

La Programación Basada en Component Pascal : De la Programación Orientada a Objetos a los Componentes Software.

D^a. Ana Esteban Sanchez, D. Eduardo Falces Larraga
D. Jose Luis Vicén Cruz, D. Rafael Embid Romero

Escuela Universitaria Politécnica La Almunia de D^a Godina (Zaragoza)

E-mail : eduardo@eupla.unizar.es

joselo@eupla.unizar.es

Resumen

En esta ponencia, se presenta la experiencia de la Escuela Universitaria Politécnica de La Almunia de D^a Godina (Zaragoza), en la docencia de la programación de ordenadores utilizando un lenguaje de programación de la familia del Pascal, pero con características modernas, que incorporan elementos de ingeniería del software, tales como : P.O.O, Frameworks, Patterns, Componentes Software, COM.

1 Introducción

Desde hace tres años, la docencia de la programación de ordenadores en primer curso de Ingeniería Técnica en Informática de Sistemas, se viene realizando en Component Pascal.

La experiencia de estos tres años, la consideramos muy positiva y la vamos a intentar plasmar en esta ponencia.

También se plasmarán las actividades extracurriculares, que se han realizado con este lenguaje de programación.

No queremos dar una idea exclusivista y debe quedar claro que aunque la ponencia trata de Component Pascal, parte de las actividades de programación se realizan, entre otros lenguajes, en C++ y JAVA.

2 Component Pascal en la programación de ordenadores

La asignatura de programación del primer curso, se puede contemplar como tres bloques básicos :

2.1 Primer Bloque

El primer bloque es un primer contacto con la programación y se intenta que el alumno asuma la programación con rigor.

Se ve especificación basada en lógica de predicados, tipos de datos estáticos y estructuras de control. Se hace especial énfasis en el diseño descendente de algoritmos, como método básico de desarrollo.

2.2 Segundo Bloque

El segundo bloque trata de estructuras dinámicas de datos, tipos procedimiento y programación visual desarrollada bajo Windows. Se ven estructuras abstractas de datos y tipos abstractos de datos, que son la antesala a la programación orientada a objetos.

2.3 Tercer Bloque

El tercer bloque trata a fondo la programación orientada a objetos, asentando los siguientes conceptos:

encapsulado, herencia, polimorfismo, redefinición de métodos, binding dinámico, persistencia y estructuras de datos heterogéneas.

Se contempla la gestión de ficheros desde el punto de vista de clases.

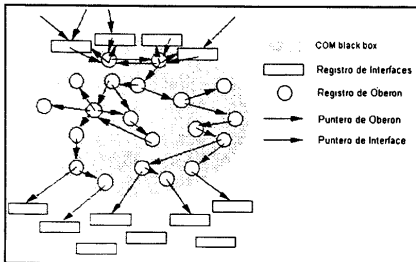
3 Component Pascal en el desarrollo de Componentes Software

El entorno de desarrollo de aplicaciones bajo Component Pascal, se denomina Black Box Component Builder, es una potente herramienta de desarrollo R.A.D e integra su propio modelo

de Componente Software.

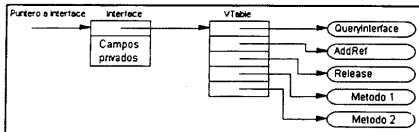
Uno de los motivos por los que se ha elegido este lenguaje es el de la seguridad. Posee un colector de basura (Garbage Collector) el cual descarga automática-mente los componentes de memoria y además solventa el conteo cíclico de referencias.

Esta herramienta de desarrollo, puede incorporar una herramienta de generación de componentes COM de Microsoft "Figura 1" (Direct To COM) y su correspondiente debugger.



(Figura 1)

A continuación un ejemplo de La estructura del interface de un Componente COM de Microsoft ver "Figura 2".



(Figura 2)

Direct To Com, es una herramienta que se instala en el entorno de desarrollo Black Box Component Builder que facilita el desarrollo de COM's, posee su propio generador de números GUID.

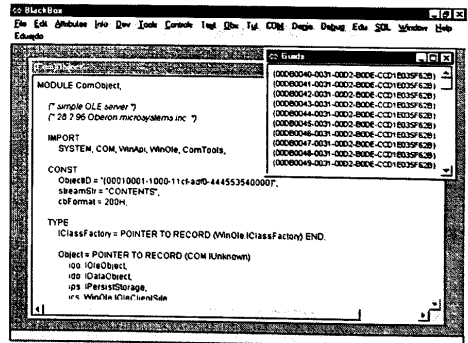
También dispone de la opción *Show Interfaces* "Figura 3" que muestra los interfaces de los componentes generados, así como el número de referencias a estos.

```

ref: 0 OleData.IdataObject^ [01009850H] ◆
(StdLog.???)
ref: 0 HostMechanisms.IdropTarget^ [0100EBB0H] ◆
(StdLog.???)
ref: 1 HostMechanisms.IdropTarget^ [0100EF00H] ◆
(StdLog.???)
ref: 1 OleServer.IClassFactory^ [010314C0H] ◆
(OleServer.factory)
ref: 1 HostMechanisms.IdropTarget^ [0103A620H] ◆
(StdLog.???)
ref: 1 HostMechanisms.IdropTarget^ [010455F0H] ◆
(StdLog.???)
ref: 0 OleData.IdataObject^ [0105E230H] ◆
(StdLog.???)
ref: 0 OleData.IenumFORMATETC^ [0105E2E0H] ◆
(StdLog.???)
ref: 1 ComKoala.KoalaFactory^ [0105E600H] ◆
ref: 1 HostMechanisms.IdropTarget^ [01064860H] ◆
ref: 2 OleData.IdataObject^ [010666D0H] ◆
ref: 0 OleData.IdataObject^ [01071F50H] ◆
(StdLog.???)
ref: 1 HostMechanisms.IdropTarget^ [010A63D0H] ◆
(StdLog.???)
    
```

(Figura 3)

A continuación el ejemplo del diseño de un componente COM así como la generación de GUI's (identificadores globales únicos) "Figura 4" para asociarlos a los interfaces de cada componente.



(Figura 4)

4 Component Pascal en el desarrollo de aplicaciones de tiempo real

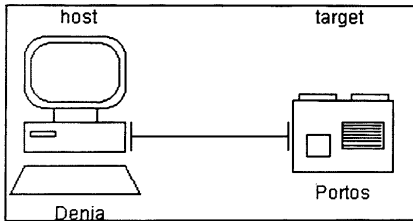
Referenced Interface Records: [New](#) [Update](#)

Dentro del entorno de desarrollo Black Box Component Builder, se instala el compilador cruzado Denia, con el cual podremos diseñar aplicaciones en Component Pascal o JAVA.

Estas aplicaciones se ejecutan remotamente como procesos sobre un sistema operativo de tiempo real llamado Portos. Este sistema operativo está instalado en una tarjeta MVME162IG/D2 (bus VME) de Motorola, diseñada para control industrial (68040 a 25Mhz "32 Bits" desde 1Mb a 8Mb de DRAM).

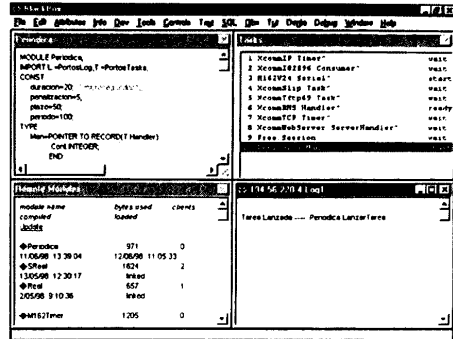
Esta tarjeta podrá estar conectada directamente a un PC via serie, "Figura 5" o bien podrá estar conectada en red utilizando un protocolo TCP/IP, con multiples usuarios trabajando con PC's simultaneamente sobre esta tarjeta.

Sobre la tarjeta que está instalado Portos se pueden añadir otras tarjetas especificas llamadas "Industry Packs", las cuales se programan desde la plataforma cruzada Denia.



(Figura 5)

En el siguiente ejemplo " Figura 6 " se muestra una aplicación que lanza una tarea periodica a traves de la plataforma cruzada Denia via red, en el sistema operativo Portos, el cual está instalado en la tarjeta de Motorola, anteriormente citada.



(Figura 6)

En la figura anterior se muestra la ejecución de una tarea periódica implementada en el módulo, se muestra el estado de los módulos que están ubicados en el sistema operativo Portos, así como las tareas y su estado.

```

MODULE Periodica;
IMPORT L:=PortosLog,
       T:=PortosTasks;

CONST
    duracion=20; (* microsegundos *)
    penalizacion=5;
    plazo=50;
    periodo=100;

TYPE
    Man=POINTER TO RECORD(T.Handler)
    Cont:INTEGER ;

END;

VAR
    manejador:Man;
    tarea:T.Task;

PROCEDURE (obj:Man)Run;
BEGIN
    obj.Cont:=(obj.Cont+1) MOD 10;
END Run;

PROCEDURE LanzarTarea*;
BEGIN
    L.Ln; NEW(manejador);
    tarea:=T.NewPeriodicTask(
        manejador,duracion,
        penalizacion,plazo,periodo);
    IF tarea#NIL THEN
    
```

```

L.String("Tarea Lanzada.");
tarea.Start;
ELSE
    L.String("No puedo lanzar la tarea.")
END;
END LanzarTarea;

PROCEDURE Opcion(n:INTEGER);
BEGIN
    L.Ln;
    IF tarea=NIL THEN
        L.String("Debe lanzar una tarea.")
    ELSE
        CASE n OF
            0: tarea.Stop;
                L.String("Tarea finalizada.")
            | 1: tarea.Suspend;
                L.String("Tarea bloqueada.")
            ELSE tarea.Resume;
                L.String("Tarea reanudada.")
        END
    END;
END Opcion;

PROCEDURE FinalizarTarea*;
BEGIN
    Opcion(0)
END FinalizarTarea;

PROCEDURE BloquearTarea*;
BEGIN
    Opcion(1)
END BloquearTarea;

PROCEDURE ContinuarTarea*;
BEGIN
    Opcion(2)
END ContinuarTarea;

END Periodica.

Periodica.LanzarTarea;
Periodica.FinalizarTarea;
Periodica.BloquearTarea;
Periodica.ContinuarTarea;

```

- [1] Kraig Brockschmidt,
Inside OLE,
2nd Edition, 1995.
- [2] David Chappell,
Understanding ActiveX and OLE,
Microsoft Press, 1996.
- [3] Adam Denning,
OLE Controls Inside Out,
Microsoft Press, 1995.
- [4] Dale Rogerson,
Inside COM,
Microsoft Press, 1997
- [5] Don Box,
*Creating Components with
DCOM and C++*,
Addison Wesley, 1997.
- [6] Microsoft
COM Home Page,
<http://www.microsoft.com/com>
- [7] Oberon microsystems
<http://www.oberon.ch>

Código fuente en Component Pascal correspondiente a la aplicación anterior

Referencias