\_xjal( final WaterTreatmentPlant self, TableInput \_t ) {

self.DesalinationCapacity =

0

;

self.Name =

"South Miami Heights"

;

self.TreatmentCapacity =

24100000

;

self.StorageCapacity =

5000000

;

}

/\*\*

\* Setups an embedded object instance<br>

\* This method should not be called by user

\*/

@AnyLogicInternalCodegenAPI

private void doBeforeCreate\_WTPSouthMiamiHeights\_xjal( WaterTreatmentPlant self, TableInput \_t ) {

self.setEnvironment( this );

}

/\*\*

\* Creates an embedded object instance<br>

\* <i>This method should not be called by user</i>

\*/

protected User instantiate\_UserSouthMiamiDade\_xjal() {

User \_result\_xjal = new User( getEngine(), this, null );

return \_result\_xjal;

}

/\*\*

\* Setups parameters of an embedded object instance<br>

\* This method should not be called by user

\*/

private void setupParameters\_UserSouthMiamiDade\_xjal( final User self, TableInput \_t ) {

self.ConsumptionPerCapita =

137.2

;

self.Name =

"South Miami-Dade"

;

self.Population =

78690

;

self.PopulationGrowth =

ChosenPopulationGrowth/100

;

self.ChosenDecisionHorizon =

ChosenDecisionHorizon

;

}

/\*\*

\* Setups an embedded object instance<br>

\* This method should not be called by user

\*/

@AnyLogicInternalCodegenAPI

private void doBeforeCreate\_UserSouthMiamiDade\_xjal( User self, TableInput \_t ) {

self.setEnvironment( this );

}

/\*\*

\* Creates an embedded object instance<br>

\* <i>This method should not be called by user</i>

\*/

protected Stressor instantiate\_Stressor\_xjal() {

Stressor \_result\_xjal = new Stressor( getEngine(), this, null );

return \_result\_xjal;

}

/\*\*

\* Setups parameters of an embedded object instance<br>

\* This method should not be called by user

\*/

private void setupParameters\_Stressor\_xjal( final Stressor self, TableInput \_t ) {

self.SLRScenario =

ChosenSLRScenario

;

self.CaseScenario =

ChosenCaseScenario

;

}

/\*\*

\* Setups an embedded object instance<br>

\* This method should not be called by user

\*/

@AnyLogicInternalCodegenAPI

private void doBeforeCreate\_Stressor\_xjal( Stressor self, TableInput \_t ) {

self.setEnvironment( this );

}

/\*\*

\* Creates an embedded object instance<br>

\* <i>This method should not be called by user</i>

\*/

protected Well instantiate\_WellSouthMiamiHeights\_xjal() {

Well \_result\_xjal = new Well( getEngine(), this, null );

return \_result\_xjal;

}

/\*\*

\* Setups parameters of an embedded object instance<br>

\* This method should not be called by user

\*/

private void setupParameters\_WellSouthMiamiHeights\_xjal( final Well self, TableInput \_t ) {

self.Name =

"South Miami Heights"

;

self.ExtractionCapacity =

24000000

;

self.LocationFromSWIL =

3200

;

self.SalinatedProb =

0.2

;

}

/\*\*

\* Setups an embedded object instance<br>

\* This method should not be called by user

\*/

@AnyLogicInternalCodegenAPI

private void doBeforeCreate\_WellSouthMiamiHeights\_xjal( Well self, TableInput \_t ) {

self.setEnvironment( this );

}

/\*\*

\* Creates an embedded object instance<br>

\* <i>This method should not be called by user</i>

\*/

protected Well instantiate\_WellRobertaHunter\_xjal() {

Well \_result\_xjal = new Well( getEngine(), this, null );

return \_result\_xjal;

}

/\*\*

\* Setups parameters of an embedded object instance<br>

\* This method should not be called by user

\*/

private void setupParameters\_WellRobertaHunter\_xjal( final Well self, TableInput \_t ) {

self.Name =

"Roberta Hunter Plant"

;

self.ExtractionCapacity =

6000000

;

self.LocationFromSWIL =

3200

;

self.SalinatedProb =

0.2

;

}

/\*\*

\* Setups an embedded object instance<br>

\* This method should not be called by user

\*/

@AnyLogicInternalCodegenAPI

private void doBeforeCreate\_WellRobertaHunter\_xjal( Well self, TableInput \_t ) {

self.setEnvironment( this );

}

/\*\*

\* Creates an embedded object instance<br>

\* <i>This method should not be called by user</i>

\*/

protected Well instantiate\_WellFormerPlant\_xjal() {

Well \_result\_xjal = new Well( getEngine(), this, null );

return \_result\_xjal;

}

/\*\*

\* Setups parameters of an embedded object instance<br>

\* This method should not be called by user

\*/

private void setupParameters\_WellFormerPlant\_xjal( final Well self, TableInput \_t ) {

self.Name =

"Former Plan"

;

self.ExtractionCapacity =

4000000

;

self.LocationFromSWIL =

4400

;

self.SalinatedProb =

0.2

;

}

/\*\*

\* Setups an embedded object instance<br>

\* This method should not be called by user

\*/

@AnyLogicInternalCodegenAPI

private void doBeforeCreate\_WellFormerPlant\_xjal( Well self, TableInput \_t ) {

self.setEnvironment( this );

}

// Functions

double

calculateLevelofSupply( ) {

double demand = UserSouthMiamiDade.getYearlyConsumption();

double supply = 0;

double totalExtractionCap = 0;

//Go through all the agents in Main

for (Agent a : this.agents()){

//Check agent is a Well

if (a instanceof Well) {

Well well = (Well) a;

//totalExtractionCap += well.CurrentExtractionCapacity;

totalExtractionCap += well.YearExtractionCapacity[(int)current\_model\_year];

}

}

if(totalExtractionCap/365 > WTPSouthMiamiHeights.TreatmentCapacity)

supply = WTPSouthMiamiHeights.TreatmentCapacity\*365;

//else

//supply = totalExtractionCap\*365;

if(supply > demand)

return demand;

else

return supply;

}

double

calculateTotalSupply( ) {

double demand = UserSouthMiamiDade.getYearlyConsumption();

double supply = 0;

double totalExtractionCap = 0;

//Go through all the agents in Main

for (Agent a : this.agents()){

//Check agent is a Well

if (a instanceof Well) {

Well well = (Well) a;

totalExtractionCap += well.CurrentExtractionCapacity;

}

}

return totalExtractionCap\*365;

}

void setRelocationInDB( ) {

update(adaptation)

.where(adaptation.name.eq("Relocation"))

.set(adaptation.adaptation\_intrusion\_line\_ft, ChosenWellRelocationDistance)

.execute();

}

int

adaptMeasureCount( int adapt\_id ) {

int counter=0;

for(int i=0; i<AdaptationYear.size(); i++){

if(AdaptationYear.getY(i) == adapt\_id){

counter+=1;

}

}

return counter;

}

String

getSLRName( ) {

return (String) selectFirstValue(String.class,

"SELECT name FROM slr\_scenario WHERE " +

"id = ?; ",

ChosenSLRScenario

);

}

String

getCaseName( ) {

return (String) selectFirstValue(String.class,

"SELECT actual FROM case\_scenario WHERE " +

"id = ?; ",

ChosenCaseScenario

);

}

void totalLevelofSupply( ) {

}

// Analysis Data Elements

/\*\*

\* Auto-created data set(s) for current\_model\_year

\*/

@AnyLogicInternalCodegenAPI

public DataSet \_ds\_current\_model\_year = new DataSet( 100, new DataUpdater\_xjal() {

double \_lastUpdateTime = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateTime ) { return; }

\_d.add( time(), Main.this.current\_model\_year );

\_lastUpdateTime = time();

}

} );

/\*\*

\* Auto-created data set(s) for reliability\_index

\*/

@AnyLogicInternalCodegenAPI

public DataSet \_ds\_reliability\_index = new DataSet( 100, new DataUpdater\_xjal() {

double \_lastUpdateTime = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateTime ) { return; }

\_d.add( time(), Main.this.reliability\_index );

\_lastUpdateTime = time();

}

} );

/\*\*

\* Auto-created data set(s) for year\_supply

\*/

@AnyLogicInternalCodegenAPI

public DataSet \_ds\_year\_supply = new DataSet( 100, new DataUpdater\_xjal() {

double \_lastUpdateTime = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateTime ) { return; }

\_d.add( time(), Main.this.year\_supply );

\_lastUpdateTime = time();

}

} );

/\*\*

\* Auto-created data set(s) for year\_demand

\*/

@AnyLogicInternalCodegenAPI

public DataSet \_ds\_year\_demand = new DataSet( 100, new DataUpdater\_xjal() {

double \_lastUpdateTime = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateTime ) { return; }

\_d.add( time(), Main.this.year\_demand );

\_lastUpdateTime = time();

}

} );

/\*\*

\* Auto-created data set(s) for level\_of\_service

\*/

@AnyLogicInternalCodegenAPI

public DataSet \_ds\_level\_of\_service = new DataSet( 100, new DataUpdater\_xjal() {

double \_lastUpdateTime = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateTime ) { return; }

\_d.add( time(), Main.this.level\_of\_service );

\_lastUpdateTime = time();

}

} );

/\*\*

\* Auto-created data set(s) for supply\_over\_demand

\*/

@AnyLogicInternalCodegenAPI

public DataSet \_ds\_supply\_over\_demand = new DataSet( 100, new DataUpdater\_xjal() {

double \_lastUpdateTime = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateTime ) { return; }

\_d.add( time(), Main.this.supply\_over\_demand );

\_lastUpdateTime = time();

}

} );

@AnyLogicInternalCodegenAPI

public DataSet \_plot\_expression0\_dataSet\_xjal = new DataSet( 100, new DataUpdater\_xjal() {

double \_lastUpdateX = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateX ) { return; }

\_d.add( time(), \_\_plot\_expression0\_dataSet\_xjal\_YValue() );

\_lastUpdateX = time();

}

} );

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_\_plot\_expression0\_dataSet\_xjal\_YValue() {

return

WellSouthMiamiHeights.ClosingDistanceFromSWIL

;

}

@AnyLogicInternalCodegenAPI

public DataSet \_plot1\_expression0\_dataSet\_xjal = new DataSet( 100, new DataUpdater\_xjal() {

double \_lastUpdateX = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateX ) { return; }

\_d.add( time(), \_\_plot1\_expression0\_dataSet\_xjal\_YValue() );

\_lastUpdateX = time();

}

} );

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_\_plot1\_expression0\_dataSet\_xjal\_YValue() {

return

WellRobertaHunter.ClosingDistanceFromSWIL

;

}

@AnyLogicInternalCodegenAPI

public DataSet \_plot2\_expression0\_dataSet\_xjal = new DataSet( 100, new DataUpdater\_xjal() {

double \_lastUpdateX = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateX ) { return; }

\_d.add( time(), \_\_plot2\_expression0\_dataSet\_xjal\_YValue() );

\_lastUpdateX = time();

}

} );

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_\_plot2\_expression0\_dataSet\_xjal\_YValue() {

return

WellFormerPlant.ClosingDistanceFromSWIL

;

}

@AnyLogicInternalCodegenAPI

public DataSet \_chart2\_expression0\_dataSet\_xjal = new DataSet( 100, new DataUpdater\_xjal() {

double \_lastUpdateX = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateX ) { return; }

\_d.add( time(), \_\_chart2\_expression0\_dataSet\_xjal\_YValue() );

\_lastUpdateX = time();

}

} );

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_\_chart2\_expression0\_dataSet\_xjal\_YValue() {

return

WellSouthMiamiHeights.YearExtractionCapacity[(int)current\_model\_year]

;

}

@AnyLogicInternalCodegenAPI

public DataSet \_chart2\_expression1\_dataSet\_xjal = new DataSet( 100, new DataUpdater\_xjal() {

double \_lastUpdateX = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateX ) { return; }

\_d.add( time(), \_\_chart2\_expression1\_dataSet\_xjal\_YValue() );

\_lastUpdateX = time();

}

} );

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_\_chart2\_expression1\_dataSet\_xjal\_YValue() {

return

WellRobertaHunter.YearExtractionCapacity[(int)current\_model\_year]

;

}

@AnyLogicInternalCodegenAPI

public DataSet \_chart2\_expression2\_dataSet\_xjal = new DataSet( 100, new DataUpdater\_xjal() {

double \_lastUpdateX = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateX ) { return; }

\_d.add( time(), \_\_chart2\_expression2\_dataSet\_xjal\_YValue() );

\_lastUpdateX = time();

}

} );

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_\_chart2\_expression2\_dataSet\_xjal\_YValue() {

return

WellFormerPlant.YearExtractionCapacity[(int)current\_model\_year]

;

}

public DataSet SWILDataset = new DataSet( 101, new DataUpdater\_xjal() {

double \_lastUpdateX = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateX ) { return; }

\_d.add( \_SWILDataset\_XValue(), \_SWILDataset\_YValue() );

\_lastUpdateX = time();

}

} );

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_SWILDataset\_XValue() {

return

current\_model\_year

;

}

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_SWILDataset\_YValue() {

return

Stressor.getCurrentSWIL()

;

}

public DataSet BudgetDataset = new DataSet( 100, new DataUpdater\_xjal() {

double \_lastUpdateX = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateX ) { return; }

\_d.add( \_BudgetDataset\_XValue(), \_BudgetDataset\_YValue() );

\_lastUpdateX = time();

}

} );

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_BudgetDataset\_XValue() {

return

current\_model\_year

;

}

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_BudgetDataset\_YValue() {

return

Agency.Budget

;

}

public DataSet Supply\_over\_demand = new DataSet( 101, new DataUpdater\_xjal() {

double \_lastUpdateX = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateX ) { return; }

\_d.add( \_Supply\_over\_demand\_XValue(), \_Supply\_over\_demand\_YValue() );

\_lastUpdateX = time();

}

} );

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_Supply\_over\_demand\_XValue() {

return

current\_model\_year

;

}

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_Supply\_over\_demand\_YValue() {

return

(year\_demand != 0) ? (year\_supply/year\_demand)\*100 : 0

;

}

public DataSet AdaptationYear = new DataSet( 500, new DataUpdater\_xjal() {

double \_lastUpdateX = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateX ) { return; }

\_d.add( time(), 0 );

\_lastUpdateX = time();

}

@Override

public double getDataXValue() {

return time();

}

} );

public DataSet RiskUpdating = new DataSet( 100, new DataUpdater\_xjal() {

double \_lastUpdateX = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateX ) { return; }

\_d.add( time(), \_RiskUpdating\_YValue() );

\_lastUpdateX = time();

}

@Override

public double getDataXValue() {

return time();

}

} );

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_RiskUpdating\_YValue() {

return

Agency.CurrentRiskAttitude()

;

}

public DataSet year\_well\_supply = new DataSet( 100, new DataUpdater\_xjal() {

double \_lastUpdateX = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateX ) { return; }

\_d.add( time(), \_year\_well\_supply\_YValue() );

\_lastUpdateX = time();

}

@Override

public double getDataXValue() {

return time();

}

} );

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_year\_well\_supply\_YValue() {

return

year\_supply

;

}

public DataSet year\_demand\_dataset = new DataSet( 100, new DataUpdater\_xjal() {

double \_lastUpdateX = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateX ) { return; }

\_d.add( time(), \_year\_demand\_dataset\_YValue() );

\_lastUpdateX = time();

}

@Override

public double getDataXValue() {

return time();

}

} );

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_year\_demand\_dataset\_YValue() {

return

year\_demand

;

}

public DataSet actual\_sr = new DataSet( 100, new DataUpdater\_xjal() {

double \_lastUpdateX = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateX ) { return; }

\_d.add( time(), \_actual\_sr\_YValue() );

\_lastUpdateX = time();

}

@Override

public double getDataXValue() {

return time();

}

} );

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_actual\_sr\_YValue() {

return

reliability\_index/100

;

}

public DataSet perceived\_sr = new DataSet( 100, new DataUpdater\_xjal() {

double \_lastUpdateX = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateX ) { return; }

\_d.add( time(), 0 );

\_lastUpdateX = time();

}

@Override

public double getDataXValue() {

return time();

}

} );

public DataSet pswi\_dataset = new DataSet( 100, new DataUpdater\_xjal() {

double \_lastUpdateX = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateX ) { return; }

\_d.add( time(), 0 );

\_lastUpdateX = time();

}

@Override

public double getDataXValue() {

return time();

}

} );

public DataSet aswi\_min\_dataset = new DataSet( 100, new DataUpdater\_xjal() {

double \_lastUpdateX = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateX ) { return; }

\_d.add( time(), 0 );

\_lastUpdateX = time();

}

@Override

public double getDataXValue() {

return time();

}

} );

public DataSet aswi\_max\_dataset = new DataSet( 100, new DataUpdater\_xjal() {

double \_lastUpdateX = Double.NaN;

@Override

public void update( DataSet \_d ) {

if ( time() == \_lastUpdateX ) { return; }

\_d.add( time(), 0 );

\_lastUpdateX = time();

}

@Override

public double getDataXValue() {

return time();

}

} );

// View areas

public ViewArea \_origin\_VA = new ViewArea( this, "[Origin]", 0, 0, 1020.0, 640.0 );

@Override

@AnyLogicInternalCodegenAPI

public int getViewAreas(Map<String, ViewArea> \_output) {

if ( \_output != null ) {

\_output.put( "\_origin\_VA", this.\_origin\_VA );

}

return 1 + super.getViewAreas( \_output );

}

@AnyLogicInternalCodegenAPI

protected static final Font \_text\_Font = new Font("SansSerif", 0, 12 );

@AnyLogicInternalCodegenAPI

protected static final Font \_text1\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text2\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text3\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text4\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text5\_Font = new Font("SansSerif", 1, 12 );

@AnyLogicInternalCodegenAPI

protected static final Font \_text6\_Font = \_text5\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text7\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text8\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text9\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text10\_Font = new Font("arial", 0, 12 );

@AnyLogicInternalCodegenAPI

protected static final Font \_text11\_Font = \_text5\_Font;

@AnyLogicInternalCodegenAPI

protected static final int \_swil\_plot = 1;

@AnyLogicInternalCodegenAPI

protected static final int \_budget\_plot = 2;

@AnyLogicInternalCodegenAPI

protected static final int \_chart = 3;

@AnyLogicInternalCodegenAPI

protected static final int \_plot = 4;

@AnyLogicInternalCodegenAPI

protected static final int \_plot1 = 5;

@AnyLogicInternalCodegenAPI

protected static final int \_plot2 = 6;

@AnyLogicInternalCodegenAPI

protected static final int \_adaptationyearplot = 7;

@AnyLogicInternalCodegenAPI

protected static final int \_chart1 = 8;

@AnyLogicInternalCodegenAPI

protected static final int \_chart2 = 9;

@AnyLogicInternalCodegenAPI

protected static final int \_plot3 = 10;

@AnyLogicInternalCodegenAPI

protected static final int \_plot4 = 11;

@AnyLogicInternalCodegenAPI

protected static final int \_plot5 = 12;

@AnyLogicInternalCodegenAPI

protected static final int \_plot6 = 13;

@AnyLogicInternalCodegenAPI

protected static final int \_rectangle = 14;

@AnyLogicInternalCodegenAPI

protected static final int \_text = 15;

@AnyLogicInternalCodegenAPI

protected static final int \_text1 = 16;

@AnyLogicInternalCodegenAPI

protected static final int \_text2 = 17;

@AnyLogicInternalCodegenAPI

protected static final int \_text3 = 18;

@AnyLogicInternalCodegenAPI

protected static final int \_text4 = 19;

@AnyLogicInternalCodegenAPI

protected static final int \_text5 = 20;

@AnyLogicInternalCodegenAPI

protected static final int \_text6 = 21;

@AnyLogicInternalCodegenAPI

protected static final int \_text7 = 22;

@AnyLogicInternalCodegenAPI

protected static final int \_text8 = 23;

@AnyLogicInternalCodegenAPI

protected static final int \_text9 = 24;

@AnyLogicInternalCodegenAPI

protected static final int \_text10 = 25;

@AnyLogicInternalCodegenAPI

protected static final int \_text11 = 26;

/\*\* Internal constant, shouldn't be accessed by user \*/

@AnyLogicInternalCodegenAPI

protected static final int \_SHAPE\_NEXT\_ID\_xjal = 27;

@AnyLogicInternalCodegenAPI

public boolean isPublicPresentationDefined() {

return true;

}

@AnyLogicInternalCodegenAPI

public boolean isEmbeddedAgentPresentationVisible( Agent \_a ) {

return super.isEmbeddedAgentPresentationVisible( \_a );

}

@AnyLogicInternalCodegenAPI

private void \_initialize\_level\_xjal() {

level.addAll(rectangle, text, text1, text2, text3, text4, text5, text6, text7, text8, text9, text10, text11, swil\_plot, budget\_plot, chart, plot, plot1, plot2, adaptationyearplot, chart1, chart2, plot3, plot4, plot5, plot6);

}

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_chart1\_DataItem0Value() {

return

(StormSurgeBCData.containsKey(0)) ? StormSurgeBCData.get(0) : 0

;

}

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_chart1\_DataItem1Value() {

return

(StormSurgeBCData.containsKey(10)) ? StormSurgeBCData.get(10) : 0

;

}

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_chart1\_DataItem2Value() {

return

(StormSurgeBCData.containsKey(20)) ? StormSurgeBCData.get(20) : 0

;

}

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_chart1\_DataItem3Value() {

return

(StormSurgeBCData.containsKey(30)) ? StormSurgeBCData.get(30) : 0

;

}

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_chart1\_DataItem4Value() {

return

(StormSurgeBCData.containsKey(40)) ? StormSurgeBCData.get(40) : 0

;

}

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_chart1\_DataItem5Value() {

return

(StormSurgeBCData.containsKey(50)) ? StormSurgeBCData.get(50) : 0

;

}

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_chart1\_DataItem6Value() {

return

(StormSurgeBCData.containsKey(60)) ? StormSurgeBCData.get(60) : 0

;

}

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_chart1\_DataItem7Value() {

return

(StormSurgeBCData.containsKey(70)) ? StormSurgeBCData.get(70) : 0

;

}

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_chart1\_DataItem8Value() {

return

(StormSurgeBCData.containsKey(80)) ? StormSurgeBCData.get(80) : 0

;

}

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private double \_chart1\_DataItem9Value() {

return

(StormSurgeBCData.containsKey(90)) ? StormSurgeBCData.get(90) : 0

;

}

protected TimePlot swil\_plot;

protected TimePlot budget\_plot;

protected TimeStackChart chart;

protected TimePlot plot;

protected TimePlot plot1;

protected TimePlot plot2;

protected TimePlot adaptationyearplot;

protected BarChart chart1;

protected TimeStackChart chart2;

protected TimePlot plot3;

protected TimePlot plot4;

protected TimePlot plot5;

protected TimePlot plot6;

protected ShapeRectangle rectangle;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

Agency.getAdaptationMeasureName(1)

);

}

protected ShapeText text;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text1\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

Agency.getAdaptationMeasureName(2)

);

}

protected ShapeText text1;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text2\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

Agency.getAdaptationMeasureName(3)

);

}

protected ShapeText text2;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text3\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

Agency.getAdaptationMeasureName(4)

);

}

protected ShapeText text3;

protected ShapeText text4;

protected ShapeText text5;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text6\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("Resilience Index: %.2f", reliability\_index)

);

}

protected ShapeText text6;

protected ShapeText text7;

protected ShapeText text8;

protected ShapeText text9;

protected ShapeText text10;

/\*\*

\* <i>This method should not be called by user</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_text11\_SetDynamicParams\_xjal( ShapeText shape ) {

shape.setText(

String.format("Service Reliability: %.2f", level\_of\_service)

);

}

protected ShapeText text11;

protected com.anylogic.engine.markup.Level level;

private com.anylogic.engine.markup.Level[] \_getLevels\_xjal;

@Override

public com.anylogic.engine.markup.Level[] getLevels() {

return \_getLevels\_xjal;

}

@AnyLogicInternalCodegenAPI

private void \_createPersistentElementsBP0\_xjal() {

rectangle = new ShapeRectangle(

SHAPE\_DRAW\_2D3D, true,660.0, 10.0, 0.0, 0.0,

black, white,

320.0, 420.0, 10.0, 1.0, LINE\_STYLE\_SOLID );

text = new ShapeText(

SHAPE\_DRAW\_2D, true,75.0, 530.0, 0.0, 0.0,

black,"text",

\_text\_Font, ALIGNMENT\_RIGHT ) {

@Override

public void updateDynamicProperties() {

\_text\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text1 = new ShapeText(

SHAPE\_DRAW\_2D, true,75.0, 498.0, 0.0, 0.0,

black,"text",

\_text1\_Font, ALIGNMENT\_RIGHT ) {

@Override

public void updateDynamicProperties() {

\_text1\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text2 = new ShapeText(

SHAPE\_DRAW\_2D, true,75.0, 465.0, 0.0, 0.0,

black,"text",

\_text2\_Font, ALIGNMENT\_RIGHT ) {

@Override

public void updateDynamicProperties() {

\_text2\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text3 = new ShapeText(

SHAPE\_DRAW\_2D, true,75.0, 431.0, 0.0, 0.0,

black,"text",

\_text3\_Font, ALIGNMENT\_RIGHT ) {

@Override

public void updateDynamicProperties() {

\_text3\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text4 = new ShapeText(

SHAPE\_DRAW\_2D, true,750.0, 580.0, 0.0, 0.0,

black,"Extraction capacity of wells (gall/yr)",

\_text4\_Font, ALIGNMENT\_LEFT );

text5 = new ShapeText(

SHAPE\_DRAW\_2D, true,700.0, 10.0, 0.0, 0.0,

black,"Well's distance from salt water intrusion line",

\_text5\_Font, ALIGNMENT\_LEFT );

text6 = new ShapeText(

SHAPE\_DRAW\_2D, true,410.0, 380.0, 0.0, 0.0,

peru,"text",

\_text6\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text6\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

text7 = new ShapeText(

SHAPE\_DRAW\_2D, true,710.0, 780.0, 0.0, 0.0,

black,"RiskSeeker",

\_text7\_Font, ALIGNMENT\_LEFT );

text8 = new ShapeText(

SHAPE\_DRAW\_2D, true,710.0, 744.0, 0.0, 0.0,

black,"RiskNeutral",

\_text8\_Font, ALIGNMENT\_LEFT );

text9 = new ShapeText(

SHAPE\_DRAW\_2D, true,710.0, 708.0, 0.0, 0.0,

black,"RiskAverse",

\_text9\_Font, ALIGNMENT\_LEFT );

text10 = new ShapeText(

SHAPE\_DRAW\_2D, true,110.0, 670.0, 0.0, 0.0,

black,"Number of Storm Surges",

\_text10\_Font, ALIGNMENT\_LEFT );

text11 = new ShapeText(

SHAPE\_DRAW\_2D, true,410.0, 410.0, 0.0, 0.0,

peru,"text",

\_text11\_Font, ALIGNMENT\_LEFT ) {

@Override

public void updateDynamicProperties() {

\_text11\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties();

}

};

}

@AnyLogicInternalCodegenAPI

private void \_createPersistentElementsAP0\_xjal() {

{

DataSet \_item;

List<DataSet> \_items = new ArrayList<DataSet>( 1 );

\_item =

SWILDataset

;

\_items.add( \_item );

List<String> \_titles = new ArrayList<String>( 1 );

\_titles.add( "Saltwater Intrusion line" );

List<Chart2DPlot.Appearance> \_appearances = new ArrayList<Chart2DPlot.Appearance>( 1 );

\_appearances.add( new Chart2DPlot.Appearance( dodgerBlue, true, true, Chart.INTERPOLATION\_LINEAR, 2.0, Chart.POINT\_CIRCLE ) );

swil\_plot = new TimePlot(

Main.this, true, 10.0, 30.0,

330.0, 180.0,

null, null,

50.0, 10.0,

250.0, 100.0, white, black, black,

50.0, Chart.SOUTH,

100

, Chart.WINDOW\_MOVES\_WITH\_TIME, null, Chart.SCALE\_AUTO,

0, 0, Chart.GRID\_DEFAULT, Chart.GRID\_DEFAULT,

darkGray, darkGray, \_items, \_titles, \_appearances );

}

{

DataSet \_item;

List<DataSet> \_items = new ArrayList<DataSet>( 1 );

\_item =

BudgetDataset

;

\_items.add( \_item );

List<String> \_titles = new ArrayList<String>( 1 );

\_titles.add( "Budget" );

List<Chart2DPlot.Appearance> \_appearances = new ArrayList<Chart2DPlot.Appearance>( 1 );

\_appearances.add( new Chart2DPlot.Appearance( yellowGreen, true, true, Chart.INTERPOLATION\_LINEAR, 2.0, Chart.POINT\_CIRCLE ) );

budget\_plot = new TimePlot(

Main.this, true, 340.0, 30.0,

320.0, 160.0,

null, null,

50.0, 10.0,

240.0, 100.0, white, black, black,

30.0, Chart.SOUTH,

100

, Chart.WINDOW\_MOVES\_WITH\_TIME, null, Chart.SCALE\_AUTO,

0, 0, Chart.GRID\_DEFAULT, Chart.GRID\_DEFAULT,

darkGray, darkGray, \_items, \_titles, \_appearances );

}

{

DataSet \_item;

List<DataSet> \_items = new ArrayList<DataSet>( 1 );

\_item =

Supply\_over\_demand

;

\_items.add( \_item );

List<String> \_titles = new ArrayList<String>( 1 );

\_titles.add( "Level of Service (Supply/Demand)" );

List<Color> \_colors = new ArrayList<Color>( 1 );

\_colors.add( peru );

chart = new TimeStackChart(

Main.this, true, 360.0, 180.0,

300.0, 210.0,

null, null,

30.0, 30.0,

240.0, 130.0, white, black, black,

30.0, Chart.SOUTH,

100

, Chart.WINDOW\_MOVES\_WITH\_TIME, null, Chart.SCALE\_AUTO

, 0, Chart.GRID\_DEFAULT, Chart.GRID\_DEFAULT,

darkGray, darkGray, \_items, \_titles, \_colors );

}

{

DataSet \_item;

List<DataSet> \_items = new ArrayList<DataSet>( 1 );

\_items.add( \_plot\_expression0\_dataSet\_xjal );

List<String> \_titles = new ArrayList<String>( 1 );

\_titles.add( "Well South Miami Heights" );

List<Chart2DPlot.Appearance> \_appearances = new ArrayList<Chart2DPlot.Appearance>( 1 );

\_appearances.add( new Chart2DPlot.Appearance( violetRed, true, true, Chart.INTERPOLATION\_LINEAR, 2.0, Chart.POINT\_TRIANGLE ) );

plot = new TimePlot(

Main.this, true, 670.0, 30.0,

320.0, 140.0,

null, null,

50.0, 10.0,

240.0, 80.0, white, black, black,

30.0, Chart.SOUTH,

100

, Chart.WINDOW\_MOVES\_WITH\_TIME, null, Chart.SCALE\_AUTO,

0, 0, Chart.GRID\_DEFAULT, Chart.GRID\_DEFAULT,

darkGray, darkGray, \_items, \_titles, \_appearances );

}

{

DataSet \_item;

List<DataSet> \_items = new ArrayList<DataSet>( 1 );

\_items.add( \_plot1\_expression0\_dataSet\_xjal );

List<String> \_titles = new ArrayList<String>( 1 );

\_titles.add( "Well Roberta Hunter" );

List<Chart2DPlot.Appearance> \_appearances = new ArrayList<Chart2DPlot.Appearance>( 1 );

\_appearances.add( new Chart2DPlot.Appearance( coral, true, true, Chart.INTERPOLATION\_LINEAR, 2.0, Chart.POINT\_TRIANGLE ) );

plot1 = new TimePlot(

Main.this, true, 670.0, 160.0,

320.0, 140.0,

null, null,

50.0, 10.0,

240.0, 80.0, white, black, black,

30.0, Chart.SOUTH,

100

, Chart.WINDOW\_MOVES\_WITH\_TIME, null, Chart.SCALE\_AUTO,

0, 0, Chart.GRID\_DEFAULT, Chart.GRID\_DEFAULT,

darkGray, darkGray, \_items, \_titles, \_appearances );

}

{

DataSet \_item;

List<DataSet> \_items = new ArrayList<DataSet>( 1 );

\_items.add( \_plot2\_expression0\_dataSet\_xjal );

List<String> \_titles = new ArrayList<String>( 1 );

\_titles.add( "Well Former Plant" );

List<Chart2DPlot.Appearance> \_appearances = new ArrayList<Chart2DPlot.Appearance>( 1 );

\_appearances.add( new Chart2DPlot.Appearance( limeGreen, true, true, Chart.INTERPOLATION\_LINEAR, 2.0, Chart.POINT\_TRIANGLE ) );

plot2 = new TimePlot(

Main.this, true, 670.0, 270.0,

320.0, 160.0,

null, null,

50.0, 30.0,

240.0, 90.0, white, black, black,

20.0, Chart.SOUTH,

100

, Chart.WINDOW\_MOVES\_WITH\_TIME, null, Chart.SCALE\_AUTO,

0, 0, Chart.GRID\_DEFAULT, Chart.GRID\_DEFAULT,

darkGray, darkGray, \_items, \_titles, \_appearances );

}

{

DataSet \_item;

List<DataSet> \_items = new ArrayList<DataSet>( 1 );

\_item =

AdaptationYear

;

\_items.add( \_item );

List<String> \_titles = new ArrayList<String>( 1 );

\_titles.add( "Adaptation Measure" );

List<Chart2DPlot.Appearance> \_appearances = new ArrayList<Chart2DPlot.Appearance>( 1 );

\_appearances.add( new Chart2DPlot.Appearance( blue, false, true, Chart.INTERPOLATION\_LINEAR, 2.0, Chart.POINT\_CIRCLE ) );

adaptationyearplot = new TimePlot(

Main.this, true, 30.0, 390.0,

320.0, 227.0,

null, null,

50.0, 20.0,

230.0, 157.0, white, black, black,

30.0, Chart.SOUTH,

100

, Chart.WINDOW\_MOVES\_WITH\_TIME, null, Chart.SCALE\_FIXED,

0

,

5

, Chart.GRID\_DEFAULT, Chart.GRID\_NONE,

darkGray, darkGray, \_items, \_titles, \_appearances );

}

{

List<DataItem> \_items = new ArrayList<DataItem>( 10 );

\_items.add( new DataItem() {

@Override

public void update() {

setValue( \_chart1\_DataItem0Value() );

}

});

\_items.add( new DataItem() {

@Override

public void update() {

setValue( \_chart1\_DataItem1Value() );

}

});

\_items.add( new DataItem() {

@Override

public void update() {

setValue( \_chart1\_DataItem2Value() );

}

});

\_items.add( new DataItem() {

@Override

public void update() {

setValue( \_chart1\_DataItem3Value() );

}

});

\_items.add( new DataItem() {

@Override

public void update() {

setValue( \_chart1\_DataItem4Value() );

}

});

\_items.add( new DataItem() {

@Override

public void update() {

setValue( \_chart1\_DataItem5Value() );

}

});

\_items.add( new DataItem() {

@Override

public void update() {

setValue( \_chart1\_DataItem6Value() );

}

});

\_items.add( new DataItem() {

@Override

public void update() {

setValue( \_chart1\_DataItem7Value() );

}

});

\_items.add( new DataItem() {

@Override

public void update() {

setValue( \_chart1\_DataItem8Value() );

}

});

\_items.add( new DataItem() {

@Override

public void update() {

setValue( \_chart1\_DataItem9Value() );

}

});

List<String> \_titles = new ArrayList<String>( 10 );

\_titles.add( "0-9 Years" );

\_titles.add( "10-19 Years" );

\_titles.add( "20-29 Years" );

\_titles.add( "30-39 Years" );

\_titles.add( "40-49 Years" );

\_titles.add( "50-59 Years" );

\_titles.add( "60-69 Years" );

\_titles.add( "70-79 Years" );

\_titles.add( "80-89 Years" );

\_titles.add( "> 90 Years" );

List<Color> \_colors = new ArrayList<Color>( 10 );

\_colors.add( crimson );

\_colors.add( gold );

\_colors.add( yellowGreen );

\_colors.add( dodgerBlue );

\_colors.add( violetRed );

\_colors.add( coral );

\_colors.add( limeGreen );

\_colors.add( slateGray );

\_colors.add( darkMagenta );

\_colors.add( orange );

chart1 = new BarChart(

Main.this, true, 10.0, 660.0,

300.0, 250.0,

null, null,

30.0, 30.0,

260.0, 110.0, white, black, black,

100.0, Chart.SOUTH,

Chart.SOUTH, Chart.SCALE\_AUTO,

0, 0, 0.71,

Chart.GRID\_DEFAULT,

darkGray, darkGray, \_items, \_titles, \_colors );

}

{

DataSet \_item;

List<DataSet> \_items = new ArrayList<DataSet>( 3 );

\_items.add( \_chart2\_expression0\_dataSet\_xjal );

\_items.add( \_chart2\_expression1\_dataSet\_xjal );

\_items.add( \_chart2\_expression2\_dataSet\_xjal );

List<String> \_titles = new ArrayList<String>( 3 );

\_titles.add( "SouthMiamiHeights" );

\_titles.add( "RobertaHunter" );

\_titles.add( "FormerPlant" );

List<Color> \_colors = new ArrayList<Color>( 3 );

\_colors.add( violetRed );

\_colors.add( coral );

\_colors.add( limeGreen );

chart2 = new TimeStackChart(

Main.this, true, 660.0, 430.0,

320.0, 170.0,

null, black,

80.0, 20.0,

230.0, 110.0, white, black, black,

120.0, Chart.NONE,

100

, Chart.WINDOW\_MOVES\_WITH\_TIME, null, Chart.SCALE\_AUTO

, 0, Chart.GRID\_DEFAULT, Chart.GRID\_DEFAULT,

darkGray, darkGray, \_items, \_titles, \_colors );

}

{

DataSet \_item;

List<DataSet> \_items = new ArrayList<DataSet>( 1 );

\_item =

RiskUpdating

;

\_items.add( \_item );

List<String> \_titles = new ArrayList<String>( 1 );

\_titles.add( "Risk Attitude" );

List<Chart2DPlot.Appearance> \_appearances = new ArrayList<Chart2DPlot.Appearance>( 1 );

\_appearances.add( new Chart2DPlot.Appearance( brown, false, true, Chart.INTERPOLATION\_LINEAR, 1.0, Chart.POINT\_CIRCLE ) );

plot3 = new TimePlot(

Main.this, true, 730.0, 666.0,

300.0, 220.0,

null, null,

50.0, 20.0,

230.0, 130.0, white, black, black,

50.0, Chart.SOUTH,

100

, Chart.WINDOW\_MOVES\_WITH\_TIME, null, Chart.SCALE\_FIXED,

-1

,

3

, Chart.GRID\_DEFAULT, Chart.GRID\_NONE,

darkGray, darkGray, \_items, \_titles, \_appearances );

}

{

DataSet \_item;

List<DataSet> \_items = new ArrayList<DataSet>( 2 );

\_item =

year\_well\_supply

;

\_items.add( \_item );

\_item =

year\_demand\_dataset

;

\_items.add( \_item );

List<String> \_titles = new ArrayList<String>( 2 );

\_titles.add( "Year well supply (gall)" );

\_titles.add( "Year demand (gall)" );

List<Chart2DPlot.Appearance> \_appearances = new ArrayList<Chart2DPlot.Appearance>( 2 );

\_appearances.add( new Chart2DPlot.Appearance( navy, true, true, Chart.INTERPOLATION\_LINEAR, 1.0, Chart.POINT\_NONE ) );

\_appearances.add( new Chart2DPlot.Appearance( gray, true, true, Chart.INTERPOLATION\_LINEAR, 1.0, Chart.POINT\_NONE ) );

plot4 = new TimePlot(

Main.this, true, 10.0, 170.0,

340.0, 240.0,

null, null,

30.0, 30.0,

280.0, 150.0, white, black, black,

40.0, Chart.SOUTH,

100

, Chart.WINDOW\_MOVES\_WITH\_TIME, null, Chart.SCALE\_AUTO,

0, 0, Chart.GRID\_DEFAULT, Chart.GRID\_DEFAULT,

darkGray, darkGray, \_items, \_titles, \_appearances );

}

{

DataSet \_item;

List<DataSet> \_items = new ArrayList<DataSet>( 2 );

\_item =

perceived\_sr

;

\_items.add( \_item );

\_item =

actual\_sr

;

\_items.add( \_item );

List<String> \_titles = new ArrayList<String>( 2 );

\_titles.add( "Planned Service Reliability" );

\_titles.add( "Actual Service Reliability" );

List<Chart2DPlot.Appearance> \_appearances = new ArrayList<Chart2DPlot.Appearance>( 2 );

\_appearances.add( new Chart2DPlot.Appearance( red, false, true, Chart.INTERPOLATION\_LINEAR, 1.0, Chart.POINT\_TRIANGLE ) );

\_appearances.add( new Chart2DPlot.Appearance( new Color( 0x81000000, true ), false, true, Chart.INTERPOLATION\_LINEAR, 1.0, Chart.POINT\_CIRCLE ) );

plot5 = new TimePlot(

Main.this, true, 380.0, 650.0,

290.0, 240.0,

null, null,

30.0, 23.0,

240.0, 150.0, white, black, black,

40.0, Chart.SOUTH,

100

, Chart.WINDOW\_MOVES\_WITH\_TIME, null, Chart.SCALE\_FIXED,

0

,

1.1

, Chart.GRID\_DEFAULT, Chart.GRID\_DEFAULT,

darkGray, darkGray, \_items, \_titles, \_appearances );

}

{

DataSet \_item;

List<DataSet> \_items = new ArrayList<DataSet>( 3 );

\_item =

pswi\_dataset

;

\_items.add( \_item );

\_item =

aswi\_min\_dataset

;

\_items.add( \_item );

\_item =

aswi\_max\_dataset

;

\_items.add( \_item );

List<String> \_titles = new ArrayList<String>( 3 );

\_titles.add( "PSWI" );

\_titles.add( "ASWI Min" );

\_titles.add( "ASWI Max" );

List<Chart2DPlot.Appearance> \_appearances = new ArrayList<Chart2DPlot.Appearance>( 3 );

\_appearances.add( new Chart2DPlot.Appearance( gray, false, true, Chart.INTERPOLATION\_LINEAR, 1.0, Chart.POINT\_SQUARE ) );

\_appearances.add( new Chart2DPlot.Appearance( red, false, true, Chart.INTERPOLATION\_LINEAR, 1.0, Chart.POINT\_TRIANGLE ) );

\_appearances.add( new Chart2DPlot.Appearance( dodgerBlue, false, true, Chart.INTERPOLATION\_LINEAR, 1.0, Chart.POINT\_CIRCLE ) );

plot6 = new TimePlot(

Main.this, true, 350.0, 410.0,

300.0, 220.0,

null, null,

40.0, 30.0,

240.0, 140.0, white, black, black,

30.0, Chart.SOUTH,

100

, Chart.WINDOW\_MOVES\_WITH\_TIME, null, Chart.SCALE\_AUTO,

0, 0, Chart.GRID\_DEFAULT, Chart.GRID\_DEFAULT,

darkGray, darkGray, \_items, \_titles, \_appearances );

}

}

@AnyLogicInternalCodegenAPI

private void \_createPersistentElementsBS0\_xjal() {

}

// Static initialization of persistent elements

{

level = new com.anylogic.engine.markup.Level(this, "level", SHAPE\_DRAW\_2D3D, 0.0, true, true);

\_getLevels\_xjal = new com.anylogic.engine.markup.Level[] {

level };

\_createPersistentElementsBP0\_xjal();

}

protected ShapeTopLevelPresentationGroup presentation;

protected ShapeModelElementsGroup icon;

@Override

@AnyLogicInternalCodegenAPI

public ShapeTopLevelPresentationGroup getPresentationShape() {

return presentation;

}

@Override

@AnyLogicInternalCodegenAPI

public ShapeModelElementsGroup getModelElementsShape() {

return icon;

}

/\*\*

\* Constructor

\*/

public Main( Engine engine, Agent owner, AgentList<? extends Main> ownerPopulation ) {

super( engine, owner, ownerPopulation );

instantiateBaseStructureThis\_xjal();

}

@AnyLogicInternalCodegenAPI

public void onOwnerChanged\_xjal() {

super.onOwnerChanged\_xjal();

setupReferences\_xjal();

}

@AnyLogicInternalCodegenAPI

public void instantiateBaseStructure\_xjal() {

super.instantiateBaseStructure\_xjal();

instantiateBaseStructureThis\_xjal();

}

@AnyLogicInternalCodegenAPI

private void instantiateBaseStructureThis\_xjal() {

Agency = instantiate\_Agency\_xjal();

WTPSouthMiamiHeights = instantiate\_WTPSouthMiamiHeights\_xjal();

UserSouthMiamiDade = instantiate\_UserSouthMiamiDade\_xjal();

Stressor = instantiate\_Stressor\_xjal();

WellSouthMiamiHeights = instantiate\_WellSouthMiamiHeights\_xjal();

WellRobertaHunter = instantiate\_WellRobertaHunter\_xjal();

WellFormerPlant = instantiate\_WellFormerPlant\_xjal();

setupReferences\_xjal();

// Registering in Engine continuous part

getEngine().registerAgentWithEquations( this );

}

@AnyLogicInternalCodegenAPI

private void setupReferences\_xjal() {

}

/\*\*

\* Simple constructor. Please add created agent to some population by calling goToPopulation() function

\*/

public Main() {

}

/\*\*

\* Simple constructor. Please add created agent to some population by calling goToPopulation() function

\*/

public Main( double ChosenCapitalFunding, int ChosenSLRScenario, int ChosenCaseScenario, int ChosenRiskAttitude, double ChosenWellRelocationDistance, int ChosenDecisionHorizon, double ChosenPopulationGrowth, int ChosenNoAdaptation, double ChosenRiskThreshold, int WellRecoveryLimit, int RecoveryDelayMin, int RecoveryDelayMax ) {

markParametersAreSet();

this.ChosenCapitalFunding = ChosenCapitalFunding;

this.ChosenSLRScenario = ChosenSLRScenario;

this.ChosenCaseScenario = ChosenCaseScenario;

this.ChosenRiskAttitude = ChosenRiskAttitude;

this.ChosenWellRelocationDistance = ChosenWellRelocationDistance;

this.ChosenDecisionHorizon = ChosenDecisionHorizon;

this.ChosenPopulationGrowth = ChosenPopulationGrowth;

this.ChosenNoAdaptation = ChosenNoAdaptation;

this.ChosenRiskThreshold = ChosenRiskThreshold;

this.WellRecoveryLimit = WellRecoveryLimit;

this.RecoveryDelayMin = RecoveryDelayMin;

this.RecoveryDelayMax = RecoveryDelayMax;

}

/\*\*

\* Creating embedded object instances

\*/

@AnyLogicInternalCodegenAPI

private void instantiatePopulations\_xjal() {

}

@Override

@AnyLogicInternalCodegenAPI

public void doCreate() {

super.doCreate();

// Creating embedded object instances

instantiatePopulations\_xjal();

// Assigning initial values for plain variables

setupPlainVariables\_Main\_xjal();

// Dynamic initialization of persistent elements

\_createPersistentElementsAP0\_xjal();

\_initialize\_level\_xjal();

level.initialize();

presentation = new ShapeTopLevelPresentationGroup( Main.this, true, 0, 0, 0, 0 , level );

addAll( RowHeader, new String[]

{

"Adaptation ID",

"Case",

"SLR",

"Perceived SWIL",

"Perceived SS",

"Year",

"Well Name",

"Well - Distance from SWIL",

"Well - Current Extraction",

"Planned Adaptation Line"}

);

// Creating embedded object instances

instantiatePopulations\_xjal();

icon = new ShapeModelElementsGroup( Main.this, getElementProperty( "wateradaptationmodelv8.Main.icon", IElementDescriptor.MODEL\_ELEMENT\_DESCRIPTORS ) );

icon.setIconOffsets( 0.0, 0.0 );

// Environments setup

{

double \_x\_xjal =

500

;

double \_y\_xjal =

500

;

double \_z\_xjal =

0

;

setupSpace( \_x\_xjal, \_y\_xjal, \_z\_xjal );

}

disableSteps();

setNetworkUserDefined();

setLayoutType( LAYOUT\_USER\_DEFINED );

// Creating non-replicated embedded objects

setupParameters\_Agency\_xjal( Agency, null );

doBeforeCreate\_Agency\_xjal( Agency, null );

Agency.createAsEmbedded();

setupParameters\_WTPSouthMiamiHeights\_xjal( WTPSouthMiamiHeights, null );

doBeforeCreate\_WTPSouthMiamiHeights\_xjal( WTPSouthMiamiHeights, null );

WTPSouthMiamiHeights.createAsEmbedded();

setupParameters\_UserSouthMiamiDade\_xjal( UserSouthMiamiDade, null );

doBeforeCreate\_UserSouthMiamiDade\_xjal( UserSouthMiamiDade, null );

UserSouthMiamiDade.createAsEmbedded();

setupParameters\_Stressor\_xjal( Stressor, null );

doBeforeCreate\_Stressor\_xjal( Stressor, null );

Stressor.createAsEmbedded();

setupParameters\_WellSouthMiamiHeights\_xjal( WellSouthMiamiHeights, null );

doBeforeCreate\_WellSouthMiamiHeights\_xjal( WellSouthMiamiHeights, null );

WellSouthMiamiHeights.createAsEmbedded();

setupParameters\_WellRobertaHunter\_xjal( WellRobertaHunter, null );

doBeforeCreate\_WellRobertaHunter\_xjal( WellRobertaHunter, null );

WellRobertaHunter.createAsEmbedded();

setupParameters\_WellFormerPlant\_xjal( WellFormerPlant, null );

doBeforeCreate\_WellFormerPlant\_xjal( WellFormerPlant, null );

WellFormerPlant.createAsEmbedded();

// Port connectors with non-replicated objects

// Creating replicated embedded objects

setupInitialConditions\_xjal( Main.class );

// Dynamic initialization of persistent elements

\_createPersistentElementsBS0\_xjal();

}

@Override

@AnyLogicInternalCodegenAPI

public void doStart() {

super.doStart();

update\_demand\_los.start();

\_SWILDataset\_autoUpdateEvent\_xjal.start();

\_BudgetDataset\_autoUpdateEvent\_xjal.start();

\_Supply\_over\_demand\_autoUpdateEvent\_xjal.start();

\_RiskUpdating\_autoUpdateEvent\_xjal.start();

\_year\_well\_supply\_autoUpdateEvent\_xjal.start();

\_year\_demand\_dataset\_autoUpdateEvent\_xjal.start();

\_actual\_sr\_autoUpdateEvent\_xjal.start();

\_swil\_plot\_autoUpdateEvent\_xjal.start();

\_budget\_plot\_autoUpdateEvent\_xjal.start();

\_chart\_autoUpdateEvent\_xjal.start();

\_plot\_autoUpdateEvent\_xjal.start();

\_plot1\_autoUpdateEvent\_xjal.start();

\_plot2\_autoUpdateEvent\_xjal.start();

\_chart1\_autoUpdateEvent\_xjal.start();

\_chart2\_autoUpdateEvent\_xjal.start();

\_plot3\_autoUpdateEvent\_xjal.start();

\_plot4\_autoUpdateEvent\_xjal.start();

\_autoCreatedDS\_xjal.start();

Agency.startAsEmbedded();

WTPSouthMiamiHeights.startAsEmbedded();

UserSouthMiamiDade.startAsEmbedded();

Stressor.startAsEmbedded();

WellSouthMiamiHeights.startAsEmbedded();

WellRobertaHunter.startAsEmbedded();

WellFormerPlant.startAsEmbedded();

}

/\*\*

\* Assigning initial values for plain variables<br>

\* <em>This method isn't designed to be called by user and may be removed in future releases.</em>

\*/

@AnyLogicInternalCodegenAPI

public void setupPlainVariables\_xjal() {

setupPlainVariables\_Main\_xjal();

}

/\*\*

\* Assigning initial values for plain variables<br>

\* <em>This method isn't designed to be called by user and may be removed in future releases.</em>

\*/

@AnyLogicInternalCodegenAPI

private void setupPlainVariables\_Main\_xjal() {

total\_level\_of\_supply =

calculateLevelofSupply()

;

total\_demand =

UserSouthMiamiDade.getYearlyConsumption()

;

Row =

2

;

sum\_los =

supply\_over\_demand

;

}

// User API -----------------------------------------------------

@AnyLogicInternalCodegenAPI

public static LinkToAgentAnimationSettings \_connections\_commonAnimationSettings\_xjal = new LinkToAgentAnimationSettingsImpl( false, black, 1.0, LINE\_STYLE\_SOLID, ARROW\_NONE, 0.0 );

public LinkToAgentCollection<Agent, Agent> connections = new LinkToAgentStandardImpl<Agent, Agent>(this, \_connections\_commonAnimationSettings\_xjal);

@Override

public LinkToAgentCollection<? extends Agent, ? extends Agent> getLinkToAgentStandard\_xjal() {

return connections;

}

@AnyLogicInternalCodegenAPI

public void drawLinksToAgents(boolean \_underAgents\_xjal, LinkToAgentAnimator \_animator\_xjal) {

super.drawLinksToAgents(\_underAgents\_xjal, \_animator\_xjal);

if ( \_underAgents\_xjal ) {

\_animator\_xjal.drawLink( this, connections, true, true );

}

}

public List<Object> getEmbeddedObjects() {

List<Object> list = super.getEmbeddedObjects();

if (list == null) {

list = new LinkedList<Object>();

}

list.add( Agency );

list.add( WTPSouthMiamiHeights );

list.add( UserSouthMiamiDade );

list.add( Stressor );

list.add( WellSouthMiamiHeights );

list.add( WellRobertaHunter );

list.add( WellFormerPlant );

return list;

}

public AgentList<? extends Main> getPopulation() {

return (AgentList<? extends Main>) super.getPopulation();

}

public List<? extends Main> agentsInRange( double distance ) {

return (List<? extends Main>) super.agentsInRange( distance );

}

@Override

@AnyLogicInternalCodegenAPI

public boolean isLoggingToDB(EventOriginator \_e) {

if ( \_e == \_SWILDataset\_autoUpdateEvent\_xjal ) return false;

if ( \_e == \_BudgetDataset\_autoUpdateEvent\_xjal ) return false;

if ( \_e == \_Supply\_over\_demand\_autoUpdateEvent\_xjal ) return false;

if ( \_e == \_RiskUpdating\_autoUpdateEvent\_xjal ) return false;

if ( \_e == \_year\_well\_supply\_autoUpdateEvent\_xjal ) return false;

if ( \_e == \_year\_demand\_dataset\_autoUpdateEvent\_xjal ) return false;

if ( \_e == \_actual\_sr\_autoUpdateEvent\_xjal ) return false;

if ( \_e == \_swil\_plot\_autoUpdateEvent\_xjal ) return false;

if ( \_e == \_budget\_plot\_autoUpdateEvent\_xjal ) return false;

if ( \_e == \_chart\_autoUpdateEvent\_xjal ) return false;

if ( \_e == \_plot\_autoUpdateEvent\_xjal ) return false;

if ( \_e == \_plot1\_autoUpdateEvent\_xjal ) return false;

if ( \_e == \_plot2\_autoUpdateEvent\_xjal ) return false;

if ( \_e == \_chart1\_autoUpdateEvent\_xjal ) return false;

if ( \_e == \_chart2\_autoUpdateEvent\_xjal ) return false;

if ( \_e == \_plot3\_autoUpdateEvent\_xjal ) return false;

if ( \_e == \_plot4\_autoUpdateEvent\_xjal ) return false;

if ( \_e == \_autoCreatedDS\_xjal ) return false;

return super.isLoggingToDB( \_e );

}

@AnyLogicInternalCodegenAPI

public void onDestroy() {

update\_demand\_los.onDestroy();

\_SWILDataset\_autoUpdateEvent\_xjal.onDestroy();

\_BudgetDataset\_autoUpdateEvent\_xjal.onDestroy();

\_Supply\_over\_demand\_autoUpdateEvent\_xjal.onDestroy();

\_RiskUpdating\_autoUpdateEvent\_xjal.onDestroy();

\_year\_well\_supply\_autoUpdateEvent\_xjal.onDestroy();

\_year\_demand\_dataset\_autoUpdateEvent\_xjal.onDestroy();

\_actual\_sr\_autoUpdateEvent\_xjal.onDestroy();

\_swil\_plot\_autoUpdateEvent\_xjal.onDestroy();

\_budget\_plot\_autoUpdateEvent\_xjal.onDestroy();

\_chart\_autoUpdateEvent\_xjal.onDestroy();

\_plot\_autoUpdateEvent\_xjal.onDestroy();

\_plot1\_autoUpdateEvent\_xjal.onDestroy();

\_plot2\_autoUpdateEvent\_xjal.onDestroy();

\_chart1\_autoUpdateEvent\_xjal.onDestroy();

\_chart2\_autoUpdateEvent\_xjal.onDestroy();

\_plot3\_autoUpdateEvent\_xjal.onDestroy();

\_plot4\_autoUpdateEvent\_xjal.onDestroy();

\_autoCreatedDS\_xjal.onDestroy();

Agency.onDestroy();

WTPSouthMiamiHeights.onDestroy();

UserSouthMiamiDade.onDestroy();

Stressor.onDestroy();

WellSouthMiamiHeights.onDestroy();

WellRobertaHunter.onDestroy();

WellFormerPlant.onDestroy();

// Unregistering in Engine continuous part

getEngine().unregisterAgentWithEquations( this );

// Analysis Data Elements

\_ds\_current\_model\_year.destroyUpdater\_xjal();

\_ds\_reliability\_index.destroyUpdater\_xjal();

\_ds\_year\_supply.destroyUpdater\_xjal();

\_ds\_year\_demand.destroyUpdater\_xjal();

\_ds\_level\_of\_service.destroyUpdater\_xjal();

\_ds\_supply\_over\_demand.destroyUpdater\_xjal();

\_plot\_expression0\_dataSet\_xjal.destroyUpdater\_xjal();

\_plot1\_expression0\_dataSet\_xjal.destroyUpdater\_xjal();

\_plot2\_expression0\_dataSet\_xjal.destroyUpdater\_xjal();

\_chart2\_expression0\_dataSet\_xjal.destroyUpdater\_xjal();

\_chart2\_expression1\_dataSet\_xjal.destroyUpdater\_xjal();

\_chart2\_expression2\_dataSet\_xjal.destroyUpdater\_xjal();

SWILDataset.destroyUpdater\_xjal();

BudgetDataset.destroyUpdater\_xjal();

Supply\_over\_demand.destroyUpdater\_xjal();

AdaptationYear.destroyUpdater\_xjal();

RiskUpdating.destroyUpdater\_xjal();

year\_well\_supply.destroyUpdater\_xjal();

year\_demand\_dataset.destroyUpdater\_xjal();

actual\_sr.destroyUpdater\_xjal();

perceived\_sr.destroyUpdater\_xjal();

pswi\_dataset.destroyUpdater\_xjal();

aswi\_min\_dataset.destroyUpdater\_xjal();

aswi\_max\_dataset.destroyUpdater\_xjal();

logToDB( SWILDataset, "SWILDataset" );

logToDB( BudgetDataset, "BudgetDataset" );

logToDB( Supply\_over\_demand, "Supply\_over\_demand" );

logToDB( AdaptationYear, "AdaptationYear" );

logToDB( RiskUpdating, "RiskUpdating" );

logToDB( year\_well\_supply, "year\_well\_supply" );

logToDB( year\_demand\_dataset, "year\_demand\_dataset" );

logToDB( actual\_sr, "actual\_sr" );

logToDB( perceived\_sr, "perceived\_sr" );

logToDB( pswi\_dataset, "pswi\_dataset" );

logToDB( aswi\_min\_dataset, "aswi\_min\_dataset" );

logToDB( aswi\_max\_dataset, "aswi\_max\_dataset" );

logToDB( \_plot\_expression0\_dataSet\_xjal, "plot : Well South Miami Heights" );

logToDB( \_plot1\_expression0\_dataSet\_xjal, "plot1 : Well Roberta Hunter" );

logToDB( \_plot2\_expression0\_dataSet\_xjal, "plot2 : Well Former Plant" );

logToDB( \_chart2\_expression0\_dataSet\_xjal, "chart2 : SouthMiamiHeights" );

logToDB( \_chart2\_expression1\_dataSet\_xjal, "chart2 : RobertaHunter" );

logToDB( \_chart2\_expression2\_dataSet\_xjal, "chart2 : FormerPlant" );

super.onDestroy();

}

@AnyLogicInternalCodegenAPI

@Override

public void doFinish() {

super.doFinish();

Agency.doFinish();

WTPSouthMiamiHeights.doFinish();

UserSouthMiamiDade.doFinish();

Stressor.doFinish();

WellSouthMiamiHeights.doFinish();

WellRobertaHunter.doFinish();

WellFormerPlant.doFinish();

}

}