dit
UPM

Diseño de sistemas de tiempo real - Introducción a HRT-HOOD

Juan Antonio de la Puente
DIT/UPM

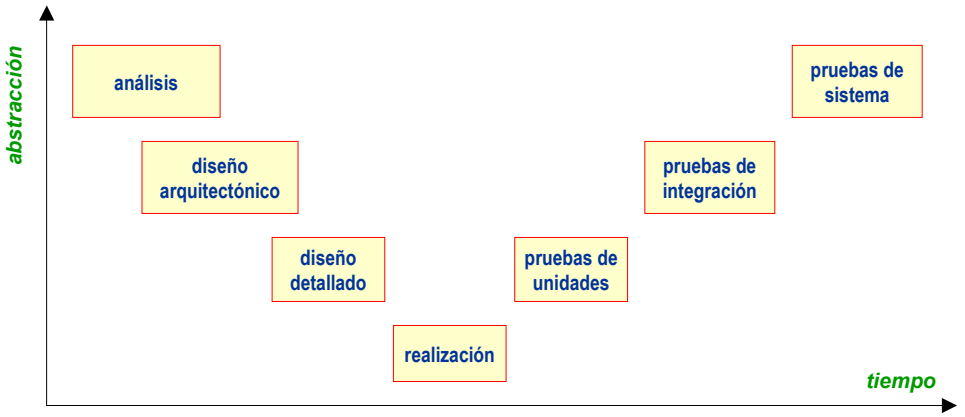
Índice

HRT-HOOD

(Hard Real-Time Hierarchical Object-Oriented Design

- Es un método de diseño estructurado, basado en objetos, para sistemas de tiempo real estricto
  - derivado de HOOD *(Hierarchical Object-Oriented Design)*
- Un sistema se diseña como una jerarquía de **objetos abstractos**
  - un objeto se caracteriza por sus **operaciones** y su **comportamiento** (abstracción y ocultamiento de información)
  - cada objeto se puede descomponer en otros de más bajo nivel
- Se puede analizar el **comportamiento temporal** si el entorno de ejecución es conocido y predecible
  - los objetos tienen **atributos temporales**
  - las **relaciones** entre objetos están **restringidas** para asegurar que el diseño se puede analizar

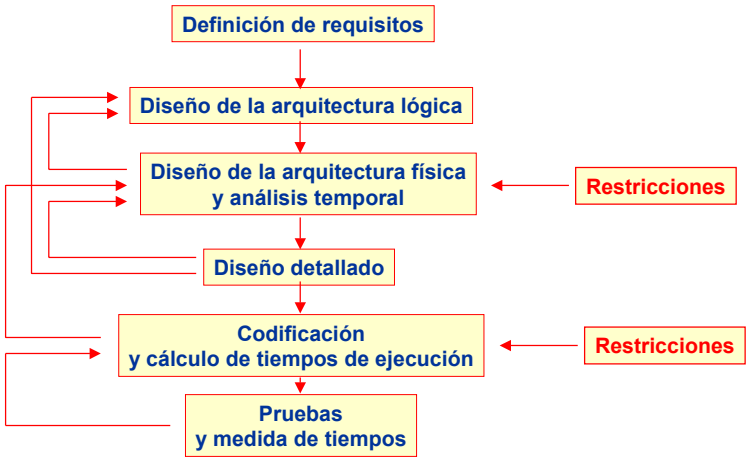
Proceso de desarrollo secuencial



Características

- Se descompone en una secuencia de ***etapas***
  - Hay que completar cada etapa antes de empezar la siguiente
- Las pruebas se llevan a cabo después de la realización
  - Muchos errores se encuentran sólo al final
  - Volver atrás es muy costoso
  - A veces se hace sin documentar y de forma poco rigurosa
- Es mejor utilizar un proceso iterativo
  - Veremos uno centrado en la etapa de diseño
  - Se trata de *validar* todos los aspectos que se pueda en la etapa de diseño del sistema
  - Prestaremos especial atención a la validación del comportamiento temporal

Proceso de desarrollo iterativo



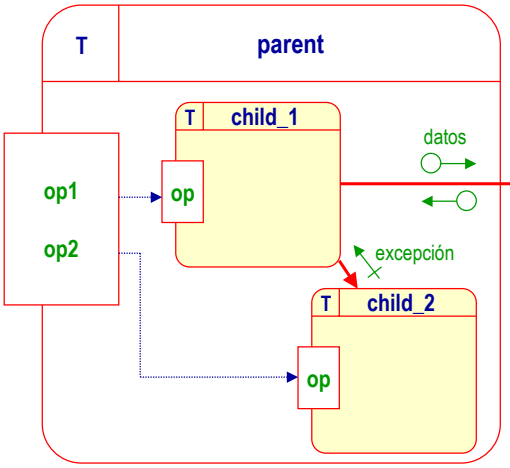
Características

- Elementos en cada nivel de abstracción
  - **Compromisos:** propiedades que ya no se cambiarán
  - **Obligaciones** que se dejan para niveles inferiores
- En el diseño se van transformando las obligaciones en compromisos
  - Este proceso está sujeto a **restricciones** impuestas por el entorno de ejecución
- Dos niveles de diseño
  - **Modelo lógico** - compromisos independientes del entorno
  - **Modelo físico** - compromisos dependientes del entorno

Índice

- Introducción
- **Arquitectura lógica**
  - objetos y operaciones
  - reglas de descomposición jerárquica y uso
- Arquitectura física
  - atributos temporales
- Realización en Ada

Objetos y relaciones



El objeto **parent** ***contiene***
los objetos **child_1** y **child_2**

parent ***usa*** **uncle**
child_1 ***usa*** **child_2** y **uncle**

child_1.op ***implementa*** **parent.op1**
child_2.op ***implementa*** **parent.op2**

Tipos de objetos

- **Pasivos**
  - » no controlan cuándo se ejecutan sus operaciones
  - » no invocan operaciones de otros objetos espontáneamente
- **Protegidos**
  - » pueden controlar cuándo se ejecutan sus operaciones
  - » no invocan operaciones de otros objetos espontáneamente
- **Activos**
  - » pueden controlar cuándo se ejecutan sus operaciones
  - » pueden invocar operaciones de otros objetos espontáneamente
  - – **Cíclicos**
  - – **Esporádicos**

Objetos y operaciones (1)

- **Objetos pasivos**
  - sus operaciones se ejecutan en cuanto se invocan
- **Objetos activos**
  - sus operaciones pueden bloquearse
    - » por *restricciones de invocación* (asíncrona, síncrona, invocación remota)
      - lós dos últimos tipos pueden tener un *límite de tiempo*
    - » por *restricciones funcionales*, dependiendo del estado interno del objeto
- **Objetos protegidos**
  - sus operaciones se ejecutan en *exclusión mutua*
    - » la invocación puede ser *asíncrona* o *síncrona*, y en este caso pueden tener temporizaciones
    - » también pueden tener restricciones funcionales

Objetos y operaciones (2)

- **Objetos cíclicos**
  - las únicas operaciones que pueden tener son *transferencias de control asíncronas* (ATC)
    - » pueden tener restricciones de invocación y temporizacinoes
- **Objetos esporádicos**
  - tienen una única operación asíncrona: START
    - » puede estar ligada a una interrupción
  - pueden tener operaciones de transferencia de control asíncrona

Diseño de la arquitectura lógica

- Aplicación repetida de descomposición descendente
- Resultado: sistema de objetos **terminales**
  - ***Cíclicos***
  - ***Esporádicos***
  - ***Protegidos***
  - ***Pasivos***
- Los objetos activos sólo se permiten en actividades de segundo plano *(background)*, ya que no se pueden analizar
- Hay *reglas de descomposición y uso* que posibilitan el análisis del comportamiento temporal

Reglas de descomposición

utilizador \ utilizado	*activo*	*cíclico*	*esporádico*	*protegido*	*pasivo*
activo	sí	sí	sí	sí	sí
cíclico	no	sí	sí	sí	sí
esporádico	no	sí	sí	sí	sí
protegido	no	no	no	sí	sí
pasivo	no	no	no	no	sí

Reglas de uso

utilizador \ utilizado	*activo*	*cíclico*	*esporádico*	*protegido*	*pasivo*
activo	sí	sí	sí	sí	sí
cíclico	sólo asíncrono	sí	sí	sí	sí
esporádico	sólo asíncrono	sí	sí	sí	sí
protegido	sólo asíncrono	sólo ATC	sólo ATC / START	sí	sí
pasivo	no	no	no	no	sí

Índice

Diseño de la arquitectura física

- Objetivo
  - relacionar la arquitectura lógica con recursos de ejecución
  - asegurar el cumplimiento de los requisitos no funcionales
- Actividades
  - Asignación de objetos a procesadores
  - Planificación de la red de comunicaciones
  - Planificación de procesador
  - Análisis de fiabilidad
- Resultado:
  - sistema de objetos **con atributos temporales** y de otros tipos
  - **análisis temporal**
  - otros tipos de análisis

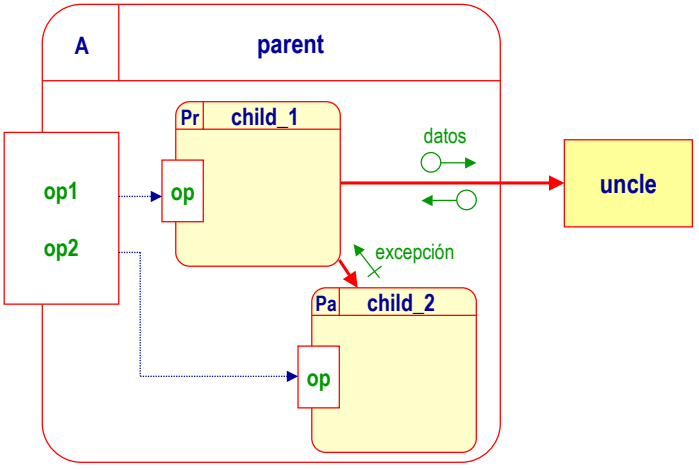
Índice

- Introducción
- Arquitectura lógica
  - objetos y operaciones
  - reglas de descomposición jerárquica y uso
- Arquitectura física
  - atributos temporales
- **Realización en Ada**

Diseño detallado y codificación

- Se trata de diseñar la estructura de módulos del sistema y de efectuar la codificación en un lenguaje de programación
- Ada 95 permite una codificación sencilla y directa del diseño arquitectónico
  - cada **objeto** se realiza mediante un *paquete*
  - las relaciones de **uso** se representan mediante cláusulas *with*
  - los objetos **hijos** se realizan mediante paquetes *hijos privados*
  - la **implementación** de operaciones se realiza mediante cláusulas de *renombrado*
  - los objetos **cíclicos** y **esporádicos** tienen una *tarea* y, posiblemente, un *objeto protegido* de sincronización
  - los objetos **protegidos** se realizan mediante *objetos protegidos* de Ada
  - puede haber elementos adicionales si es necesario

Ejemplo (1)



Ejemplo (2)

```
with Uncle;
package Parent is

  type T is ...;
  procedure OP1(...);
  procedure OP2(...);

end Parent;
```

```
private package Parent.Child_1 is
-- Uncle es visible aquí
   procedure OP(...);
end Parent.Child_1;
```

```
private package Parent.Child_2 is
  procedure OP(...);
end Parent.Child_2;
```


Ejemplo (3)

```
with Parent.Child_1, Parent.Child_2;
package body Parent is

  procedure OP1(...) renames Child_1.OP;
  procedure OP2(...) renames Child_2.OP;

end Parent;
```

Atributos de tiempo real (1)

```
with System;        use System;
with Ada.Real_Time; use Ada.Real_Time,
package Real_Time_Attributes is

  type System_Wide_Mode is (Initial, Normal, Emergency);

  type Operation_Attributes is tagged
    record
      Execution_Time : Time_Span;
    end record;

  type Thread_Attributes is tagged
    record
      P              : Priority;
      Execution_Time : Time_Span;
      Deadline       : Time_Span;
    end record;
```

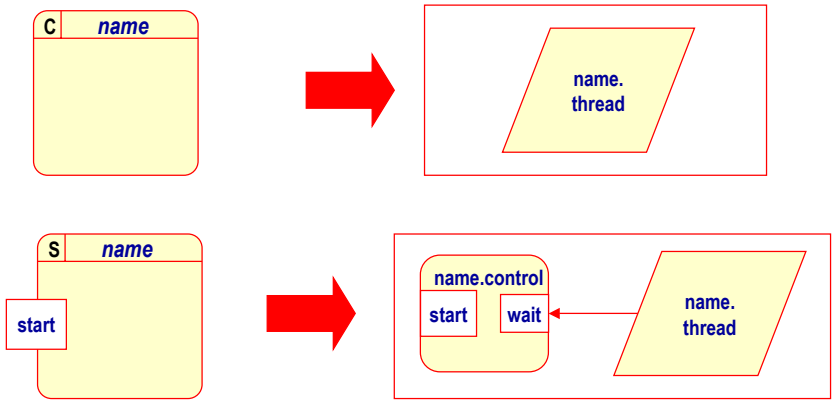

Atributos de tiempo real (2)

```
  type Periodic_Thread_Attributes
    is new Thread_Attributes with
    record
      Period : Time_Span;
      Offset : Time_Span;
    end record;

   --  otras definiciones

   Start_Mode   : System_Wide_Mode := Initial;
   Current_Mode : System_Wide_Mode;
   Start_Time   : Time := Clock;

end Real_Time_Attributes;
```

Atributos de tiempo real (3)

```
package Name.Real_Time_Attributes is

  Operation : Operation_Attributes := (...);

  Thread    : Periodic_Thread_Attributes := (...);

  ...

end Name.Real_Time_Attributes
```


Objetos cíclicos y esporádicos



Resumen

- HRT-HOOD es un método de diseño basado en objetos
- Hay varios tipos de objetos
  - Pasivos
  - Protegidos
  - Activos
    - » cíclicos
    - » esporádicos
- Dos etapas
  - diseño lógico
  - diseño físico
- Las relaciones de descomposición jerárquica y uso permiten mantener la posibilidad de analizar el comportamiento temporal