

CAPÍTULO 6

EJEMPLOS

En este capítulo se presentan algunos ejemplos desarrollados como parte de esta tesis, la ejecución de los mismos se hace utilizando el modelo (MIA), el lenguaje (LIA) y el sistema (SEA) propuestos en esta tesis. Durante la ejecución de un sistema en SEA se invocan a los módulos de planeación, comparador de ontologías mixtas (COM) y el manejador de eventos inesperados (MEI). En la sección 6.1 se presenta un ejemplo en donde se tienen agentes compradores y vendedores, estos interactúan en una interacción (escenario) mercado. Para el problema del mercado se presentan tres ejecuciones, en la primera se muestra el problema utilizando ontologías donde hay equivalencia entre los conceptos que manejan los agentes y sin eventos inesperados, en la segunda se manejan dos ontologías diferentes y en la tercera se incluyen eventos inesperados. En la sección 6.2 se presenta un problema de agentes transportadores, donde cada mercancía se considera un agente y estas son capaces de moverse de un lugar de origen a uno de destino. En este caso se presenta el problema sin eventos inesperados y con ellos.

La especificación de un ambiente se hace mediante variables globales y regionales, de esta forma para un agricultor que vende su producción de maíz, es importante la producción lograda, información a la cual otros agentes compradores pueden tener acceso, el tipo de transporte disponible para llevar la mercancía a algún mercado, esto se describe en LIA como:

```
global
{
    int ProdMaiz ; // EXPRESADA EN TONELADAS
    float Precio ; // PRECIO DE GARANTIA MINIMO (POR TONELADA DE MAIZ)
}

regional
{
    char transporte[30] ;
}
```

Los agentes se describen mediante sus variables regionales, sus recursos y características expresados en las variables internas, sus propósitos, los eventos inesperados que percibe y los papeles que le permiten reaccionar ante estos. Un agente persona por ejemplo se describe:

```

agent Persona
{
    regional {
        transporte ;
    }
    internal {
        char ubicacionBodega[30] ;
        int dinero ;
    }
    purpose { // PROPOSITOS DEL AGENTE
        comprar_auto ;
    }
    percieve { // EVENTOS INESPERADOS QUE PERCIBE EL AGENTE
        lluvia, terremoto, plaga ;
    }
    action { // ACCIONES ANTE EVENTOS INESPERADOS
        fumigar, pedirAyuda ;
    }
    initial { // PAPELES INICIALES
        esperar ;
    }
}

```

Las interacciones (escenarios) contienen los papeles que toman los agentes para alcanzar sus propósitos como se ha mencionando e ilustrado en el capítulo 2.

La sección principal de un sistema descrito en LIA generalmente contiene instrucciones para crear instancias de agentes y de interacciones. Aquí se asignan propósitos y recursos en forma individual a cada agente, en el ejemplo siguiente a dos agentes persona, uno quiere tortillas y otro ganar 10 pesos.

```

main()
{
    // CREACION DE INTERACCIONES
    addinteract(Mercado,CentralAbasto);

    // CREACION DE AGENTES
    addagent(Persona, Antonio,
        tener_TORTILLA, // PROPOSITOS DEL AGENTE
        ofrece := "dinero"; quiere := "TORTILLA"; // RECURSOS DEL AGENTE
        alimento := "TORTILLA"; dinero = 100;
        ontology := "comun"); // ONTOLOGÍA UTILIZADA POR EL AGENTE

    addagent(Persona, Rafael,
        gana_10, // PROPOSITOS DEL AGENTE
        ofrece := "TORTILLA"; quiere := "10"; // RECURSOS DEL AGENTE
        recurso := "10"; dinero = 7;
        ontology := "tortillador" ); // ONTOLOGÍA UTILIZADA POR EL AGENTE
}

```

6.1 AGENTES COMPRADORES Y VENDEDORES

En este caso se tiene un (escenario) mercado en donde interactúan dos tipos de personas, cada una con diferentes propósitos, una tiene un producto y quiere venderlo para obtener dinero, mientras que otra tiene el propósito de adquirir el producto. En este caso se intercambiar mensajes entre los participantes en el mercado en donde uno es el comprador y otro el vendedor. El vendedor en este

caso toma la iniciativa y le envía un mensaje al vendedor sobre el producto que desea, el vendedor le responde el precio del artículo, el comprador verifica si el precio es aceptable para sus recursos económicos y en caso afirmativo realiza el pago y el vendedor le entrega la mercancía. Los principales mensajes intercambiados en este caso están en la figura 6.1.

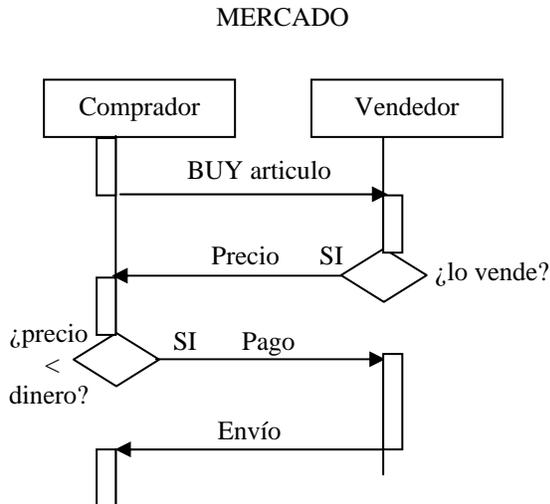


Figura 6.1 Principales mensajes que se intercambian entre un comprador y un vendedor en un mercado

El listado completo del ejemplo del mercado escrito en LIA donde se tienen un agente persona y los papeles de comprador y vendedor es el siguiente:

```

// EJEMPLO 1: Agentes compradores y vendedores
// Autor: Jesús Manuel Olivares Ceja mayo 2002
global { // VARIABLES GLOBALES
  int Reloj ;
  port pcte ;
  port pprov ;
  char mensaje[200], tmp[100] ,cual[100],
  dato[50] ;
  int duracion, lectores ;
}
regional { // VARIABLES REGIONALES
  char localidad[30] ;
}
agent Persona { // AGENTE PERSONA
  regional {
    localidad ;
  }
  internal {
    char quiere[30], ofrece[30],
    alimento[30], recurso[30] ;
    char ontology[50] ;
    float dinero ;
  }
  purpose { // PROPOSITOS DEL AGENTE
    vacaciones_cancun ;
  }
  percieve { // EVENTOS INESP. QUE PERCIBE
    perderDinero ,encontrarDinero;
  }
  action { // ACCIONES ANTE EVENTOS INESP.
    pedirAyuda ,llorar ;
  }
}
}
initial { // PAPELES INICIALES
  esperar ;
}
main()
{
  print("EJEMPLO 1: Agentes compradores y
  vendedores");

  // CREACION DE INTERACCIONES
  addinteract(Mercado,CentralAbasto);

  // CREACION DE AGENTES
  addagent(Persona, Antonio,
  tener_TORTILLA,
  ofrece := "dinero";
  quiere := "TORTILLA";
  alimento := "TORTILLA"; dinero = 100;
  ontology := "comun");

  addagent(Persona, Rafael,
  gana_10,
  ofrece := "TORTILLA"; quiere := "10";
  recurso := "10"; dinero = 7;
  ontology := "tortillador" );
}
interaction Mercado {
  role comprador(producto, recurso)
  [instances 1]
  {

```

```

requisite
{
    quiere_producto, ofrece_recurso,
    alimento_producto ;
}
positive {
    tener_producto ;
}
negative {
    ofrece_recurso ;
}
local {
    char logro[100] ;
    int resultado ;
    char id_msg[30] ;
    float costo ;
    char cantidad[30] ;
    char mensaje[200] ;
    int estado ;
}

print("Papel: comprador");
// SOLICITA UN PRODUCTO
mensaje := "( BUY ARTICULO ";
append(mensaje ,quiere);
tmp := " )";
append(mensaje ,tmp);
id_msg := "solicitud";
lectores = 2;
duracion = 60; // EN SEGUNDOS
out(pcte, id_msg, mensaje, lectores,
    duracion);
// ACEPTA LA RESPUESTA
id_msg := "respuesta";
espera = 60;
accept(pcte, id_msg, mensaje, espera);
logro := "SI";
dato := "RESPUESTA";
extract(mensaje, dato, cual);
compare(cual, logro, resultado);
dato := "PRECIO";
extract(mensaje, dato, cantidad);
costo = cantidad ;
estado = 0; // SE PREPARA SI LA RESPUESTA
// FUE 'NO' O NO LE ALCANZA EL DINERO
// VERIFICA EL PRECIO Y DETERMINA SI SE
// COMPRA
if( resultado == 1 )
{
    resultado = 0;
    if( costo <= dinero )
    {
        resultado = 1;
    }
    print("RESULTADO ");
    print(resultado);
    if( resultado == 1 )
    {
        // ACEPTA EL PRODUCTO Y CONCLUYE
        // LA COMPRA
        mensaje := "( PAY ";
        append(mensaje ,costo);
        tmp := " )";
        append(mensaje ,tmp);
        id_msg := "pago";
        lectores = 1;
        duracion = 60; // EN SEGUNDOS
        out(pcte, id_msg, mensaje,
            lectores, duracion);
        dinero = dinero - costo;
        estado = 1;
    }
}

```

```

}
if( resultado == 0 )
{
    // ACEPTA EL PRODUCTO Y CONCLUYE
    // LA COMPRA
    mensaje := "( PAY NO )";
    id_msg := "pago";
    lectores = 1;
    duracion = 90; // EN SEGUNDOS
    out(pcte, id_msg, mensaje,
        lectores, duracion);
}
}
print(estado);
if( estado == 1 )
{
    // ACEPTA LA CONFIRMACION DE ENVIO
    id_msg := "confirmacion";
    espera = 100;
    accept(pcte, id_msg, mensaje, espera);
    dato := "SEND";
    extract(mensaje, dato, cual);
    tmp := "SI";
    compare(cual, tmp, resultado);
    if( resultado == 1 )
    {
        logro := "tener_";
        append(logro ,quiere);
        goal(logro);
        print("Dinero comprador");
        print(dinero);
    }
}
print(".");
}
role vendedor (bien, producto)[instances 1] {
    requisite {
        quiere_bien, ofrece_producto
        ,recurso_bien ;
    }
    positive {
        gana_bien ;
    }
    negative {
        ofrece_producto ;
    }
    local {
        char logro[100] ;
        char id_msg[30] ;
        char dato[30], cual[50] ;
        int resultado ;
    }
    print("Papel vendedor");
    // ACEPTA EL PRODUCTO A VENDER
    id_msg := "solicitud";
    espera = 100;
    accept(pcte, id_msg, mensaje, espera);
    dato:= "ARTICULO";
    extract(mensaje, dato, cual);
    compare(cual, ofrece, resultado);
    mensaje := "( SELL ARTICULO ";
    append(mensaje ,ofrece);
    tmp := " RESPUESTA NO )";
    if( resultado == 1 )
    {
        tmp := " RESPUESTA SI PRECIO ";
    }
    append(mensaje ,tmp);
    append(mensaje ,quiere);
    tmp := " )";
    append(mensaje ,tmp);
}

```

```

id_msg := "respuesta";
lectores = 1;
duracion = 180; // EN SEGUNDOS
print(mensaje);
out(pcte, id_msg, mensaje, lectores,
    duracion);

// ACEPTA EL PAGO O EL RECHAZO AL PRECIO
id_msg := "pago";
espera = 100;
accept(pcte, id_msg, mensaje, espera);
dato:= "PAY";
extract(mensaje, dato, cual);
dato := "NO";
compare(cual, ofrece, resultado);
if( resultado == 1 )
{
    print("No se realizó la venta");
}
if( resultado == 0 )
{
    // NOTIFICA EL ENVIO DEL PRODUCTO
    mensaje := "( SEND SI )";
    id_msg := "confirmacion";
    lectores = 1;
    duracion = 180; // EN SEGUNDOS
    out(pcte, id_msg, mensaje,
        lectores, duracion);
    print(
        "ya envíe la confirmacion");
    print(mensaje);
    dinero = dinero + quiere;
    logro := "gana_10";
    goal(logro);
    print("Dinero vendedor");
    print(dinero);
}
}
}
}

```

En la figura 6.2 se muestran los eventos que se generan al ejecutar el sistema de compra-venta y donde hay agentes persona y el planificador le asigna el papel de comprador a uno y al otro le asigna el papel de vendedor. En este caso los agentes utilizan ontologías con equivalencia entre sus conceptos lo cual se nota al invocar al COM (comparador de ontologías mixtas). Es de notar que en este caso se está exento de eventos inesperados. Los mensajes que se intercambian indican el producto que desea el comprador y que en este caso el vendedor si puede surtir: (BUY ARTICULO TORTILLA) (se utilizan mayúsculas aquí para indicar conceptos). Luego se tiene la respuesta que da el vendedor, (SELL ARTICULO TORTILLA RESPUESTA SI PRECIO 10), a lo que el comprador acepta el precio en este caso realizando el pago (PAY 10.00) y finalmente el vendedor le notifica que ha enviado el producto (SEND SI). Se observa que en este caso el comprador originalmente contaba con dinero = 100 y el vendedor dinero = 7, después de la interacción el primero quedó con 90, mientras que el segundo queda con 17. Ambos agentes alcanzan sus respectivos propósitos, aunque dado que cada uno tenía un propósito (vacaciones_cancun) que no cubren quedan con solamente 50% de satisfacción cada uno y por lo tanto el grado de satisfacción del sistema también es del mismo porcentaje como se indica en las estadísticas finales.

En este ejemplo los eventos inesperados ocurren en cualquier punto de la interacción entre agentes (figura 6.3), representados con una nube de estrella. En la figura 6.4 se observa la pantalla de ejecución con los eventos inesperados. En esos puntos se utilizan papeles de emergencia para que el agente afectado supere el evento inesperado o se cambien sus propósitos. Al ocurrir en el ejemplo el evento inesperado perderDinero se afecta al agente comprador y como se observa en la bitácora resultante (figura 6.5) al quedar sin dinero: 1/01/2000 9:04:30 Antonio: Al 'perder dinero' le queda: 0.000000 ya no es posible que alcance su propósito de comprar tortillas. El código del papel que afecta a los agentes del evento inesperado es el siguiente, donde notamos que solamente los agentes con más de 50 de dinero son los afectados:

```

role perderDinero()
{
    requisite
    {
        ninguno ;
    }
    local
    {
        int resulta ;
        int cantidad ;
    }
    {
        print("Dinero inicial");
        print(dinero);
        cantidad = dinero;
        print("CANTIDAD");
        print(cantidad);
        resulta = 0;
        if( cantidad > 50 )
        {
            resulta = 1;
        }
    }
}

```

```

}
if( resulta == 1 )
{
    dinero = 0;
}
}
print("Al 'perder dinero' le queda");
print(dinero);
}
}

```

```

1/01/2000 9:00:00 EJEMPLO 1: Agentes compradores y vendedores
1/01/2000 9:00:05 Interaction CentralAbasto from Mercado created
1/01/2000 9:00:10 Agent Antonio from Persona created
1/01/2000 9:00:55 Antonio starts esperar
1/01/2000 9:00:55 Plan for Antonio
    vacaciones_cancun
        Plan not found
    Tener_TORTILLA
        0 comprador from Mercado
1/01/2000 9:00:55 Antonio starts comprador from Mercado
1/01/2000 9:00:56 Antonio: Papel: comprador
1/01/2000 9:01:00 Agent Rafael from Persona created
1/01/2000 9:01:45 Rafael starts esperar
1/01/2000 9:01:45 Plan for Rafael
    vacaciones_cancun
        Plan not found
    Gana_10
        0 vendedor from Mercado
1/01/2000 9:01:45 Rafael starts vendedor from Mercado
1/01/2000 9:01:46 Rafael: Papel: vendedor
1/01/2000 9:01:56 Antonio out solicitud ( BUY ARTICULO TORTILLA )
1/01/2000 9:02:11 COM from Antonio Rafael
1/01/2000 9:02:11 Rafael accept solicitud ( BUY ARTICULO TORTILLA )
1/01/2000 9:04:26 Rafael: ( SELL ARTICULO TORTILLA RESPUESTA SI PRECIO 10 )
1/01/2000 9:04:31 Rafael out respuesta ( SELL ARTICULO TORTILLA RESPUESTA SI PRECIO 10 )
1/01/2000 9:04:31 COM from Rafael Antonio
1/01/2000 9:04:31 Antonio accept respuesta ( SELL ARTICULO TORTILLA RESPUESTA SI PRECIO
10 )
1/01/2000 9:07:21 Antonio: RESULTADO
1/01/2000 9:09:06 Antonio out pago ( PAY 10.000000 )
1/01/2000 9:09:06 COM from Antonio Rafael
1/01/2000 9:09:06 Rafael accept pago ( PAY 10.000000 )
1/01/2000 9:11:21 Rafael out confirmacion ( SEND SI )
1/01/2000 9:11:26 Rafael: ya envíe la confirmacion
1/01/2000 9:11:31 Rafael: ( SEND SI )
1/01/2000 9:11:46 COM from Rafael Antonio
1/01/2000 9:11:46 Antonio accept confirmacion ( SEND SI )
1/01/2000 9:12:21 Rafael reached gana_10
1/01/2000 9:12:26 Rafael: Dinero vendedor
1/01/2000 9:12:31 Rafael: 17.000000
1/01/2000 9:12:36 Rafael: .
1/01/2000 9:13:01 Antonio reached tener_TORTILLA
1/01/2000 9:13:06 Antonio: Dinero comprador
1/01/2000 9:13:11 Antonio: 90.000000
1/01/2000 9:13:16 Antonio: .

```

GRADO DE SATISFACCIÓN DE LOS AGENTES CREADOS

```

-----
AGENTE      INSTANCIA          PROPÓSITOS  ALCANZADOS  % SATISFACCIÓN
Persona
    Rafael                2           1      50.00
    Antonio               2           1      50.00
ESTADISTICA DEL SISTEMA                4           2      50.00

```

Figura 6.2 Principales mensajes que se intercambian entre un comprador y un vendedor en un mercado

MERCADO

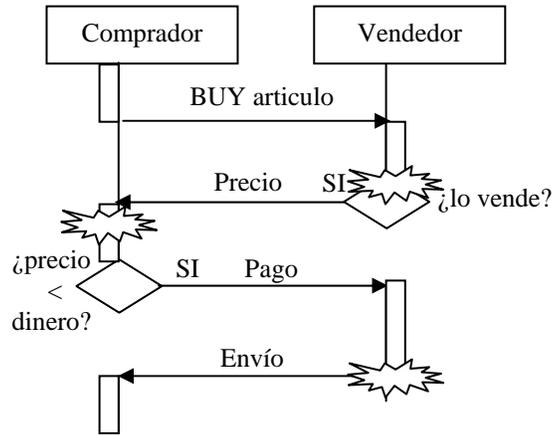


Figura 6.3 La ocurrencia de eventos inesperados sucede en cualquier punto del flujo de una interacción. La afectación es directa a un agente o indirecta cuando un agente se queda sin su interlocutor

The screenshot shows the SEA interface with the following components:

- Header:** "Ambiente de Ejecución de LIA" and a timestamp "1/01/2000 9:01:40".
- Left Panel:** A tree view with categories: GLOBAL (Reloj), REGIONAL (localidad), AGENTE (Mercancia), and INTERACCION (PlantaProduccion, Transportacion, Bodega).
- Center Log:** A list of system events such as "1/01/2000 9:00:00 PROBLEMA DE LOGISTICA" and "1/01/2000 9:01:26 Television: Cargando la mercancía".
- Right Panel:** Control buttons for "start", "stop", "step", "EXIT", and "Threads" with a "Count" set to 25.
- Unexpected Events Window (highlighted in red):**

#	Date Time	Event	DU
1	10/01/2000 11:14:00	lluvia	15
2	10/01/2000 11:14:10	terremotn	45

Below the table is a search tree for the event "lluvia":

 - evento: pasmarse.
 - naturaleza: llorar, correr.
 - lluvia: abrirParaguas, vestirGabardina.
 - terremoto: salir, llamarTelefono, orar.
 - ciudadinos: llorar, orar.

- Bottom Right:** "Planning Level: 5" and a list of items like "0 movil veracruz", "1 carga veracruz", etc.
- Footer:** "9:27:27 PM" and "20/06/03".

Figura 6.4 Pantalla de SEA con los eventos inesperados y el árbol de búsqueda del papel más específico

La atención al evento inesperado es el resultado de seleccionar un papel del árbol de eventos inesperados, en este ejemplo el papel hallado como el más específico para la atención del evento inesperado es llorar, su código LIA es:

```

role llorar()
{
  requisite
  {
    ninguno ;
  }
  local
  {
    int paso ;
  }
  print("llorando");
  paso = 0;
  while( paso < 3 )
  {
    paso = paso + 1;
    wait(1); // EN décimas de segundo
    print("... llora ");
  }
  print("Termino de llorar");
}

```

Cuando los agentes participantes utilizan ontologías diferentes por ejemplo entre un agente mecánico y un agente biologo y no cuentan con equivalencias entre sus conceptos entonces el módulo COM le es imposible encontrar la equivalencia entre los conceptos de los agentes, cuando ocurre esto, en los mensajes aparece la palabra reservada not_found en cada concepto en que no se encontró equivalencia, estos elementos los detecta entonces el agente y le permite conocer que no se logró entender con otro agente de donde resulta necesario tomar alguna acción o concluir la ejecución de su papel debido a que no logrará comunicarse, el fragmento de la ejecución en donde se observa la situación del mensaje cuando se manejan ontologías mixtas que no tienen equivalencia es:

```

1/01/2000 9:01:45 Rafael starts vendedor from Mercado
1/01/2000 9:01:46 Rafael: Papel vendedor
1/01/2000 9:01:56 Antonio out solicitud ( BUY ARTICULO TORTILLA )
1/01/2000 9:02:11 COM from Antonio Rafael
1/01/2000 9:02:11 Rafael accept solicitud ( BUY not_found not_found )
1/01/2000 9:04:21 Rafael: ( SELL ARTICULO TORTILLA RESPUESTA NO )10 )
1/01/2000 9:04:26 Rafael out respuesta ( SELL ARTICULO TORTILLA RESPUESTA NO )10 )
1/01/2000 9:04:26 COM from Rafael Antonio
1/01/2000 9:04:26 Antonio accept respuesta ( SELL not_found not_found not_found
not_found )10 )
1/01/2000 9:06:06 Antonio: 0.000000
1/01/2000 9:06:51 Antonio: .

```

En este caso los agentes participantes no alcanzan sus propósitos por no entenderse:

GRADO DE SATISFACCIÓN DE LOS AGENTES CREADOS				
AGENTE	INSTANCIA	PROPÓSITOS	ALCANZADOS	% SATISFACCIÓN
Persona	Rafael	2	0	0.00
	Antonio	2	0	0.00
	ESTADISTICA DEL SISTEMA	4	0	0.00

```

1/01/2000 9:00:00 EJEMPLO 1: Agentes compradores y vendedores
1/01/2000 9:00:05 Interaction CentralAbasto from Mercado created
1/01/2000 9:00:10 Agent Antonio from Persona created
1/01/2000 9:00:55 Antonio starts esperar
1/01/2000 9:00:55 Plan for Antonio
    vacaciones_cancun
        Plan not found
    tener_TORTILLA
        0 comprador from Mercado
1/01/2000 9:00:55 Antonio starts comprador from Mercado
1/01/2000 9:00:56 Antonio: Papel: comprador
1/01/2000 9:01:00 Agent Rafael from Persona created
1/01/2000 9:01:45 Rafael starts esperar
1/01/2000 9:01:45 Plan for Rafael
    vacaciones_cancun
        Plan not found
    gana_10
        0 vendedor from Mercado
1/01/2000 9:01:45 Rafael starts vendedor from Mercado
1/01/2000 9:01:46 Rafael: Papel: vendedor
1/01/2000 9:01:56 Antonio out solicitud ( BUY ARTICULO TORTILLA )
1/01/2000 9:02:11 COM from Antonio Rafael
1/01/2000 9:02:11 Rafael accept solicitud ( BUY ARTICULO TORTILLA )
1/01/2000 9:04:26 Rafael: ( SELL ARTICULO TORTILLA RESPUESTA SI PRECIO 10 )
1/01/2000 9:04:30 ***** perderDinero
1/01/2000 9:04:30 Antonio: Dinero inicial
1/01/2000 9:04:30 Antonio: 100.000000
1/01/2000 9:04:30 Antonio: CANTIDAD
1/01/2000 9:04:30 Antonio: 100.000000
1/01/2000 9:04:30 Antonio: Al 'perder dinero' le queda
1/01/2000 9:04:30 Antonio: 0.000000
1/01/2000 9:04:30 Rafael: Dinero inicial
1/01/2000 9:04:30 Rafael: 7.000000
1/01/2000 9:04:30 Rafael: CANTIDAD
1/01/2000 9:04:30 Rafael: 7.000000
1/01/2000 9:04:35 Rafael: llorando
1/01/2000 9:04:35 Antonio: llorando
1/01/2000 9:04:45 *end* perderDinero
1/01/2000 9:04:31 Rafael out respuesta ( SELL ARTICULO TORTILLA RESPUESTA SI PRECIO 10 )
1/01/2000 9:04:31 COM from Rafael Antonio
1/01/2000 9:04:31 Antonio accept respuesta ( SELL ARTICULO TORTILLA RESPUESTA SI PRECIO
10 )
1/01/2000 9:06:31 Antonio: ... llora
1/01/2000 9:06:31 Rafael: ... llora
1/01/2000 9:07:06 Antonio: 0.000000
1/01/2000 9:08:17 Rafael: ... llora
1/01/2000 9:08:17 Antonio: ... llora
1/01/2000 9:09:11 Antonio out pago ( PAY NO )
1/01/2000 9:09:11 COM from Antonio Rafael
1/01/2000 9:09:11 Rafael accept pago ( PAY NO )
1/01/2000 9:10:03 Antonio: ... llora
1/01/2000 9:10:03 Rafael: ... llora
1/01/2000 9:10:16 Rafael: No se realizó la venta
1/01/2000 9:11:03 Antonio: Termino de llorar
1/01/2000 9:11:03 Rafael: Termino de llorar
1/01/2000 9:11:15 Antonio: termina la espera
1/01/2000 9:12:05 Rafael: termina la espera

          GRADO DE SATISFACCIÓN DE LOS AGENTES CREADOS
          -----
AGENTE      INSTANCIA          PROPÓSITOS  ALCANZADOS  % SATISFACCIÓN
Persona
    Rafael                2            0            0.00
    Antonio                2            0            0.00
ESTADISTICA DEL SISTEMA                4            0            0.00

```

Figura 6.5 Bitacora ante la ocurrencia del evento inesperado perderDinero

6.2 AGENTES TRANSPORTADORES

En esta sección se tiene mercancía que se espera llegue a un destino específico. Para evitar que un agente externo se encargue de estar validando el destino de cada producto, en este caso cada agente es una mercancía. Las mercancías participarán en diferentes interacciones para cargarse en el transporte adecuado, luego esperarán a llegar a su destino y ahí se descargarán.

En el proceso principal se crean tres agentes, dos con destino a Puebla y uno a Tijuana.

Las interacciones son Almacén, en donde se puede cargar mercancía. Transportadora, donde se realiza el movimiento de la mercancía y el Destinatario en donde se descarga la mercancía.

El listado en LIA del sistema en donde se tiene este problema se indica a continuación.

```

global
{
  int Reloj ;
}
regional
{
  float localidad ;
}
agent Mercancia
{
  regional
  {
    localidad ;
  }
  internal
  {
    char carga[30],
        movil[30],
        transporte[30] ,
        origen[30] ,
        destino[30],
        nuevologar[30],
        soporte[30] ;
  }
  purpose // PROPOSITOS DEL AGENTE
  {
    evitar_romperse ;
  }
  percieve // EVENTOS INESP.
  {
    lluvia, terremoto, caidas ;
  }
  action // ACCIONES ANTE EVENTOS INESP.
  {
    pedirAyuda ;
  }
  initial // PAPELES INICIALES
  {
    esperar ;
  }
}

main()
{
  print(
    "EJEMPLO 2: PROBLEMA DE TRANSPORTACION");
  addinteract(Almacen ,centralAbasto );
  addagent(Mercancia ,Television,
    carga_puebla,
    carga := "veracruz";
    movil := "veracruz";
    transporte := "camion";
    soporte := "camion";
    nuevologar := "puebla";
    origen:="veracruz";
    destino := "puebla";
    money=1000;
    ontology := "electrodomestico" );
  addagent(Mercancia , Radio,
    carga_tijuana,
    carga := "veracruz";
    movil := "veracruz";
    transporte := "camion";
    soporte := "camion";
    nuevologar := "tijuana";
    origen:="veracruz";
    destino := "tijuana";
    money=10000;
    ontology := "radios" );
  addagent(Mercancia , Refrigerador,
    carga_puebla,
    carga := "veracruz";
    movil := "veracruz";
    transporte := "camion";
    soporte := "camion";
    nuevologar := "puebla";
    origen:="veracruz";
    destino := "puebla";
    money=1000;
    ontology := "blancos" );
}
interaction Almacen
{
  role cargar (lugar, transporte)[instances 1]
  {
    requisite
    {
      movil_lugar, carga_lugar,
      soporte_transporte ;
    }
  }
}

```



```

1/01/2000 9:00:00 EJEMPLO 2: PROBLEMA DE TRANSPORTACIÓN
1/01/2000 9:00:05 Interaction plantaA from PlantaProduccion created
1/01/2000 9:00:10 Interaction TransportesPatito from Transportacion created
1/01/2000 9:00:15 Interaction CentralAbasto from Bodega created
1/01/2000 9:00:20 Agent Television from Mercancia created
1/01/2000 9:01:25 Television starts esperar
1/01/2000 9:01:25 Plan for Television
    evitar_romperse
        Plan not found
    carga_puebla
        2 descargar from Bodega
        1 mover from Transportacion
        0 cargar from PlantaProduccion
1/01/2000 9:01:25 Television starts cargar from PlantaProduccion
1/01/2000 9:01:26 Television: Cargando la mercancía
1/01/2000 9:01:30 Agent Radio from Mercancia created
1/01/2000 9:01:41 Television reached carga_camion
1/01/2000 9:01:46 starts mover from Transportacion
1/01/2000 9:02:12 Television: Moviendo desde veracruz hacia puebla
1/01/2000 9:02:27 Television reached movil_puebla
1/01/2000 9:02:32 starts descargar from Bodega
1/01/2000 9:02:33 Television: Descargando el camión
1/01/2000 9:02:35 Radio starts esperar
1/01/2000 9:02:35 Plan for Radio
    evitar_romperse
        Plan not found
    carga_tijuana
        2 descargar from Bodega
        1 mover from Transportacion
        0 cargar from PlantaProduccion
1/01/2000 9:02:35 Radio starts cargar from PlantaProduccion
1/01/2000 9:02:36 Radio: Cargando la mercancía
1/01/2000 9:02:40 Agent Refrigerador from Mercancia created
1/01/2000 9:02:48 Television reached carga_puebla
1/01/2000 9:02:51 Radio reached carga_camion
1/01/2000 9:02:56 starts mover from Transportacion
1/01/2000 9:03:22 Radio: Moviendo desde veracruz hacia tijuana
1/01/2000 9:03:37 Radio reached movil_tijuana
1/01/2000 9:03:42 starts descargar from Bodega
1/01/2000 9:03:43 Radio: Descargando el camión
1/01/2000 9:03:45 Refrigerador starts esperar
1/01/2000 9:03:45 Plan for Refrigerador
    evitar_romperse
        Plan not found
    carga_puebla
        2 descargar from Bodega
        1 mover from Transportacion
        0 cargar from PlantaProduccion
1/01/2000 9:03:58 Radio reached carga_tijuana
1/01/2000 9:11:45 Television: termina la espera
1/01/2000 9:12:55 Radio: termina la espera
1/01/2000 9:14:05 Refrigerador: termina la espera

                GRADO DE SATISFACCIÓN DE LOS AGENTES CREADOS
                -----
AGENTE      INSTANCIA          PROPÓSITOS  ALCANZADOS  % SATISFACCIÓN
Mercancia
    Refrigerador          2           0           0.00
    Radio                 2           1           50.00
    Television            2           1           50.00
ESTADISTICA DEL SISTEMA          6           2           33.33

```

Figura 6.6 Bitacora del problema de transportación donde las mercancías son agentes