

jueves julio 18, 2013

Date_01.lkt

Page 1/1

```

1  /*****
2  * SOURCEFILE => Data_01.lkt
3  *****/
4
5  lektaProject
6
7      projectHead
8          projectDescriptor      : "Date_01"
9          projectLanguageScope   : [ es ]
10         projectOperationScope  : [ test ]
11         projectInterfaceScope  : [ dialogue ]
12         projectCompileOutput   : ".Date_01.olk"
13
14     projectSetup
15         setupParserRoots = Command
16
17     classModel
18         classDef:ElementRange ( askInfo:
19             { 'day', 'hour' } )
20         classDef:StructureComplex ( askCommand, Command : (askInfo) )
21
22     lexicalModel forLanguage es
23         ("dime el dÃ-a", askCommand, (askInfo:'day'))
24         ("dime la hora", askCommand, (askInfo:'hour'))
25
26     grammaticalModel forLanguage es
27         (R1: [ Command -> askCommand ]
28             { ^ <- #1; } )
29
30     conversationalModel
31         MindBoardStructure: {
32             (Mind / Command)
33         }
34
35         SensoSchemata
36             (SensoScheme SenseCommand : [Command]
37                 SensoAction {
38                     $MINDBOARD@Mind <- #OBJSENDO-1;
39                 }
40             )
41
42         RespondoSchemata
43             (RespondoScheme Command: [Command]
44                 RespondoAction {
45                     ^OBJRESPONDO <- $MINDBOARD@Mind;
46                 }
47             )
48
49     scriboModel forLanguage es
50         ScriboSchemata
51             (ScriboScheme Command: [Command]
52                 ScriboCapture {
53                     #OBJLOCUTIO-1 == (askInfo:'day')
54                 }
55                 ScriboAction {
56                     SetMainAnswer("Es el dÃ-a ");
57                     SetMainAnswer(ClockAskDayOfTheMonth());
58                 }
59             )
60
61             (ScriboScheme Command: [Command]
62                 ScriboCapture {
63                     #OBJLOCUTIO-1 == (askInfo:'hour')
64                 }
65                 ScriboAction {
66                     SetMainAnswer("Son las ");
67                     SetMainAnswer(ClockAskHour());
68                     SetMainAnswer(":");
69                     SetMainAnswer(ClockAskMinute());
70                 }
71             )
72

```

jueves julio 18, 2013

Date_01.slk

Page 1/1

```
1 LaunchLektaKernel()  
2  
3 UseProject (ProjectCompile      : ".Date_01.olk")  
4  
5 DisplayProcessAbstractOn  
6  
7 CreateDialogue()
```

jueves julio 18, 2013

Date_02.lkt

Page 1/2

```

1  /*****
2  * SOURCEFILE => Data_02.lkt
3  *****/
4
5  lektaProject
6
7      projectHead
8          projectDescriptor      : "Date_02"
9          projectLanguageScope   : [ es ]
10         projectOperationScope  : [ test ]
11         projectInterfaceScope  : [ dialogue ]
12         projectCompileOutput   : ".Date_02.olk"
13
14     projectSetup
15         setupParserRoots = Command
16
17     classModel
18         classDef:ElementRange ( askInfo:
19             { 'day', 'hour' } )
20         classDef:StructureComplex ( askCommand, Command : (askInfo) )
21         classDef:StructureBatch ( Commands : (Command) )
22         classDef:StructureComplex ( Task : (Commands) )
23
24     lexicalModel forLanguage es
25         ("dime el dÃ-a", askCommand, (askInfo:'day'))
26         ("dime la hora", askCommand, (askInfo:'hour'))
27
28     grammaticalModel forLanguage es
29         (R1: [ Command -> askCommand ]
30             { ^ <- #1; } )
31
32     conversationalModel
33         MindBoardStructure: {
34             (Mind / Task)
35         }
36
37         SensoSchemata
38             (SensoScheme SenseCommand : [Command]
39                 SensoAction {
40                     BatchInsertEnd($MINDBOARD@Mind.Commands, #OBJSENSO-1);
41                 }
42             )
43
44         RespondoSchemata
45             (RespondoScheme DS: [Command]
46                 RespondoCapture {
47                     (BatchSize($MINDBOARD@Mind.Commands) > 0)
48                 }
49                 RespondoAction {
50                     Command command;
51                     BatchExtractInit($MINDBOARD@Mind.Commands,command);
52                     ^OBJRESPONDO <- command;
53                     RespondoRetry();
54                 }
55             )
56
57     scriboModel forLanguage es
58         ScriboSchemata
59             (ScriboScheme Command: [Command]
60                 ScriboCapture {
61                     #OBJLOCUTIO-1 == (askInfo:'day')
62                 }
63                 ScriboAction {
64                     SetMainAnswer("Es el dÃ-a ");
65                     SetMainAnswer(ClockAskDayOfTheMonth());
66                     SetMainAnswer(". ");
67                 }
68             )
69
70             (ScriboScheme Command: [Command]
71                 ScriboCapture {
72                     #OBJLOCUTIO-1 == (askInfo:'hour')
73                 }
74                 ScriboAction {
75                     SetMainAnswer("Son las ");
76                     SetMainAnswer(ClockAskHour());
77                     SetMainAnswer(":");

```

jueves julio 18, 2013

Date_02.lkt

Page 2/2

```
78 SetMainAnswer(ClockAskMinute());
79 SetMainAnswer(". ");
80 }
81 )
82
```

jueves julio 18, 2013

Date_02.slk

Page 1/1

```
1 LaunchLektaKernel()  
2  
3 UseProject (ProjectCompile      : ".Date_02.olk")  
4  
5 DisplayProcessAbstractOn  
6  
7 CreateDialogue()
```

jueves julio 18, 2013

Date_03.lkt

Page 1/3

```

1  /*****
2  * SOURCEFILE => Data_03.lkt
3  *****/
4
5  lektaProject
6
7      projectHead
8          projectDescriptor      : "Date_03"
9          projectLanguageScope   : [ es ]
10         projectOperationScope  : [ test ]
11         projectInterfaceScope  : [ dialogue ]
12         projectCompileOutput    : ".Date_03.olk"
13
14     projectSetup
15         setupParserRoots = Command
16
17     classModel
18         classDef:ElementBool (boolean)
19         classDef:ElementInt (integer)
20         classDef:ElementLiteral (string)
21         classDef:ElementRange ( askInfo:
22             { 'date', 'day', 'hour' } )
23         classDef:StructureComplex ( askCommand, Command : (askInfo) )
24         classDef:StructureBatch ( Commands : (Command) )
25         classDef:StructureComplex ( Task : ( Commands ) )
26
27     lexicalModel forLanguage es
28         ("dime el día-a", askCommand, (askInfo:'day'))
29         ("dime la hora", askCommand, (askInfo:'hour'))
30         ("dime la fecha", askCommand, (askInfo:'date'))
31
32     grammaticalModel forLanguage es
33         (R1: [ Command -> askCommand ]
34             { ^ <- #1; } )
35
36     conversationalModel
37         MindBoardStructure: {
38             (Mind / Task)
39         }
40
41         SensoSchemata
42             (SensoScheme SenseCommand : [Command]
43                 SensoAction {
44                     BatchInsertEnd($MINDBOARD@Mind.Commands, #OBJSENSO-1);
45                 }
46             )
47
48         RespondoSchemata
49             (RespondoScheme DS: [Command]
50                 RespondoCapture {
51                     (BatchSize($MINDBOARD@Mind.Commands) > 0)
52                 }
53                 RespondoAction {
54                     Command command;
55                     BatchExtractInit($MINDBOARD@Mind.Commands,command);
56                     ^OBJRESPONDO <- command;
57                     RespondoRetry();
58                 }
59             )
60
61     functionModel
62         string monthName (integer month) {
63             switch (month) {
64                 case 1 return 'enero';
65                 case 2 return 'febrero';
66                 case 3 return 'marzo';
67                 case 4 return 'abril';
68                 case 5 return 'mayo';
69                 case 6 return 'junio';
70                 case 7 return 'julio';
71                 case 8 return 'agosto';
72                 case 9 return 'septiembre';
73                 case 10 return 'octubre';
74                 case 11 return 'noviembre';
75                 case 12 return 'diciembre';
76             }
77         }

```

jueves julio 18, 2013

Date_03.lkt

Page 2/3

```

78
79     string weekdayName (integer day) {
80         switch (day) {
81             case 1 return 'lunes';
82             case 2 return 'martes';
83             case 3 return 'miercoles';
84             case 4 return 'jueves';
85             case 5 return 'viernes';
86             case 6 return 'sabado';
87             case 7 return 'domingo';
88         }
89     }
90
91     string timeExact (integer hour, integer minute, integer second) {
92         boolean am;
93         string time;
94
95         if (hour > 12) {
96             am <- False;
97             hour -= 12;
98         } else {
99             am <- True;
100        }
101
102        if (hour == 1) {
103            time <- 'Es la 1 ';
104        } else {
105            time <- 'Son las ';
106            time <- LiteralConcat(time, ShapeToLiteral(hour));
107            time <- LiteralConcat(time, ' ');
108        }
109
110        if (am) {
111            time <- LiteralConcat(time, ' de la mañana, ');
112        } else {
113            if (hour < 10)
114                time <- LiteralConcat(time, ' de la tarde, ');
115            else
116                time <- LiteralConcat(time, ' de la noche, ');
117        }
118
119        time <- LiteralConcat(time, ShapeToLiteral(minute));
120        time <- LiteralConcat(time, ' minutos y ');
121        time <- LiteralConcat(time, ShapeToLiteral(second));
122        time <- LiteralConcat(time, ' segundos. ');
123
124        return time;
125    }
126
127 scriboModel forLanguage es
128     ScriboSchemata
129         (ScriboScheme Command: [Command]
130             ScriboCapture {
131                 #OBJLOCUTIO-1 == (askInfo: 'day')
132             }
133             ScriboAction {
134                 SetMainAnswer("Es el dÃ-a ");
135                 SetMainAnswer(ClockAskDayOfTheMonth());
136                 SetMainAnswer(". ");
137             }
138         )
139
140         (ScriboScheme Command: [Command]
141             ScriboCapture {
142                 #OBJLOCUTIO-1 == (askInfo: 'hour')
143             }
144             ScriboAction {
145                 SetMainAnswer("Son las ");
146                 SetMainAnswer(ClockAskHour());
147                 SetMainAnswer(":");
148                 SetMainAnswer(ClockAskMinute());
149                 SetMainAnswer(". ");
150             }
151         )
152
153     (ScriboScheme Command: [Command]

```

jueves julio 18, 2013

Date_03.lkt

Page 3/3

```
155      ScriboCapture {
156          #OBJLOCUTIO-1 == (askInfo:'date')
157      }
158      ScriboAction {
159          SetMainAnswer("Hoy es el ");
160          SetMainAnswer(ClockAskDayOfTheMonth());
161          SetMainAnswer(" ");
162          SetMainAnswer(monthName(ClockAskMonth()));
163          SetMainAnswer(" de ");
164          SetMainAnswer(ClockAskYear());
165          SetMainAnswer(". Es ");
166          SetMainAnswer(weekdayName(ClockAskDayOfTheWeek()));
167          SetMainAnswer(". ");
168          SetMainAnswer(timeExact(
169              ClockAskHour(),
170              ClockAskMinute(),
171              ClockAskSecond()));
172      }
173  )
174
```


jueves julio 18, 2013

Date_03.slk

Page 1/1

```
1 LaunchLektaKernel()  
2  
3 UseProject (ProjectCompile      : ".Date_03.olk")  
4  
5 DisplayProcessAbstractOn  
6  
7 CreateDialogue()
```

jueves julio 18, 2013

Devices_01.lkt

Page 1/1

```

1  /*****
2  * SOURCEFILE => Devices_01.lkt
3  *****/
4
5  lektaProject
6
7      projectHead
8          projectDescriptor      : "Devices"
9          projectLanguageScope   : [ es ]
10         projectOperationScope  : [ test ]
11         projectInterfaceScope  : [ dialogue ]
12         projectCompileOutput    : ".Devices_01.olk"
13
14     projectSetup
15         setupParserRoots = Device, Operation, Command
16         setupRuntimeSpyMessageOn
17
18     classModel
19         #Include "Devices_01_FM.lkt"
20
21     lexicalModel forLanguage es
22         #Include "Devices_01_NLU_L.lkt"
23
24     grammaticalModel forLanguage es
25         #Include "Devices_01_NLU_G.lkt"
26
27     functionModel
28         #Include "Devices_01_FUN_Decl.lkt"
29
30     conversationalModel
31         MindBoardStructure: {
32             (Input / InputModel)
33             (Control / Control)
34             (Output / OutputModel)
35         }
36
37         LaunchConversation: {
38             LaunchSendToMindBoard:
39                 (Control = (Position: 'start') )
40         }
41
42     SensoSchemata
43         #Include "Devices_01_DM_Senso.lkt"
44
45     CogitoSchemata
46         #Include "Devices_01_DM_Cogito.lkt"
47
48     RespondoSchemata
49         #Include "Devices_01_DM_Respondo.lkt"
50
51
52     functionModel
53         #Include "Devices_01_FUN_Spec.lkt"
54
55     scriboModel forLanguage es
56         ScriboSchemata
57             #Include "Devices_01_NLG.lkt"
58
59

```

jueves julio 18, 2013

Devices_01_FM.lkt

Page 1/1

```

1  /*****
2  * SOURCEFILE => Devices_01_FM.lkt
3  *****/
4
5  // AUXILIAR
6  classDef:ElementInt      (integer)
7  classDef:ElementLiteral (string)
8
9  // INPUT
10 classDef:ElementRange    (DeviceId:
11     { 'alarm', 'heating', 'irrigation' } )
12
13 classDef:ElementRange    (OperationId:
14     { 'activate', 'deactivate' } )
15
16 classDef:ElementRange    (CommandId:
17     { 'help', 'back', 'exit' } )
18
19 classDef:StructureComplex (Device: (DeviceId) )
20 classDef:StructureComplex (Operation: (OperationId) )
21 classDef:StructureComplex (Command: (CommandId) )
22
23 // OUTPUT
24 classDef:ElementRange    (AnswerCommand:
25     { 'interactivate', 'mensaje', 'outcome' })
26
27 classDef:ElementRange    (AnswerType:
28     { 'welcome', 'goodbye',
29       'askDevice', 'askHelp',
30       'askOperationAlarm', 'askOperationHeating', 'askOperationIrrigation',
31       'informAlarmOn', 'informAlarmOff', 'helpAlarm',
32       'informHeatingOn', 'informHeatingOff', 'helpHeating',
33       'informIrrigationOn', 'informIrrigationOff', 'helpIrrigation' })
34
35 classDef:StructureComplex      (OutputItem: (AnswerCommand, AnswerType) )
36
37 classDef:StructureBatch      (OutputBatch: (OutputItem) )
38
39 classDef:StructureComplex      (OutputModel: (OutputBatch) )
40
41 // MEMORY
42 classDef:StructureComplex (InputModel: (Device, Operation, Command) )
43
44 classDef:ElementRange    (Position: { 'start', 'quit',
45     'askDevice', 'askHelp',
46     'askOperationAlarm', 'askOperationHeating', 'askOperationIrrigation' } )
47
48 classDef:StructureComplex (Control: (Position) )

```

jueves julio 18, 2013

Devices_01_NLU_G.lkt

Page 1/1

```
1  /*****
2  * SOURCEFILE => Devices_01_NLU_G.lkt
3  *****/
4
5      (R1: [ Device -> DeviceId ]
6          { ^.DeviceId <- #1; } )
7
8      (R2: [ Operation -> OperationId ]
9          { ^.OperationId <- #1; } )
10
11      (R3: [ Command -> CommandId ]
12          { ^.CommandId <- #1; } )
13
```

jueves julio 18, 2013

Devices_01_NLU_L.ikt

Page 1/1

```
1  /*****
2  * SOURCEFILE => Devices_01_NLU_L.ikt
3  *****/
4
5      ("alarma",      DeviceId,    'alarm')
6      ("calefaccion", DeviceId,    'heating')
7      ("riego",       DeviceId,    'irrigation')
8
9      ("activar",     OperationId,  'activate')
10     ("desactivar",   OperationId,  'deactivate')
11
12     ("ayuda",        CommandId,    'help')
13     ("volver",       CommandId,    'back')
14     ("adios",        CommandId,    'exit')
```

jueves julio 18, 2013

Devices_01_DM_Senso.lkt

Page 1/1

```
1  /*****
2  * SOURCEFILE => Devices_01_DM_Senso.lkt
3  *****/
4
5      (SensoScheme RSetup : [Control]
6          SensoCapture {
7              #OBJSENSO-1.Position == 'start'
8          }
9          SensoAction {
10             $MINDBOARD@Control <- #OBJSENSO-1;
11         }
12     )
13
14     (SensoScheme RDis : [Device]
15         SensoAction {
16             $MINDBOARD@Input.Device <- #OBJSENSO-1;
17         }
18     )
19
20     (SensoScheme RFun : [Operation]
21         SensoAction {
22             $MINDBOARD@Input.Operation <- #OBJSENSO-1;
23         }
24     )
25
26     (SensoScheme RGen : [Command]
27         SensoAction {
28             $MINDBOARD@Input.Command <- #OBJSENSO-1;
29         }
30     )
```

jueves julio 18, 2013

Devices_01_DM_Cogito.lkt

Page 1/3

```

1  /*****
2  * SOURCEFILE => Devices_01_DM_Cogito.lkt
3  *****/
4
5      (CogitoScheme CogitoStart :
6          CogitoCapture {
7              $MINDBOARD@Control.Position == 'start'
8          }
9          CogitoAction {
10             OutputItem outputitem <- (AnswerCommand:'interactivate',AnswerType:'welcome');
11             BatchInsertEnd($MINDBOARD@Output.OutputBatch,outputitem);
12             $MINDBOARD@Control.Position <- 'askDevice';
13         }
14     )
15
16     (CogitoScheme CogitoQuit :
17         CogitoCapture {
18             $MINDBOARD@Control.Position == 'quit'
19         }
20         CogitoAction {
21             Quit();
22         }
23     )
24
25     (CogitoScheme CogitoAskDevice :
26         CogitoCapture {
27             $MINDBOARD@Control.Position == 'askDevice'
28         }
29         CogitoAction {
30             if ($MINDBOARD@Input.Device) {
31                 switch ($MINDBOARD@Input.Device.DeviceId) {
32                     case 'alarm' {
33                         $MINDBOARD@Control.Position <- 'askOperationAlarm';
34                     }
35                     case 'heating' {
36                         $MINDBOARD@Control.Position <- 'askOperationHeating';
37                     }
38                     case 'irrigation' {
39                         $MINDBOARD@Control.Position <- 'askOperationIrrigation';
40                     }
41                 }
42                 Erase($MINDBOARD@Input.Device);
43             } else {
44                 if ($MINDBOARD@Input.Command) {
45                     switch ($MINDBOARD@Input.Command.CommandId) {
46                         case 'help' {
47                             $MINDBOARD@Control.Position <- 'askHelp';
48                         }
49                         case 'exit' {
50                             $MINDBOARD@Control.Position <- 'quit';
51                             Quit();
52                         }
53                     }
54                     Erase($MINDBOARD@Input.Command);
55                 } else {
56                     Menu();
57                 }
58             }
59         }
60     )
61
62     (CogitoScheme CogitoAskHelp :
63         CogitoCapture {
64             $MINDBOARD@Control.Position == 'askHelp'
65         }
66         CogitoAction {
67             OutputItem outputitem <- (AnswerCommand:'mensaje',AnswerType:'askHelp');
68             BatchInsertEnd($MINDBOARD@Output.OutputBatch,outputitem);
69             $MINDBOARD@Control.Position <- 'askDevice';
70         }
71     )
72
73     (CogitoScheme CogitoAskOperationAlarm :
74         CogitoCapture {
75             $MINDBOARD@Control.Position == 'askOperationAlarm'
76         }
77         CogitoAction {

```

jueves julio 18, 2013

Devices_01_DM_Cogito.Ikt

Page 2/3

```

78         if ($MINDBOARD@Input.Operation) {
79             switch ($MINDBOARD@Input.Operation.OperationId) {
80                 case 'activate' {
81                     AlarmActivate();
82                     $MINDBOARD@Control.Position <- 'askDevice';
83                     Menu();
84                 }
85                 case 'deactivate' {
86                     AlarmDeactivate();
87                     $MINDBOARD@Control.Position <- 'askDevice';
88                     Menu();
89                 }
90             }
91             Erase($MINDBOARD@Input.Operation);
92         } else {
93             if ($MINDBOARD@Input.Command) {
94                 switch ($MINDBOARD@Input.Command.CommandId) {
95                     case 'help' {
96                         AlarmHelp();
97                     }
98                     case 'back' {
99                         $MINDBOARD@Control.Position <- 'askDevice';
100                         Menu();
101                     }
102                     case 'exit' {
103                         $MINDBOARD@Control.Position <- 'quit';
104                         Quit();
105                     }
106                 }
107                 Erase($MINDBOARD@Input.Command);
108             } else {
109                 OutputItem outputitem <- (AnswerCommand:'mensaje',
110                                         AnswerType:'askOperationAlarm');
111                 BatchInsertEnd($MINDBOARD@Output.OutputBatch,outputitem);
112             }
113         }
114     }
115 )
116
117 (CogitoScheme CogitoAskOperationHeating :
118     CogitoCapture {
119         $MINDBOARD@Control.Position == 'askOperationHeating'
120     }
121     CogitoAction {
122         if ($MINDBOARD@Input.Operation) {
123             switch ($MINDBOARD@Input.Operation.OperationId) {
124                 case 'activate' {
125                     HeatingActivate();
126                     $MINDBOARD@Control.Position <- 'askDevice';
127                     Menu();
128                 }
129                 case 'deactivate' {
130                     HeatingDeactivate();
131                     $MINDBOARD@Control.Position <- 'askDevice';
132                     Menu();
133                 }
134             }
135             Erase($MINDBOARD@Input.Operation);
136         } else {
137             if ($MINDBOARD@Input.Command) {
138                 switch ($MINDBOARD@Input.Command.CommandId) {
139                     case 'help' {
140                         HeatingHelp();
141                     }
142                     case 'back' {
143                         $MINDBOARD@Control.Position <- 'askDevice';
144                         Menu();
145                     }
146                     case 'exit' {
147                         $MINDBOARD@Control.Position <- 'quit';
148                         Quit();
149                     }
150                 }
151             }
152             Erase($MINDBOARD@Input.Command);
153         } else {
154             OutputItem outputitem <- (AnswerCommand:'mensaje',
155                                     AnswerType:'askOperationHeating');

```


jueves julio 18, 2013

Devices_01_DM_Cogito.lkt

Page 3/3

```

155         BatchInsertEnd($MINDBOARD@Output.OutputBatch,outputitem);
156     }
157 }
158 )
159 )
160
161 (CogitoScheme CogitoAskOperationIrrigation :
162     CogitoCapture {
163         $MINDBOARD@Control.Position == 'askOperationIrrigation'
164     }
165     CogitoAction {
166         if ($MINDBOARD@Input.Operation) {
167             switch ($MINDBOARD@Input.Operation.OperationId) {
168                 case 'activate' {
169                     IrrigationActivate();
170                     $MINDBOARD@Control.Position <- 'askDevice';
171                     Menu();
172                 }
173                 case 'deactivate' {
174                     IrrigationDeactivate();
175                     $MINDBOARD@Control.Position <- 'askDevice';
176                     Menu();
177                 }
178             }
179             Erase($MINDBOARD@Input.Operation);
180         } else {
181             if ($MINDBOARD@Input.Command) {
182                 switch ($MINDBOARD@Input.Command.CommandId) {
183                     case 'help' {
184                         IrrigationHelp();
185                     }
186                     case 'back' {
187                         $MINDBOARD@Control.Position <- 'askDevice';
188                         Menu();
189                     }
190                     case 'exit' {
191                         $MINDBOARD@Control.Position <- 'quit';
192                         Quit();
193                     }
194                 }
195                 Erase($MINDBOARD@Input.Command);
196             } else {
197                 OutputItem outputitem <- (AnswerCommand:'mensaje',
198                                         AnswerType:'askOperationIrrigation');
199                 BatchInsertEnd($MINDBOARD@Output.OutputBatch,outputitem);
200             }
201         }
202     }
203 )
204

```

jueves julio 18, 2013

Devices_01_DM_Respondo.lkt

Page 1/1

```
1  /*****
2  * SOURCEFILE => Devices_01_DM_Respondo.lkt
3  *****/
4
5      (RespondoScheme DS: [OutputItem]
6        RespondoCapture {
7          (BatchSize($MINDBOARD@Output.OutputBatch) > 0)
8        }
9        RespondoAction {
10          OutputItem outputitem;
11          BatchExtractInit($MINDBOARD@Output.OutputBatch,outputitem);
12          ^OBJRESPONDO <- outputitem;
13          RespondoRetry();
14        }
15      )
```

jueves julio 18, 2013

Devices_01_NLG.lkt

Page 1/2

```

1  /*****
2  * SOURCEFILE => Devices_01_NLG.lkt
3  *****/
4
5      (ScriboScheme GenerateWelcome: [OutputItem]
6          ScriboCapture {
7              #OBJLOCUTIO-1 == (AnswerCommand:'interactivate',AnswerType:'welcome')
8          }
9          ScriboAction {
10             SetMainAnswer("Bienvenido al sistema Devices. ");
11         }
12     )
13
14     (ScriboScheme GenerateAskDevice: [OutputItem]
15         ScriboCapture {
16             #OBJLOCUTIO-1 == (AnswerCommand:'mensaje',AnswerType:'askDevice')
17         }
18         ScriboAction {
19             SetMainAnswer("Indique que dispositivo desea controlar o bien diga ayuda. ");
20         }
21     )
22
23     (ScriboScheme GenerateAskHelp: [OutputItem]
24         ScriboCapture {
25             #OBJLOCUTIO-1 == (AnswerCommand:'mensaje',AnswerType:'askHelp')
26         }
27         ScriboAction {
28             SetMainAnswer("Desde este sistema puede controlar la alarma, la calefaccion ");
29             SetMainAnswer("y el riego. ");
30         }
31     )
32
33     (ScriboScheme GenerateGoodbye: [OutputItem]
34         ScriboCapture {
35             #OBJLOCUTIO-1 == (AnswerCommand:'interactivate',AnswerType:'goodbye')
36         }
37         ScriboAction {
38             SetMainAnswer("Adios. Muchas gracias por usar Devices. ");
39         }
40     )
41
42     (ScriboScheme GenerateOperationAlarm: [OutputItem]
43         ScriboCapture {
44             #OBJLOCUTIO-1 == (AnswerCommand:'mensaje',AnswerType:'askOperationAlarm')
45         }
46         ScriboAction {
47             SetMainAnswer("Indique si desea activar o desactivar la alarma. ");
48         }
49     )
50
51     (ScriboScheme GenerateHelpAlarm: [OutputItem]
52         ScriboCapture {
53             #OBJLOCUTIO-1 == (AnswerCommand:'mensaje',AnswerType:'helpAlarm')
54         }
55         ScriboAction {
56             SetMainAnswer("Usted puede activar o desactivar la alarma, pedir ayuda ");
57             SetMainAnswer("o volver al menu principal. ");
58         }
59     )
60
61     (ScriboScheme GenerateAlarmActivated: [OutputItem]
62         ScriboCapture {
63             #OBJLOCUTIO-1 == (AnswerCommand:'outcome',AnswerType:'informAlarmOn')
64         }
65         ScriboAction {
66             SetMainAnswer("La alarma ha sido activada. ");
67         }
68     )
69
70     (ScriboScheme GenerateAlarmDeactivated: [OutputItem]
71         ScriboCapture {
72             #OBJLOCUTIO-1 == (AnswerCommand:'outcome',AnswerType:'informAlarmOff')
73         }
74         ScriboAction {
75             SetMainAnswer("La alarma ha sido desactivada. ");
76         }
77     )

```

```

78
79      (ScriboScheme GenerationOperationHeating: [OutputItem]
80        ScriboCapture {
81          #OBJLOCUTIO-1 ::= (AnswerCommand:'mensaje',AnswerType:'askOperationHeating')
82        }
83        ScriboAction {
84          SetMainAnswer("Indique si desea activar o desactivar la calefaccion. ");
85        }
86      )
87
88      (ScriboScheme GenerateHelpHeating: [OutputItem]
89        ScriboCapture {
90          #OBJLOCUTIO-1 ::= (AnswerCommand:'mensaje',AnswerType:'helpHeating')
91        }
92        ScriboAction {
93          SetMainAnswer("Usted puede activar o desactivar la calefaccion, pedir ayuda ");
94          SetMainAnswer("o volver al menu principal. ");
95        }
96      )
97
98      (ScriboScheme GenerateHeatingActivated: [OutputItem]
99        ScriboCapture {
100          #OBJLOCUTIO-1 ::= (AnswerCommand:'outcome',AnswerType:'informHeatingOn')
101        }
102        ScriboAction {
103          SetMainAnswer("La calefaccion ha sido activada. ");
104        }
105      )
106
107      (ScriboScheme GenerateHeatingDeactivated: [OutputItem]
108        ScriboCapture {
109          #OBJLOCUTIO-1 ::= (AnswerCommand:'outcome',AnswerType:'informHeatingOff')
110        }
111        ScriboAction {
112          SetMainAnswer("La calefaccion ha sido desactivada. ");
113        }
114      )
115
116      (ScriboScheme GenerateOperationIrrigation: [OutputItem]
117        ScriboCapture {
118          #OBJLOCUTIO-1 ::= (AnswerCommand:'mensaje',AnswerType:'askOperationIrrigation')
119        }
120        ScriboAction {
121          SetMainAnswer("Indique si desea activar o desactivar el riego. ");
122        }
123      )
124
125      (ScriboScheme GenerateHelpIrrigation: [OutputItem]
126        ScriboCapture {
127          #OBJLOCUTIO-1 ::= (AnswerCommand:'mensaje',AnswerType:'helpIrrigation')
128        }
129        ScriboAction {
130          SetMainAnswer("Usted puede activar o desactivar el riego, pedir ayuda ");
131          SetMainAnswer("o volver al menu principal. ");
132        }
133      )
134
135      (ScriboScheme GenerateIrrigationActivated: [OutputItem]
136        ScriboCapture {
137          #OBJLOCUTIO-1 ::= (AnswerCommand:'outcome',AnswerType:'informIrrigationOn')
138        }
139        ScriboAction {
140          SetMainAnswer("El riego ha sido activado. ");
141        }
142      )
143
144      (ScriboScheme GenerateIrrigationDeactivated: [OutputItem]
145        ScriboCapture {
146          #OBJLOCUTIO-1 ::= (AnswerCommand:'outcome',AnswerType:'informIrrigationOff')
147        }
148        ScriboAction {
149          SetMainAnswer("El riego ha sido desactivado. ");
150        }
151      )
152

```

jueves julio 18, 2013

Devices_01_FUN_Decl.lkt

Page 1/1

```
1  /*****
2  * SOURCEFILE => Devices_01_FUN_Decl.lkt
3  *****/
4
5      integer Menu();
6      integer Quit();
7
8      integer AlarmActivate();
9      integer AlarmDeactivate();
10     integer AlarmHelp();
11
12     integer HeatingActivate();
13     integer HeatingDeactivate();
14     integer HeatingHelp();
15
16     integer IrrigationActivate();
17     integer IrrigationDeactivate();
18     integer IrrigationHelp();
```

jueves julio 18, 2013

Devices_01_FUN_Spec.lkt

Page 1/1

```

1  /*****
2  * SOURCEFILE => Devices_01_FUN_Spec.lkt
3  *****/
4
5  integer Menu() {
6      OutputItem ol <- (AnswerCommand:'mensaje',AnswerType:'askDevice');
7      BatchInsertEnd($MINDBOARD@Output.OutputBatch,ol);
8  }
9
10 integer Quit() {
11     OutputItem ol <- (AnswerCommand:'interactivate',AnswerType:'goodbye');
12     BatchInsertEnd($MINDBOARD@Output.OutputBatch,ol);
13     CloseDialogue();
14     return 1;
15 }
16
17 integer AlarmActivate() {
18     OutputItem ol <- (AnswerCommand:'outcome',AnswerType:'informAlarmOn');
19     BatchInsertEnd($MINDBOARD@Output.OutputBatch,ol);
20     return 1;
21 }
22
23 integer AlarmDeactivate() {
24     OutputItem ol <- (AnswerCommand:'outcome',AnswerType:'informAlarmOff');
25     BatchInsertEnd($MINDBOARD@Output.OutputBatch,ol);
26     return 1;
27 }
28
29 integer AlarmHelp() {
30     OutputItem ol <- (AnswerCommand:'mensaje',AnswerType:'helpAlarm');
31     BatchInsertEnd($MINDBOARD@Output.OutputBatch,ol);
32     return 1;
33 }
34
35 integer HeatingActivate() {
36     OutputItem ol <- (AnswerCommand:'outcome',AnswerType:'informHeatingOn');
37     BatchInsertEnd($MINDBOARD@Output.OutputBatch,ol);
38     return 1;
39 }
40
41 integer HeatingDeactivate() {
42     OutputItem ol <- (AnswerCommand:'outcome',AnswerType:'informHeatingOff');
43     BatchInsertEnd($MINDBOARD@Output.OutputBatch,ol);
44     return 1;
45 }
46
47 integer HeatingHelp() {
48     OutputItem ol <- (AnswerCommand:'mensaje',AnswerType:'helpHeating');
49     BatchInsertEnd($MINDBOARD@Output.OutputBatch,ol);
50     return 1;
51 }
52
53 integer IrrigationActivate() {
54     OutputItem ol <- (AnswerCommand:'outcome',AnswerType:'informIrrigationOn');
55     BatchInsertEnd($MINDBOARD@Output.OutputBatch,ol);
56     return 1;
57 }
58
59 integer IrrigationDeactivate() {
60     OutputItem ol <- (AnswerCommand:'outcome',AnswerType:'informIrrigationOff');
61     BatchInsertEnd($MINDBOARD@Output.OutputBatch,ol);
62     return 1;
63 }
64
65 integer IrrigationHelp() {
66     OutputItem ol <- (AnswerCommand:'mensaje',AnswerType:'helpIrrigation');
67     BatchInsertEnd($MINDBOARD@Output.OutputBatch,ol);
68     return 1;
69 }
70

```

jueves julio 18, 2013

Calculator_02.lkt

Page 1/3

```

1  /*****
2  * SOURCEFILE => Calculator_02.lkt
3  *****/
4
5  lektaProject
6
7  projectHead
8      projectDescriptor      : "Calculator"
9      projectLanguageScope   : [ es ]
10     projectOperationScope   : [ test ]
11     projectInterfaceScope   : [ dialogue ]
12     projectCompileOutput    : ".Calculator.olk"
13
14  projectSetup
15     setupParserRoots = Expresion
16
17  classModel
18     classDef:Void ( suma, resta, multiplica, divide, elevado )
19     classDef:Void ( abre_parentesis, cierra_parentesis )
20     classDef:ElementInt ( integer )
21
22     classDef:ElementInt ( Valor, ValorActual )
23     classDef:ElementRange ( Operador: { '+', '-', '*', '/', '^' } )
24     classDef:StructureComplex ( Expresion, Expresion1, Expresion2, ExpresionEntrada, ExpresionCa
lculo, ExpresionSalida:
25         (Valor, Operador, Expresion1, Expresion2) )
26
27     classDef:StructureComplex ( Memoria: ( ExpresionEntrada, ExpresionCalculo ) )
28
29  lexicalModel forLanguage es
30     setupTokenizerIntegerClass (integer, ".")
31     setupTokenizerPunctuation ("+", suma)
32     setupTokenizerPunctuation ("-", resta)
33     setupTokenizerPunctuation ("*", multiplica)
34     setupTokenizerPunctuation ("/", divide)
35     setupTokenizerPunctuation ("^", elevado)
36     setupTokenizerPunctuation ("(", abre_parentesis)
37     setupTokenizerPunctuation (")", cierra_parentesis)
38
39     ("cero", integer, 0)
40     ("uno", integer, 1)
41     ("dos", integer, 2)
42     ("tres", integer, 3)
43     ("cuatro", integer, 4)
44     ("cinco", integer, 5)
45     ("seis", integer, 6)
46     ("siete", integer, 7)
47     ("ocho", integer, 8)
48     ("nueve", integer, 9)
49
50     ("mas", suma)
51     ("menos", resta)
52     ("entre", divide)
53     ("por", multiplica)
54     ("elevado a", elevado)
55
56     ("abre", abre_parentesis)
57     ("abre parentesis", abre_parentesis)
58     ("cierra", cierra_parentesis)
59     ("cierra parentesis", cierra_parentesis)
60
61
62  grammaticalModel forLanguage es
63
64     (R1: [ Expresion/0 -> integer ]
65       { ^.Valor <- #1; } )
66
67     (R2: [ Expresion/0 -> &[-Expresion] resta integer ]
68       { ^.Valor <- - #2; } )
69
70     (R3: [ Expresion/0 -> abre_parentesis Expresion/10 cierra_parentesis ]
71       { ^ <- #2; } )
72
73     (R4: [ Expresion/3 -> Expresion/3 suma Expresion/2 ]
74       { ^.Operador <- '+';
75         ^.Valor <- #1.Valor + #3.Valor;
76         ^.Expresion1 <- #1;

```

jueves julio 18, 2013

Calculator_02.ikt

Page 2/3

```

77         ^.Expresion2 <- #3; } )
78
79     (R5: [ Expresion/3 -> Expresion/3 resta Expresion/2 ]
80     { ^.Operador <- '-';
81       ^.Valor <- #1.Valor - #3.Valor;
82       ^.Expresion1 <- #1;
83       ^.Expresion2 <- #3; } )
84
85     (R6: [ Expresion/2 -> Expresion/2 multiplica Expresion/1 ]
86     { ^.Operador <- '*';
87       ^.Valor <- #1.Valor * #3.Valor;
88       ^.Expresion1 <- #1;
89       ^.Expresion2 <- #3; } )
90
91     (R7: [ Expresion/2 -> Expresion/2 divide Expresion/1 ]
92     { ^.Operador <- '/';
93       ^.Valor <- #1.Valor / #3.Valor;
94       ^.Expresion1 <- #1;
95       ^.Expresion2 <- #3; } )
96
97     (R8: [ Expresion/1 -> Expresion/1 elevado Expresion/0 ]
98     { ^.Operador <- '^';
99       ^.Valor <- Pow(#1.Valor,#3.Valor);
100      ^.Expresion1 <- #1;
101      ^.Expresion2 <- #3; } )
102
103     (R9: [ Expresion/20 -> suma Expresion/10 ]
104     { ^.Operador <- '+';
105       ^.Expresion2 <- #2; } )
106
107     (R10: [ Expresion/20 -> resta Expresion/10 ]
108     { ^.Operador <- '-';
109       ^.Expresion2 <- #2; } )
110
111     (R11: [ Expresion/20 -> multiplica Expresion/10 ]
112     { ^.Operador <- '*';
113       ^.Expresion2 <- #2; } )
114
115     (R12: [ Expresion/20 -> divide Expresion/10 ]
116     { ^.Operador <- '/';
117       ^.Expresion2 <- #2; } )
118
119     (R13: [ Expresion/20 -> elevado Expresion/10 ]
120     { ^.Operador <- '^';
121       ^.Expresion2 <- #2; } )
122
123
124     conversationalModel
125     MindBoardStructure: {
126     (Memoria / Memoria)
127     }
128
129     SensoSchemata
130     (SensoScheme RExpresion : [Expresion]
131     SensoAction {
132     $MINDBOARD@Memoria.ExpresionEntrada <- #OBJSENSO-1;
133     }
134     )
135
136     CogitoSchemata
137     (CogitoScheme TExpresion :
138     CogitoAction {
139     if ($MINDBOARD@Memoria.ExpresionEntrada.Operador) {
140     if ($MINDBOARD@Memoria.ExpresionEntrada.Expresion1) {
141     $MINDBOARD@Memoria.ExpresionCalculo <-
142     $MINDBOARD@Memoria.ExpresionEntrada;
143     } else {
144     Expression fexp <- $MINDBOARD@Memoria.ExpresionEntrada;
145     fexp.Expresion1 <- $MINDBOARD@Memoria.ExpresionCalculo;
146     $MINDBOARD@Memoria.ExpresionCalculo <- fexp;
147     switch ($MINDBOARD@Memoria.ExpresionCalculo.Operador) {
148     case '+' {
149     $MINDBOARD@Memoria.ExpresionCalculo.Valor <-
150     $MINDBOARD@Memoria.ExpresionCalculo.Expresion1.Valor +
151     $MINDBOARD@Memoria.ExpresionCalculo.Expresion2.Valor;
152     }
153     case '-' {

```


jueves julio 18, 2013

Calculator_02.ikt

Page 3/3

```

154         $MINDBOARD@Memoria.ExpresionCalculo.Valor <-
155         $MINDBOARD@Memoria.ExpresionCalculo.Expresion1.Valor -
156         $MINDBOARD@Memoria.ExpresionCalculo.Expresion2.Valor;
157     }
158     case '*' {
159         $MINDBOARD@Memoria.ExpresionCalculo.Valor <-
160         $MINDBOARD@Memoria.ExpresionCalculo.Expresion1.Valor *
161         $MINDBOARD@Memoria.ExpresionCalculo.Expresion2.Valor;
162     }
163     case '/' {
164         $MINDBOARD@Memoria.ExpresionCalculo.Valor <-
165         $MINDBOARD@Memoria.ExpresionCalculo.Expresion1.Valor /
166         $MINDBOARD@Memoria.ExpresionCalculo.Expresion2.Valor;
167     }
168     case '^' {
169         $MINDBOARD@Memoria.ExpresionCalculo.Valor <-
170         Pow($MINDBOARD@Memoria.ExpresionCalculo.Expresion1.Valor,
171         $MINDBOARD@Memoria.ExpresionCalculo.Expresion2.Valor);
172     }
173 }
174 }
175 } else {
176     $MINDBOARD@Memoria.ExpresionCalculo <- $MINDBOARD@Memoria.ExpresionEntrada;
177 }
178 }
179 )
180
181 RespondoSchemata
182 (RespondoScheme DExpresion: [ExpresionSalida]
183     RespondoAction {
184         ^OBJRESPONDO <- $MINDBOARD@Memoria.ExpresionCalculo;
185     }
186 )
187
188
189 scriboModel forLanguage es
190 ScriboSchemata
191 (ScriboScheme GExpresion: [ExpresionSalida]
192     ScriboAction {
193         SetMainAnswer(#OBJLOCUTIO-1.Valor);
194     }
195 )
196

```

jueves julio 18, 2013

Calculator_02.slk

Page 1/1

```
1  /*****
2   * SOURCEFILE => Calculator_02.lkt
3   *****/
4
5  LaunchLektaKernel()
6
7  UseProject
8      (ProjectCompile      : ".Calculator.olk")
9
10 CreateDialogue()
```