

Especificación formal de un sistema de información temporalizado

María Lourdes Montes López
Facultad de Electrotecnia y Computación
Universidad Nacional de Ingeniería, Nicaragua
montes@datatex.com.ni

Ignacio Trejos Zelaya
Computación, Instituto Tecnológico de Costa Rica, y
Cenfotec
itrejos@cenfotec.com, itrejos@itcr.ac.cr

Resumen

En este artículo aplicamos una técnica para la especificación de sistemas de información temporalizados sobre un sistema de reservaciones de una aerolínea. La técnica, TempER-Z, usa de manera prominente una biblioteca de esquemas genéricos útiles para describir los aspectos temporales de un sistema de información transaccional.

En la especificación del sistema de reservaciones se describe cómo reservar un recurso y programarlo. No se almacenan fechas y horas ni se hace uso de operaciones para controlar el conflicto en asignaciones de aviones a vuelos programados; en lugar de eso se emplean esquemas genéricos para asociaciones con reserva de tiempo, que describen la reservación de un recurso y su programación en el tiempo.

Palabras claves: especificación formal, sistema de información temporalizado, modelo conceptual.

1. Introducción

En [Montes 2000, Montes 2000b] se presenta una técnica, TempER-Z, que facilita la especificación de un modelo entidad-asociación con características temporales. Las propiedades estáticas de un sistema de información temporalizado se describen mediante un modelo entidad-asociación temporalizado [Antunes 1996, Antunes 1997a] y una biblioteca de esquemas genéricos que permite describir tanto los aspectos estáticos como los dinámicos. La biblioteca contiene patrones para especificar entidades, atributos, asociaciones y operaciones básicas para el modelaje de sistemas de información temporalizados. La biblioteca y las restricciones adicionales sobre los sistemas están escritos en la notación Z [Spivey 1992, Trejos 1995, Wordsworth 1992].

En este artículo se aplican los esquemas genéricos de la biblioteca de TempER-Z sobre un sistema de reservaciones de una aerolínea que tiene características temporales.

Este artículo está organizado así: la sección 2 presenta la descripción de los requerimientos (simplificados) de un sistema de reservación de una aerolínea; la sección 3 presenta el diagrama ER temporalizado, su diccionario de datos y su descripción; la sección 4 presenta la especificación del sistema con la técnica TempER-Z; la sección 5 presenta las conclusiones.

2. Requerimientos

Una compañía de transporte aéreo se encarga de transportar pasajeros en las rutas predeterminadas para sus vuelos. Los pasajeros pueden reservar cupo en un vuelo para viajar en una fecha específica. Para reservar, la compañía les solicita una identificación (usualmente un pasaporte), posteriormente les asigna un número de asiento, el número de vuelo y les comunica la hora de salida. Un pasajero puede reservar como máximo un asiento por vuelo.

La compañía tiene disponible una flota de aviones sobre la cual realiza la programación de los vuelos por ejecutar en las rutas establecidas, en conformidad con la cantidad máxima de pasajeros, fecha de vuelo, hora de salida, país de origen, país de destino y días de la semana en que se realiza el vuelo. Los vuelos tienen una tarifa estipulada según la ruta, la cual está sujeta a modificación antes de iniciarse el viaje. Los vuelos son caracterizados por tres estados: *reserva* es el estado inicial de un vuelo programado y se mantiene mientras se encuentra en periodo de reservaciones; *realizado* corresponde a un vuelo ejecutado y *cancelado* corresponde a un vuelo que por criterio de la compañía no se realiza.

Se supone que los pasajeros ya están registrados en el sistema de reservación como resultado de la compra previa del boleto. La información que se registra de cada pasajero es el identificador del pasajero (número de pasaporte), el nombre y el número de teléfono.

3. Modelo entidad-asociación temporalizado

A continuación presentamos el modelo entidad-asociación temporalizado [Antunes 1996, Montes 2000]. Las cardinalidades hacen referencia a la participación de una entidad en una asociación.

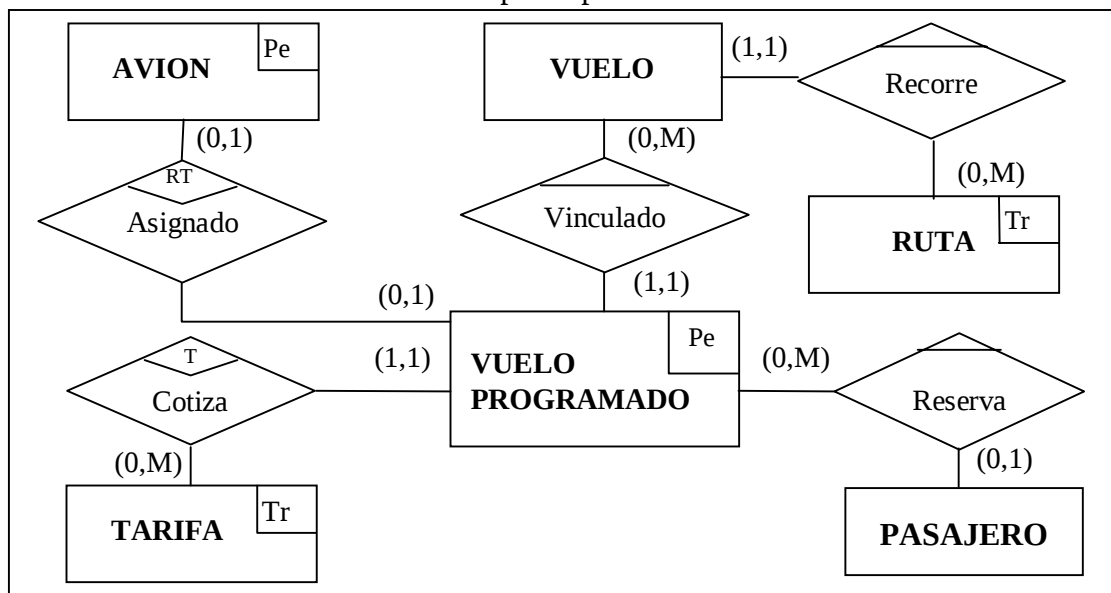


Figura No. 1: Modelo entidad-asociación temporalizado del sistema de reservaciones.

La entidad *tarifa* es de tipo transitoria porque es posible que una tarifa pueda tomar otro valor en el transcurso del tiempo, lo que indica que la existencia de la entidad tarifa puede estar compuesta por más de un intervalo de tiempo y su validez temporal no necesariamente está formada por momentos consecutivos.

La entidad *ruta* es de tipo transitoria porque es posible que la trayectoria y distancia recorrida por una ruta puedan ser redefinidas en el transcurso del tiempo, lo que indica que la existencia de la entidad ruta puede estar compuesta por más de un intervalo de tiempo y su validez temporal no necesariamente está formada por momentos consecutivos.

Las entidades *vuelo* y *pasajero* son intemporales porque no es necesario conocer los diferentes valores que toman los atributos en el transcurso del tiempo y por consiguiente no toman en consideración la dimensión temporal.

Las entidades *avión* y *vuelo_programado* son perennes porque, aunque no contienen atributos temporales, se consideran válidas en toda la dimensión temporal. Esto es necesario para conocer los puntos de tiempo en que ambas instancias se asocian, es decir aquellas en que un avión está asignado a un vuelo programado.

Los atributos *trayecto*, *distancia* y *monto* se definen como temporales porque presentan un valor en cada momento de tiempo. El trayecto de una ruta puede ser rediseñado en un momento específico de tiempo y como la distancia depende del trayecto también tomará valores en cada momento de tiempo.

en que se modifique el trayecto. Lo mismo sucede con la tarifa, que puede ser revalorada, por lo que tomará diferentes valores en momentos específicos de tiempo.

La asociación *cotiza* se define de tipo temporal para especificar que una tarifa puede ser cotizada por varios vuelos programados en cada momento de tiempo de su existencia y que un vuelo puede ser cotizado por una tarifa única en cada momento de tiempo de su existencia. La asociación *asignado* se define de tipo temporal con reserva de tiempo porque es necesario conocer los momentos específicos de tiempo en que un avión es asignado a un vuelo programado. Las asociaciones *vinculado*, *recorre* y *reserva* se definen de tipo intemporal, porque no es necesario conocer los momentos específicos de tiempo en que las instancias de las entidades se asocian.

3.1. Diccionario de datos

Entidades

Entidad	Atributos			Atributo_llave
	atributos	tipo dato	clasificación temporal	
AVIÓN	id_avión	carácter	intemporal	id_avión
	capacidad	numérico	intemporal	
PASAJERO	id_pasajero	carácter	intemporal	id_pasajero
	nombre	hilera	intemporal	
	teléfono	carácter	intemporal	
RUTA	id_ruta	carácter	intemporal	id_ruta
	distancia	numérico	temporal	
	trayecto	hilera	temporal	
TARIFA	id_tarifa	carácter	intemporal	id_tarifa
	monto	dinero	temporal	
VUELO	id_vuelo	carácter	intemporal	id_vuelo
	origen	carácter	intemporal	
	destino	carácter	intemporal	
	días_vuelo	días	intemporal	
	hora_salida	hora	intemporal	
VUELO_PROGRAMADO	id_vuelopro	carácter	intemporal	id_vuelopro
	estado_vp	estado_vp	intemporal	
	hora_salida	hora	intemporal	
	fecha_salida	fecha	intemporal	
	ocupados	numérico	intemporal	
	libres	numérico	intemporal	

Cuadro No.1: Descripción de entidades del modelo entidad-asociación temporalizado

Asociaciones

Asociación	Atributos		
	atributos	tipo dato	Clasificación temporal
RESERVA	no_asiento	Numérico	intemporal

Cuadro No.2: Descripción de asociaciones del modelo entidad-asociación temporalizado

3.2. Descripción del modelo entidad- asociación temporalizado

AVION (asignado) VUELO_PROGRAMADO: Especifica que un avión puede no estar asignado a un vuelo programado o puede estar asignado a lo sumo un vuelo programado a la vez durante su

existencia. Por otra parte, un vuelo programado puede no tener un avión asignado o puede tener asignado a lo sumo un avión en cada momento de tiempo de su existencia.

VUELO (recorre) RUTA: Especifica que un vuelo obligatoriamente recorre una ruta única en su trayectoria. Por otra parte, una ruta puede ser recorrida por cero o más vuelos.

VUELO_PROGRAMADO (cotiza) TARIFA: Especifica que un vuelo programado obligatoriamente debe ser cotizado con una tarifa en cada momento de tiempo de su existencia. Por otra parte, una tarifa puede no ser cotizada o puede ser cotizada por varios vuelos programados en cada momento de tiempo.

VUELO_PROGRAMADO (reserva) PASAJEROS: Especifica que un vuelo programado puede no ser reservado por un pasajero o puede ser reservado por varios pasajeros. Por otra parte, un pasajero reserva a lo sumo un asiento por vuelo programado.

VUELO_PROGRAMADO (vinculado) VUELO: Especifica que un vuelo programado obligatoriamente debe estar vinculado a un vuelo. Por otra parte, un vuelo puede no estar vinculado a un vuelo programado o puede estar vinculado a varios.

4. Especificación

En esta sección se presenta la especificación del sistema con la biblioteca de especificación desarrollada en [Montes 2000b]. En primer lugar se define el diccionario de datos, el modelaje de las entidades, sus instancias y asociaciones. Posteriormente se presenta el modelaje del estado total y el inicial del sistema. Por último, para fines ilustrativos de la aplicación de la biblioteca, se presenta el modelaje de algunas de las operaciones del sistema, con las situaciones de error que se pueden dar para cada una de ellas y su operación total (práctica común al especificar en Z [Wordsworth 1992]).

4.1. Diccionario de datos

[Id_Avión, Id_VueloPro, Id_Pasajero]	
Conjuntos dados de identificadores de avión, vuelo programado y pasajero.	
[Id_Tarifa]	Conjunto dado de identificadores de tarifa.
[Id_Vuelo]	Identificadores de vuelos definidos en el catálogo.
[Id_Ruta]	Identificadores de ruta.
[País]	Países, utilizados para describir el país de origen y el país de destino.
[Trayecto]	El sistema maneja información sobre la descripción del trayecto que se recorre para cada ruta.
[Nombre, Teléfono]	Nombres y teléfonos de pasajeros.
[Días]	Días de la semana.
[Fecha]	Fechas de vuelos programados.
[Hora]	Tiempo medido en horas, minutos y segundos.
Hilera == Seq ₁ Char	Secuencias de caracteres.
Númérico == $\mathbb{N}$	Números enteros.
Distancia == $\mathbb{N}$	Distancias entre países.
Dinero == $\mathbb{N}$	Monto o representación monetaria.
Capacidad == $\mathbb{N}$	Cantidad de pasajeros que pueden reservar o viajar en un vuelo.
Estado_Vp := Reserva Realizado Cancelado	
Representa el estado en que puede estar un vuelo: en <i>reserva</i> , <i>realizado</i> o <i>cancelado</i> .	
Respuesta := Seq ₁ Char	
Para generar la respuesta de la ejecución de operaciones.	

4.2. Entidades/Instancias

A continuación se presentan algunas de las entidades/instancias que caracterizan cada tipo:

AVIÓN: Es el medio físico mediante el cual se realizan los vuelos. El identificador se declara como un conjunto dado. El atributo restante se incluye en el esquema de información de la entidad.

Info_Avión ———
capacidad: Capacidad

Las instancias de la entidad *avión*, de tipo perenne, se definen de la siguiente manera:

Insta_Avión_Pe $\hat{=}$ Insta_Entidad_Pe [Id_Avión, Info_Avión]
[ident_avión / ident_entidad, insta_avión/ insta_entidad,
activa_avión / activación, vigen_avión / vigencia, estado_avión / estado]

VUELO: Es un catálogo que contiene la clasificación de los vuelos que ofrece la compañía. El identificador de esta entidad se declara como un conjunto dado. Los atributos descriptivos se incluyen en el esquema de información de la entidad.

Info_Vuelo ———
origen: País
destino: País
días_vuelo: Días
hora_salida: Hora

Las instancias de la entidad *vuelo*, de tipo intemporal, se definen de la siguiente manera:

Insta_Vuelo_In $\hat{=}$ Insta_Entidad_In [Id_Vuelo, Info_Vuelo]
[ident_vuelo / ident_entidad, insta_vuelo/ insta_entidad]

ruta: Es un catálogo que contiene la información de las diferentes trayectorias que pueden seguir los vuelos que ofrece la compañía. El identificador se declara como un conjunto dado. Los atributos descriptivos se incluyen en el esquema de información de la entidad.

Info_Ruta ———
distancia: Temporalizado [Numérico]
trayecto: Temporalizado [Hilera]

Los atributos *distancia* y *trayecto* se definen como temporalizados porque pueden presentar valores en momentos de tiempo no necesariamente continuos. Cada vez que se modifique el trayecto de la ruta, la distancia tomará un nuevo valor. Para definir los atributos *distancia* y *trayecto* como temporalizados se hace uso del esquema genérico de la biblioteca *Temporalizado[X]*.

Las instancias de la entidad *ruta*, de tipo transitoria, se definen de la siguiente manera:

Insta_Ruta_Tr $\hat{=}$ Insta_Entidad_Tr [Id_Ruta, Info_Ruta]
[ident_ruta / ident_entidad,
insta_ruta / insta_entidad,
activa_ruta / activación,
vigen_ruta / vigencia,
estado_ruta / estado]

4.3. Asociaciones

ASOCITEMP_ASIGNADO: Mantiene información de los vuelos asignados a cada uno de los aviones de la compañía. Es una asociación no temporalizada de tipo 1 a 1. Es parcial en el dominio y total en el codominio, pueden existir aviones sin vuelos asignados y por otra parte un avión puede ser asignado a varios vuelos.

AsociTemp_Asignado $\cong$ AsociTemp_Reserva_01_11 [Id_Avi3n, Id_VueloPro]
 [ident_avion / ident_entidad1,
 ident_vueloprogramado/ ident_entidad2,
 asoci_asignado/ asoci_entidad1_entidad2]

ASOCITEMP_COTIZA: Mantiene informaci3n de las diferentes tarifas en que pueden cotizarse los vuelos programados. Es temporalizada de tipo 1 a muchos. La asociaci3n es total en el dominio y parcial en el codominio, ya que en un momento de tiempo un vuelo programado obligatoriamente debe ser cotizado con una tarifa 3nica. Por otra parte, una tarifa puede no ser cotizada en un momento o puede ser cotizada por varios vuelos programados.

AsociTemp_Cotiza $\cong$ AsociTemp_11_0M [Id_VueloPro, Id_Tarifa]
 [ident_vueloprogramado/ ident_entidad1,
 ident_tarifa / ident_entidad2,
 vigente_desde_cotiza / vigente_desde, vigencia_cotiza / vigencia,
 asociemp_cotiza/ asociemp_entidad1_entidad2]

ASOCI_RESERVA: Mantiene informaci3n de las reservaciones realizadas por los pasajeros en un vuelo programado. Es una asociaci3n no temporalizada de tipo 1 a muchos. La asociaci3n es parcial en ambos lados, ya que un vuelo programado puede ser reservado por varios pasajeros y por otra parte un pasajero puede reservar a lo m3nimo un asiento en un vuelo. Posee el atributo n3mero de asiento.

Info_Asoci_Reserva
 no_asiento: No_Asiiento

La asociaci3n queda definida por:

Asoci_Reserva $\cong$ Asoci_01_0M [Id_Pasajero, Id_VueloPro]
 [ident_pasajero / ident_entidad1,
 ident_vuelopro / ident_entidad2,
 asoci_reserva / asoci_entidad1_entidad2]
 $\wedge$ Insta_Asoci_Con_Atrib[Id_Pasajero, Id_VueloPro, Info_Asoci_Reserva]
 [ident_pasajero / ident_entidad1,
 ident_vuelopro / ident_entidad2,
 asoci_reserva / asoci_entidad1_entidad2,
 insta_reserva / insta_entidad1_entidad1]

ASOCI_VINCULADO: Mantiene informaci3n de las caracter3sticas de los vuelos programados. Es temporalizada de tipo 1 a muchos. Un vuelo puede estar vinculado a varios vuelos programados. La asociaci3n es total en el dominio y parcial en el codominio, ya que un vuelo programado obligatoriamente debe estar vinculado a un vuelo y por otra parte un vuelo puede estar vinculado a varios vuelos programados.

Asoci_Vinculado $\cong$ Asoci_11_0M [Id_VueloPro, Id_Vuelo]
 [ident_vuelopro / ident_entidad1,
 ident_vuelo/ ident_entidad2,
 asoci_vinculado/ asoci_entidad1_entidad2]

4.4. Variables globales

La fecha actual en el sistema ser3 representada mediante el siguiente esquema:

Fecha_Actual
 fecha_actual: Fecha

4.5. Estado del sistema

Aerolínea
Insta_Avión_Pe
Insta_VueloPro_Pe
Insta_Vuelo_In
Insta_Ruta_Tr
Insta_Tarifa_Tr
Insta_Pasajero_In
AsociTemp_Asignado
Asoci_Recorre
AsociTemp_Cotiza
Asoci_Reserva
Asoci_Vinculado
Fecha_Actual
$\forall vp: Id_VueloPro \mid vp \in \text{ran asociTemp_asignado}$ $\bullet \#(\text{asoci_reserva} \triangleright \{vp\}) \leq \text{insta_avión}(\text{asociTemp_asignado} \sim (vp)).\text{capacidad}$ $\wedge \langle \text{insta_vuelopro}(vp).\text{libres}, \text{insta_vuelopro}(vp).\text{ocupados} \rangle$ <u>partition</u> 1.. $\text{insta_avión}(\text{asociTemp_asignado} \sim (vp)).\text{capacidad}$

Al reservar un vuelo debe verificarse que la cantidad de asientos reservada no exceda la capacidad máxima del avión. Además, debe distinguirse entre los asientos libres y los ocupados; estos deben ser disyuntos y juntos igualar la capacidad del avión (por simplicidad, se supone que los asientos están numerados consecutivamente).

4.6. Estado inicial del sistema

El estado inicial del sistema se describe mediante el siguiente esquema.

Aerolínea_Inicial
Aerolínea
fecha_actual?: Fecha
$\text{insta_avión_pe} = \emptyset$ $\text{insta_vuelopro_pe} = \emptyset$ $\text{insta_vuelo_in} = \emptyset$ $\text{insta_ruta_tr} = \emptyset$ $\text{insta_tarifa_tr} = \emptyset$ $\text{insta_pasajero_in} = \emptyset$ $\text{asociTemp_asignado} = \emptyset$ $\text{asoci_recorre} = \emptyset$ $\text{asociTemp_cotiza} = \emptyset$ $\text{asoci_reserva} = \emptyset$ $\text{asoci_vinculado} = \emptyset$ $\text{fecha_actual} = \text{fecha_actual?}$

4.7. Marcos de operaciones

A continuación se identifican los marcos de las operaciones del sistema, los cuales delimitan el efecto que tiene cada operación sobre el estado del sistema. Para esto se procede de acuerdo con las indicaciones dadas en [Cordero 1996, Trejos 1998]. Se incluyen solo algunas de las operaciones propuestas en [Montes 2000].

Entidades y Asociaciones Operaciones	Insta_Avión_Pe	Insta_VueloPro_Pe	Insta_Ruta_Tr	AsociTemp_Asignado	AsociTemp_Cotiza	Asoci_Reserva	Asoci_Vinculado
Insertar_Avión_Pe	Δ	Ξ	Ξ	Ξ	Ξ	Ξ	Ξ
Insertar_Ruta_Tr	Ξ	Ξ	Δ	Ξ	Ξ	Ξ	Ξ
Modificar_Avión_Pe	Δ	Ξ	Ξ	Ξ	Ξ	Ξ	Ξ
Eliminar_Avión_Pe	Δ	Ξ	Ξ	Ξ	Ξ	Ξ	Ξ
Programar_Vuelo	Ξ	Δ	Ξ	Δ	Δ	Ξ	Δ

Cuadro No.3: Marcos de operaciones del sistema de reservaciones de una aerolínea.

Las situaciones de error y las operaciones de consulta no afectan el estado de la aerolínea.

4.8. Operaciones

Incluimos a continuación algunas de las operaciones del sistema, para ilustrar la manera en que se aplica la biblioteca de TempER-Z a la especificación de operaciones.

4.8.1. Insertar avión

Esta operación permite insertar nuevos aviones a la flota de la compañía. Para ello se necesita el identificador del avión y los atributos descriptivos. Además, se debe validar que el avión no exista en la flota. El esquema de la operación para insertar un avión se define de la siguiente manera:

Insertar_Avión_Pe₀ $\triangleq$ Operación_Avión $\wedge$
 Insertar_Instanceia_Pe [Id_Avión, Info_Avión]
 [ident_avión / ident_entidad, ident_avión' / ident_entidad',
 insta_avión/ insta_entidad, insta_avión'/ insta_entidad',
 activa_avión / activación, activa_avión' / activación',
 estado_avión / estado, estado_avión' / estado',
 vigen_avión / vigencia, vigen_avión' / vigencia',
 ident_avión? / ident_entidad?, info_avión? / info_entidad?]

Una vez modelada la operación parcial, se definen las situaciones de error que pueden darse al aplicar el esquema de la operación. En este caso la única situación de error se da cuando el avión ya existe (Avión_Ya_Existe), por lo tanto no puede insertarse nuevamente. La operación total para *insertar un avión* sería aquella en que el avión se insertó con éxito, o bien, que se genere un caso de error:

Insertar_Avión_Pe $\triangleq$ Insertar_Avión_Pe₀ $\vee$ Avión_Ya_Existe

4.8.2. Insertar ruta

Esta operación permite insertar nuevas rutas para los vuelos. Para ello se necesita el identificador y la información descriptiva de la instancia *ruta* que se desea insertar. Además, se debe verificar que la ruta no exista.

En primer lugar se define el esquema para inicializar los atributos de la información descriptiva de la entidad *ruta*:

Inicializar_Info_Ruta _____
Info_Ruta'
distancia?: Temporalizado [Numérico]
trayecto?: Temporalizado [Hilera]
momento?: Punto_Tiempo
distancia' = inicializar (distancia?, momento?)
trayecto' = inicializar (trayecto?, momento?)

Posteriormente se define el esquema de la operación para insertar una ruta tomando como base el esquema *Insertar_Instancia_Tr* de la biblioteca e incluyendo *Inicializar_Info_Ruta*.

Insertar_Ruta_Tr _o
Operación_Ruta
Inicializar_Info_Ruta
ident_ruta?: Id_Entidad
ident_ruta? $\notin$ ident_ruta
ident_ruta' = ident_ruta $\cup$ {ident_ruta?}
insta_ruta' = insta_ruta $\cup$ {ident_ruta? $\mapsto$ $\emptyset$ Info_Ruta'}
activa_ruta' = activa_ruta $\cup$ {ident_ruta? $\mapsto$ momento?}
estado_ruta' = estado_ruta $\cup$ {ident_ruta? $\mapsto$ Activa}
vigen_ruta' = vigen_ruta $\cup$ {ident_ruta? $\mapsto$ $\emptyset$ }

Una vez modelada la operación parcial, se definen las situaciones de error que pueden darse al aplicar el esquema de la operación. En este caso la única situación de error se da cuando la ruta ya existe (Ruta_Ya_Existe), por lo tanto no puede insertarse nuevamente. La operación total para *insertar una ruta* sería aquella en que la ruta se insertó con éxito, o bien, que se genere un caso de error:

Insertar_Ruta_Tr $\triangleq$ Insertar_Ruta_Tr_o $\vee$ Ruta_Ya_Existe

4.8.3. Modificar tarifa

Esta operación permite modificar la información descriptiva de una tarifa. Para ello se necesitan: el identificador de la *tarifa* y la información por modificar. Además, se debe validar que la tarifa exista.

Para definir el esquema para modificar una tarifa se deben seguir los pasos para modificar atributos temporalizados de una instancia definidos en [Montes 2000] :

- (1) Se define el esquema de promoción para modificar los atributos de la colección de instancias de la entidad *Tarifa*:

ϕ OpTarifa
Δ Info_Tarifa
Δ Insta_Tarifa_Tr
ident_tarifa?: Id_Tarifa
ident_tarifa? $\in$ ident_tarifa
estado_tarifa (ident_tarifa?) = Activa
$\emptyset$ Info_Tarifa = insta_tarifa (ident_tarifa?)
insta_tarifa' = insta_tarifa $\oplus$ {ident_tarifa? $\mapsto$ $\emptyset$ Info_Tarifa'}

- (2) Se define el esquema de operación para modificar el atributo *monto* de tipo temporalizado:

Modificar_Monto
Δ Info_Tarifa
$\exists$ Info_Tarifa \ (monto, monto')
valor_nuevo?: Dinero
ahora?: Punto_Tiempo
monto' = modificar (monto, valor_nuevo?, ahora?)

- (3) Se define el esquema de la operación para modificar el atributo *monto*:

Op_Modificar_Monto_o $\triangleq$ Operación_Tarifa $\wedge$ ϕ OpTarifa $\wedge$ Modificar_Monto

Una vez modelada la operación parcial, se definen las situaciones de error que pueden darse al aplicar el esquema de la operación. En este caso las situaciones de error se dan cuando la tarifa no existe

(Tarifa_No_Existe) y la tarifa no esta activa (Tarifa_No_Activa). La operación total para *modificar una tarifa* sería aquella en que la tarifa se modificó con éxito, o bien, que se genere un caso de error:

$\text{Modificar_Tarifa_Tr} \triangleq \text{Op_Modificar_Monto}_0 \vee \text{Tarifa_Desconocida} \vee \text{Tarifa_No_Activa}$

4.8.4. Eliminar avión

Esta operación permite eliminar un avión de la flota de la compañía. Para ello se necesita el identificador del *avión* que se desea eliminar. Además, se debe validar que el avión exista.

El esquema de la operación para eliminar un avión se define tomando como base el esquema *Eliminar_Instance_Pe* de la biblioteca y se adecua para verificar que el avión por eliminar no está asociado a vuelos programados:

Eliminar_Avión_Pe ₀
Operación_Avión
ident_avión?: Id_Avión
tiempoeliminación?: Punto_Tiempo
resp!: Respuesta
ident_avión? ∈ ident_avión
ident_avión? ∉ dom asocitemp_asignado
activa_avión (ident_avión?) <u>before</u> tiempoeliminación?
estado_avión (ident_avión?) = Activa ∨ estado_avión (ident_avión?) = Inactiva
estado_avión' = {ident_avión?} ≺ estado_avión
vigen_avión' = {ident_avión?} ≺ vigen_avión
activa_avión' = {ident_avión?} ≺ activa_avión
ident_avión' = ident_avión \ {ident_avión?}
insta_avión' = {ident_avión?} ≺ insta_avión
resp! = 'Avión eliminado con éxito'

Una vez modelada la operación parcial, se definen las situaciones de error que pueden darse al aplicar el esquema de la operación. En este caso las situaciones de error se dan cuando el avión ha sido asociado con anterioridad a vuelos programados (Avión_Asociado_A_Vuelos) o el avión es desconocido por el sistema (Avión_Desconocido). La operación total para *eliminar un avión* sería aquella en que la información del avión se eliminó con éxito, o bien, que se genere un caso de error:

$\text{Eliminar_Avión_Pe} \triangleq \text{Eliminar_Avión_Pe}_0 \vee \text{Avión_Deconocido} \vee \text{Avión_Asociado_A_Vuelos}$

4.8.5. Programar vuelo

Esta operación permite, mediante la información del catálogo de vuelos, programar un vuelo para una fecha y hora dadas. Para ello se necesita el identificador del avión y la tarifa que se asignará al vuelo, así como el identificador del vuelo que se quiere programar. Además se debe verificar que el vuelo no se haya programado con anterioridad para la fecha y hora indicadas. Una vez que el vuelo sea programado quedará en estado de *reserva*.

Primero definimos el esquema para la operación *Cotiza*, con base en el esquema genérico *Asociar*. El objetivo de la operación *Cotiza* es definir la asociación entre las instancias de vuelo programado y tarifa, en un momento específico de tiempo:

$\text{Cotiza} \triangleq \text{Operación_Vuelopro} \wedge$
 $\text{Asociar}[\text{Id_Vuelopro}, \text{Id_Tarifa}]$
 $[\text{ident_vuelopro} / \text{ident_entidad1}, \text{ident_vuelopro}' / \text{ident_entidad1}',$
 $\text{ident_tarifa} / \text{ident_entidad2}, \text{ident_tarifa}' / \text{ident_entidad2}',$

ident_vuelopro? / i₁?, ident_tarifa? / i₂?, momento? / t?,
asocitemp_cotiza / asociemp_entidad1_entidad2]

El objetivo de la operación *Asigna* es definir la asociación entre las instancias de avión y vuelo programado, en un momento específico de tiempo:

$Asigna \triangleq Operación_VueloPro \wedge$
 $Reserva_Conjunto_Tiempo [Id_Avión, Id_Vuelopro]$
 $[ident_avión / ident_entidad1, ident_avión' / ident_entidad1',$
 $ident_vuelopro / ident_entidad2, ident_vuelopro' / ident_entidad2',$
 $ident_avión? / i_1?, ident_vuelopro? / i_2?, intervalo? / ts?,$
 $asocitemp_asignado / asociemp_entidad1_entidad2,$
 $asocitemp_asignado' / asociemp_entidad1_entidad2']$

El esquema para la operación de programación de vuelo es el siguiente:

Programar_Vuelo _o Operación_VueloProgramado Cotiza Asigna ident_avión?: Id_Avión ident_vuelo?: Id_Vuelo ident_tarifa?: Id_Tarifa ident_vuelopro?: Id_VueloPro hora_salida?: Hora fecha_salida?: Fecha resp!: Respuesta	
$\exists Info_VueloPro'; capacidad: \mathbb{N}$ $ ident_avión? \in ident_avión \wedge ident_vuelo? \in ident_vuelo \wedge ident_tarifa? \in ident_tarifa \wedge$ $ident_vuelopro? \notin ident_vuelopro$ $(\forall t: intervalo?; id_vuelopro: Id_VueloPro$ $ (ident_avión?, id_vuelopro) \mapsto t \notin asociemp_asignado)$ $\neg(\exists vp: ident_vuelopro \bullet vp \in asoci_vinculado \sim \{ \{ ident_vuelo? \} \})$ $\wedge insta_vuelopro(vp).hora_salida = hora_salida? \wedge insta_vuelopro(vp).fecha_salida = fecha_salida?)$ $estado_vp' = Reserva \wedge fecha_salida' = fecha_salida? \wedge hora_salida' = hora_salida?$ $capacidad = insta_avión(ident_avión?).capacidad$ $libres' = 1..capacidad$ $ocupados' = \emptyset$ $ident_vuelopro' = ident_vuelopro \cup \{ ident_vuelopro? \}$ $insta_vuelopro' = insta_vuelopro \cup \{ ident_vuelopro? \mapsto \emptyset Info_VueloPro' \}$ $asoci_vinculado' = asoci_vinculado \cup \{ ident_vuelopro? \mapsto ident_vuelo? \}$ $resp! = 'Programación de vuelo concluida con éxito'$	①

Por la forma en que se declara *AsociTemp_Asignado*, se puede deducir que el avión (ident_avión?) no ha sido asignado a algún vuelo programado en los puntos de tiempo contenidos en el intervalo. Esta restricción se expresa en el predicado ①. Una vez modelada la operación parcial, se definen las situaciones de error que pueden darse al aplicar el esquema de la operación: el vuelo ya fue programado con anterioridad (VueloProgramado_Ya_Existente), el avión ya fue asignado a otro vuelo en una fecha y hora que entran en conflicto con la fecha y hora dadas (Avión_Ya_Fue_Asignado), el avión dado no está definido dentro de la flota de aviones de la compañía (avión_desconocido), el vuelo

dado no ha sido definido en el catálogo (Vuelo_Desconocido), la tarifa dada no ha sido definida en el catálogo (Tarifa_Desconocida).

La operación total para *programar un vuelo* sería aquella en que el vuelo se programó con éxito, o bien, que se genere un caso de error:

$$\text{Programar_Vuelo} \triangleq \text{Programar_Vuelo}_o \vee \text{VueloProgramado_Ya_Existe} \vee \text{Avión_Desconocido} \vee \text{Avión_Ya_Fue_Asignado} \vee \text{Vuelo_Desconocido} \vee \text{Tarifa_Desconocida}$$

5. Conclusión

En general la biblioteca de TempER-Z se aplicó de forma directa. Solamente en algunos casos se hizo necesario ajustar los esquemas, como el de eliminación de instancias.

Se presentó un caso excepcional con la aplicación de los esquemas de asociaciones *AsociTemp*, que nos hicieron descubrir la necesidad de definir esquemas genéricos para el manejo de asociaciones temporalizadas con reserva de tiempo.

En la operación para programar un vuelo se considera la definición de entidades perennes (Avión, Vuelo_Programado) y asociaciones temporales (Asignado), para manejar el periodo de tiempo en que un avión es asignado a un vuelo programado.

En [Montes 2000] se pueden encontrar detalles completos del meta-modelo formal, la biblioteca de esquemas genéricos y su aplicación en el caso de estudio completo.

Bibliografía

- [Antunes 1996] Antunes, Dante Carlos; *Modelagem Temporal de Sistemas: uma Abordagem Fundamentada em Redes de petri* (Modelaje Temporal de Sistemas: Una metodología Fundamentada en Redes de Petri); Tesis, Universidade Federal Do Rio Grande Do Sul; 1996.
- [Antunes 1997a] Antunes, Dante Carlos; Heuser, Carlos A; Edelweiss, Nina; *TempER: uma abordagem para modelagem temporal de banco de dados* (TempER: un enfoque para modelaje temporal de bases de datos); Revista de Informática Teórica e Aplicada; Volumen VI; Número 1; Agosto 1997.
- [Antunes 1997b] Antunes, Dante Carlos; Heuser, Carlos A; Edelweiss, Nina; *TempER: Modelagem temporal de transações em banco de dados* (Modelaje temporal de transacciones en bases de datos); 1997.
- [Cordero 1996] Cordero Ríos, Jessica; *Bases para la integración de métodos estructurados y formales en la especificación de sistemas de información*; Tesis, Instituto Tecnológico de Costa Rica; 1996.
- [Montes 2000] Montes López, María Lourdes; *Especificación formal de sistemas de información temporalizados*; Tesis, Instituto Tecnológico de Costa Rica; 2000.
- [Montes 2000a] Montes López, María Lourdes; Trejos Zelaya, Ignacio; *Especificación formal del modelo entidad-asociación temporalizado TempER-Tr*; 2000.
- [Montes 2000b] Montes López, María Lourdes; Trejos Zelaya, Ignacio; *Especificación formal de sistemas de información temporalizados: TempER-Z*; 2000.
- [Spivey 1992] Spivey, J. Michael; *The Z Notation: A Reference Manual*, 2nd. Ed.; Prentice-Hall; 1992.
- [Trejos 1995] Trejos Zelaya, Ignacio; *Especificación de software mediante modelos: la notación Z*; Primer Curso Iberoamericano de Informática, Santa Cruz de la Sierra, Bolivia; Mayo 1995; Páginas 1-38.
- [Trejos 1998] Trejos Zelaya, Ignacio; Cordero R. Jessica; *Integración de Métodos Estructurados y Formales*; CIC'98: IV Congreso de Informática y Computación, Costa Rica; 1998.
- [Wordsworth 1992] Wordsworth, John; *Software development with Z: A Practical Approach to Formal Methods in Software Engineering*; Addison-Wesley; 1992.