

IBM Software Group

Diseño y Modelado de Sistemas

Francisco López Minaya
Technical Sales – Rational Technical Solution Architect
francisco.lopezminaya@es.ibm.com



Rational software

→ Go to IBM

IV Mejores Prácticas en el Desarrollo de Sistemas
27 Mayo 2009 - Madrid

© 2009 IBM Corporation



| IBM Software Group

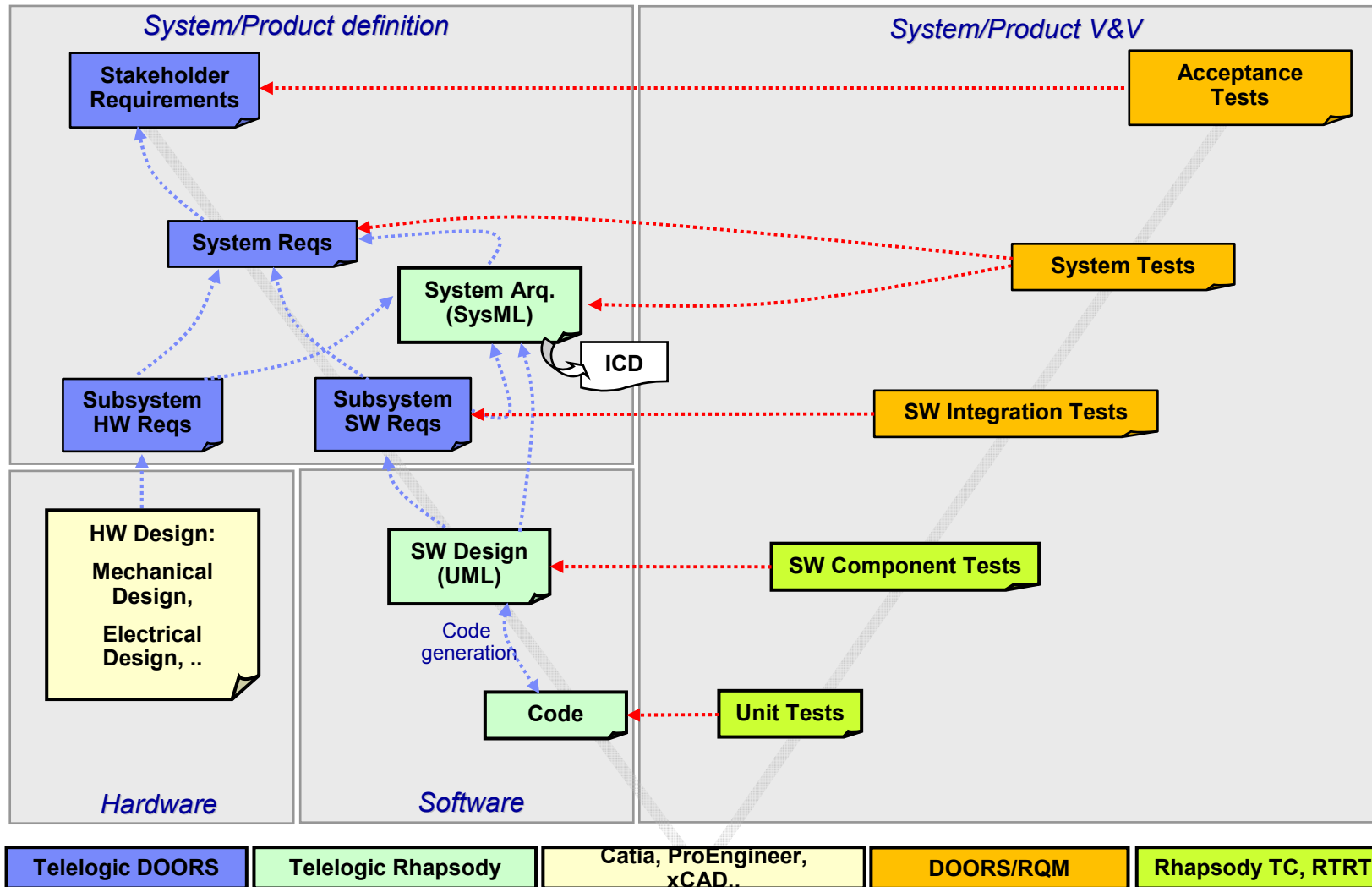
El proceso de desarrollo y construcción de sistemas

Rational. software

→ Go to **IBM**

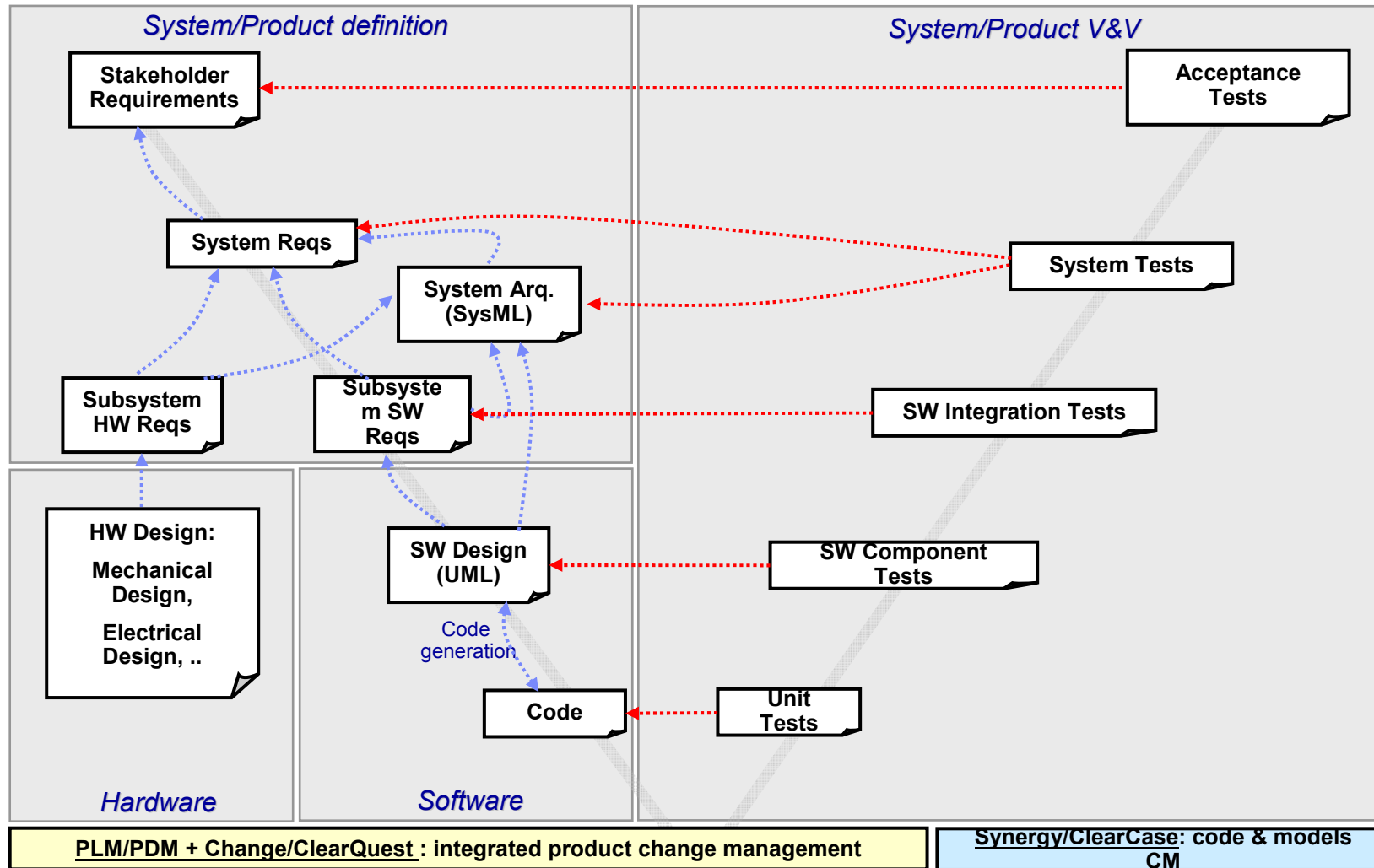
Plataforma para desarrollo de sistemas (I)

Definición, Desarrollo, Construcción y Verificación & Validación



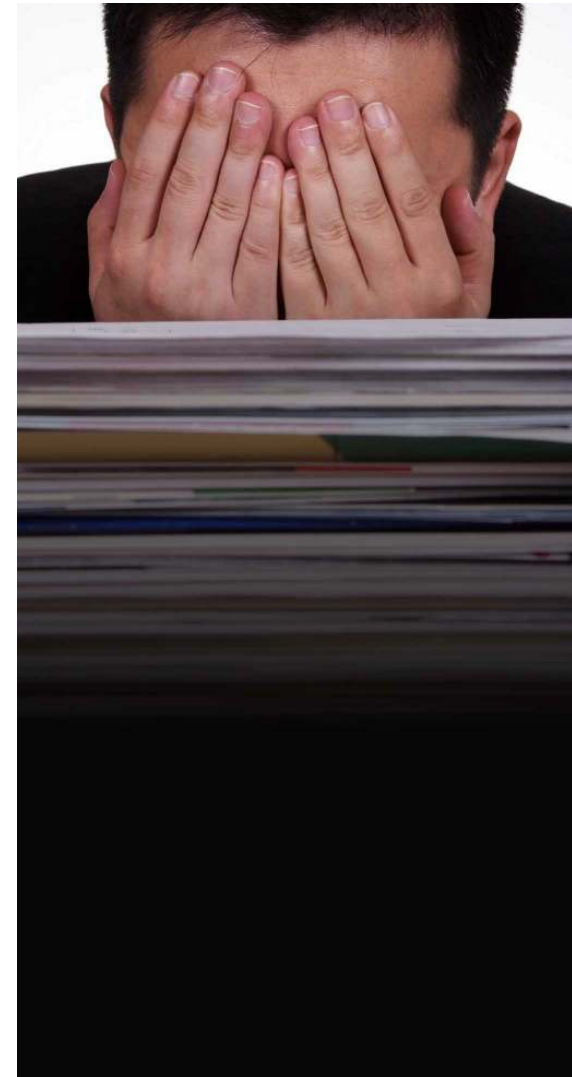
Plataforma para desarrollo de sistemas (II)

Gestión de Cambios y de la Configuración

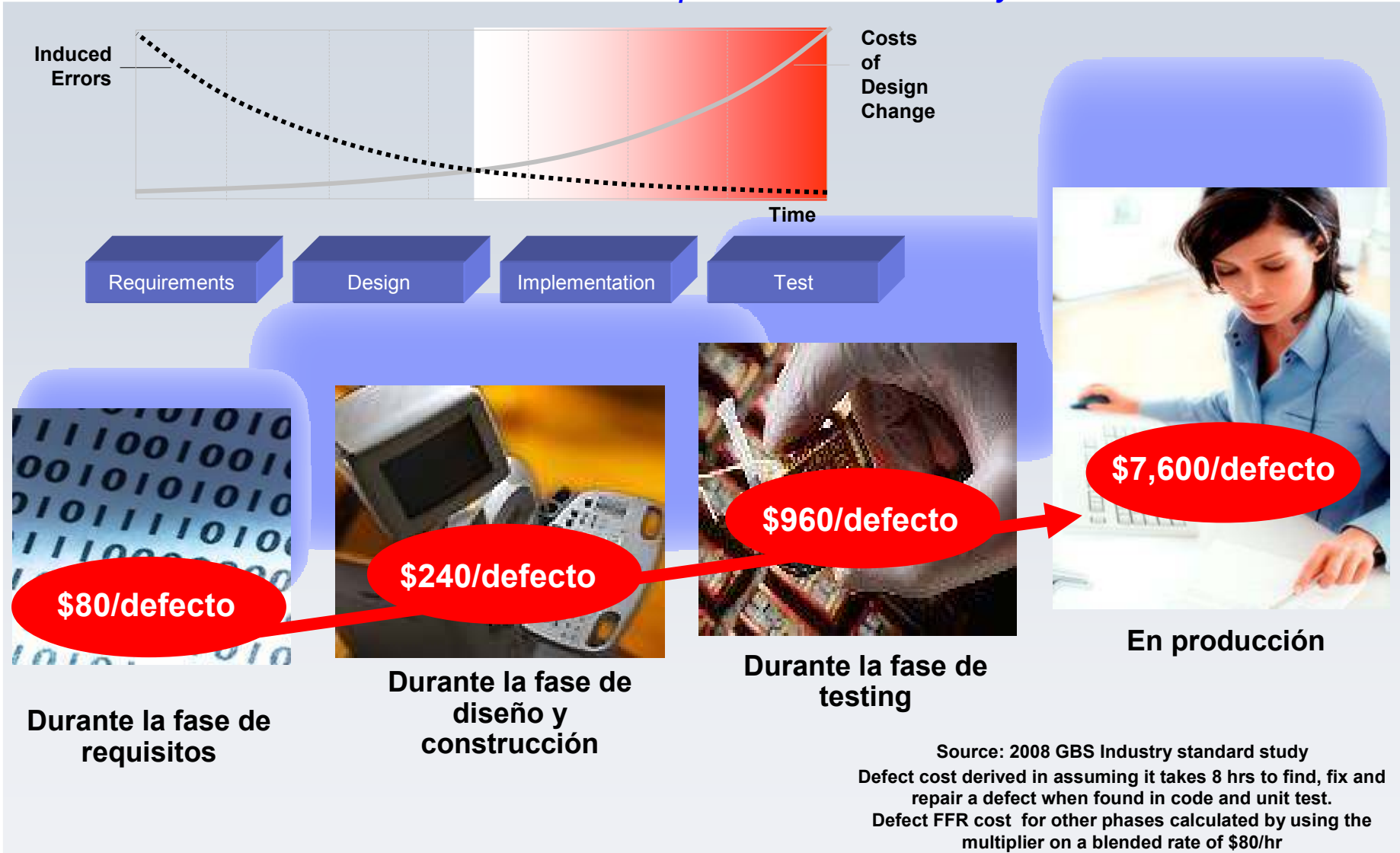


¿A qué nos enfrentamos en la realidad?

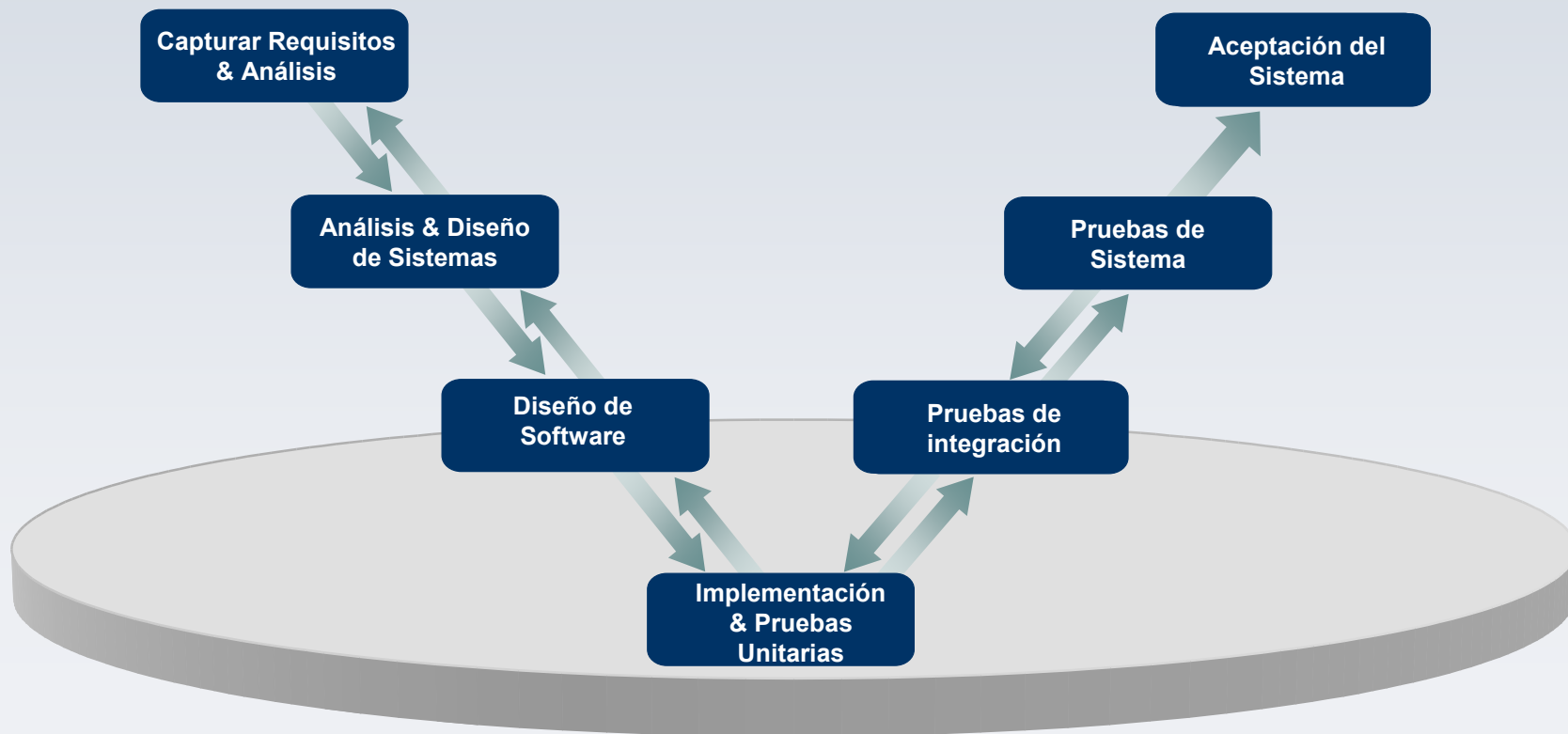
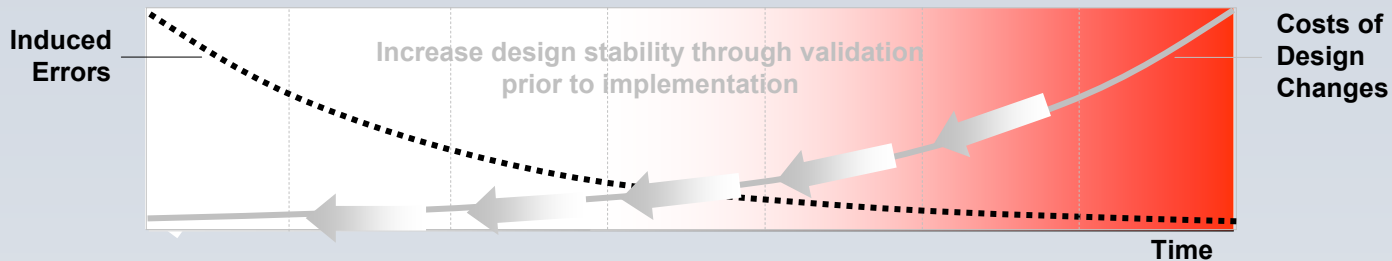
- Se encuentran **errores de diseño demasiado tarde**, cuando más cuesta solucionarlos.
- O no se encuentran: el **coste de los errores no encontrados es muy alto** (retirada o devolución del sistema, penalizaciones, reputación).
- **Baja productividad en procesos de prueba**. Procesos lentos y poco ágiles
- La **colaboración** entre equipos grandes de desarrollo que están distribuidos es cada vez más **difícil**.
- **No hay comunicación** efectiva entre diferentes disciplinas: ingenieros de sistemas, ingenieros mecánicos, ingenieros eléctricos, ingenieros de software
- Cada vez es más **difícil la gestión y la integración de diseños** cada vez más complejos
- **Poca reutilización**, se vuelve a construir algo que ya existía



El tiempo de desarrollo se debe emplear en desarrollar...
El 80% de los costes de desarrollo se emplean en identificar y solucionar errores



Cuanto antes pruebe antes encuentro defectos...



Los sistemas evolucionan, y cómo se construyen también...

ASM

```

format  ELF execut
entry   start

start:

mov     esi,logc
call   display_s

mov     [command]
pop     eax
lea    esp,[esp+
pop     eax
pop     [environm
call   get_param
jc     informati

call   init_memc

mov     edi,chara
mov     ecx,100h
xor     al,al
make_characters_tab
stosb
inc    al
loop   make_char
mov     esi,characters+'a'
    
```

C

```

void CruiseControl_init(_C_C
{
    CruiseSpeedMgt_init(&(_C_C
    CruiseStateMgt_init(&(_C_C
    (_C->M_conduct_0) = tr
    ThrottleCmd_init(&(_C->
    (_C->M_init) = true;
}

/* ===== */
/* MAIN NODE */
/* ===== */

void CruiseControl(_C_Cruise
{
    bool BrakePressed;
    bool AcceleratorPressed;
    bool SpeedOutOffLimits;
    bool _L19;
    /*#code for node CruiseContr
    /* call to node not expanded
    (_C->Cn_DetectPedalsPr
    (_C->Cn_DetectPedalsPr
    DetectPedalsPressed(&(_C
    BrakePressed = (_C->Cn
    AcceleratorPressed =
    (_C->Cn_DetectPed
    /* call to node not expanded DetectSpeedLimits */
    (_C->Cn_DetectSpeedLimits_I0_speed) = (_C->I8
    DetectSpeedLimits(&(_C->Cn_DetectSpeedLimits));
    SpeedOutOffLimits = (_C->Cn_DetectSpeedLimits._00
    /* call to node not expanded CruiseStateMgt */
    (_C->C3_CruiseStateMgt_I0_BrakePressed) = BrakeP
    
```

```

void CruiseControl_init(_C_CruiseControl *_C_C_
{
    CruiseSpeedMgt_init(&(_C->CruiseSpeedMgt);
    CruiseStateMgt_init(&(_C->CruiseStateMgt);
    (_C->M_conduct_0) = true;
    ThrottleCmd_init(&(_C->ThrottleCmd);
    (_C->M_init) = true;
}

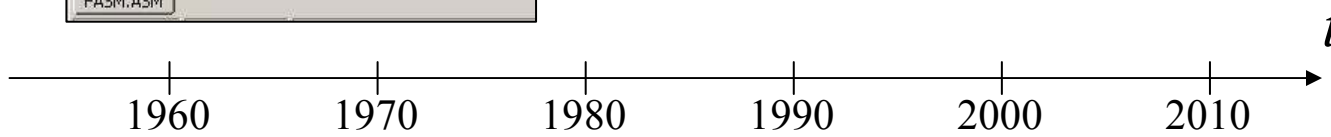
/* ===== */
/* MAIN NODE */
/* ===== */

void CruiseControl(_C_CruiseControl *_C_C_
{
    bool BrakePressed;
    bool AcceleratorPressed;
    bool SpeedOutOffLimits;
    bool _L19;
    /*code for node CruiseControl
    /* call to node not expanded DetectPedalsPressed */
    (_C->Cn_DetectPedalsPressed_I0_BrakePressed) = (_C->I8
    (_C->Cn_DetectPedalsPressed_I1_AcceleratorPressed) =
    DetectPedalsPressed(&(_C->Cn_DetectPedalsPressed);
    BrakePressed = (_C->Cn_DetectPedalsPressed._00_I
    AcceleratorPressed =
    (_C->Cn_DetectPedalsPressed._01_AcceleratorP
    /* call to node not expanded DetectSpeedLimits */
    (_C->Cn_DetectSpeedLimits_I0_speed) = (_C->I8
    DetectSpeedLimits(&(_C->Cn_DetectSpeedLimits);
    SpeedOutOffLimits = (_C->Cn_DetectSpeedLimits._
    
```

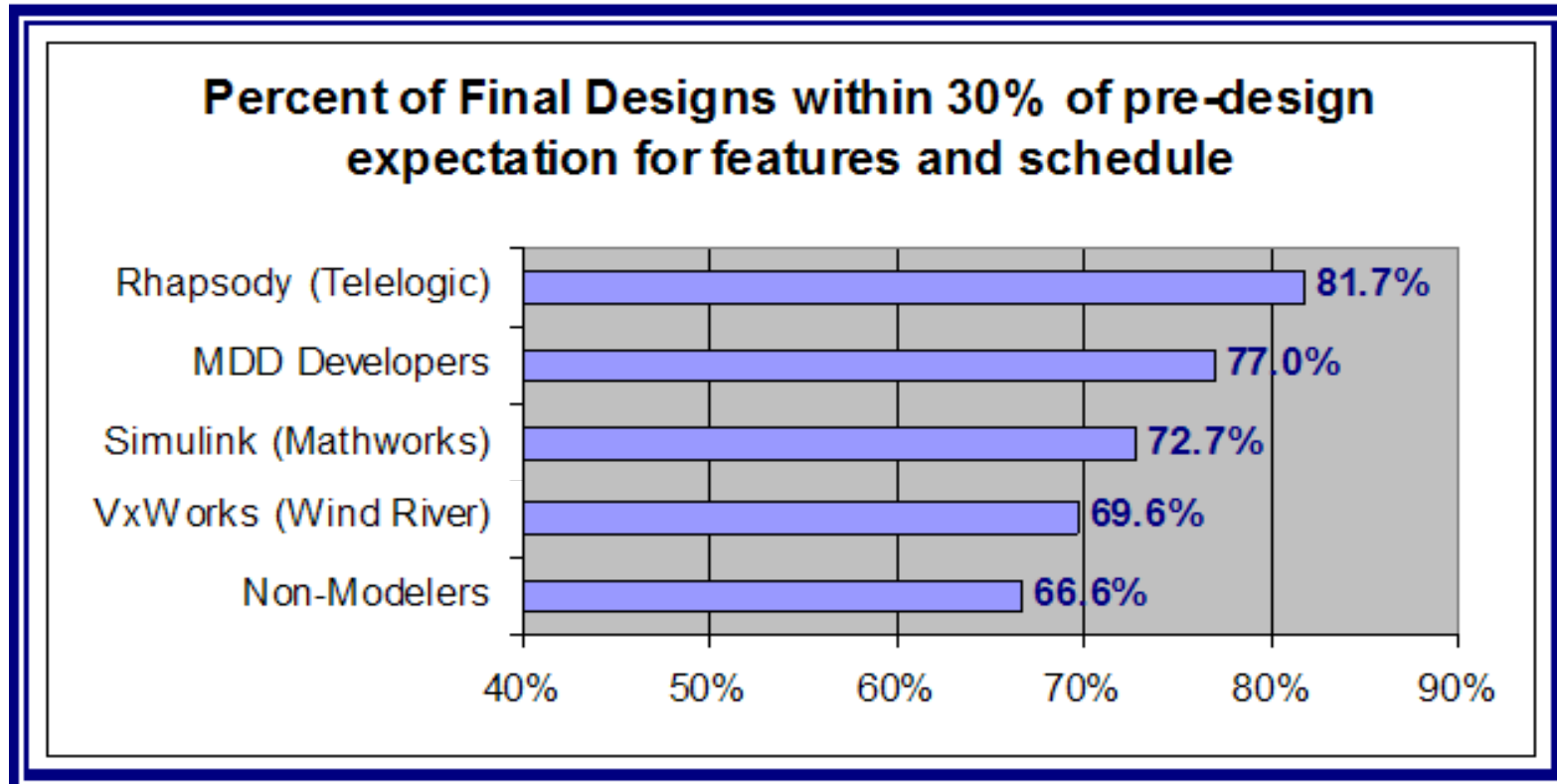
Model Driven Development MDD

Model Driven Development (MDD):

- Mejora la comunicación
- Modelos más fácilmente trazables que el código
- Modelos permiten automatización:
 - ✓ Simulación del modelo
 - ✓ Generación de Código



Model Driven Development es la forma más productiva de desarrollar sistemas: **PROBADO!!!**



Embedded Market Forecasters

Documented in "What Do You Do When the Horse You're Riding Drops Dead? Why Model Driven Design is Emerging as a Preferred Best Practice", March 2007



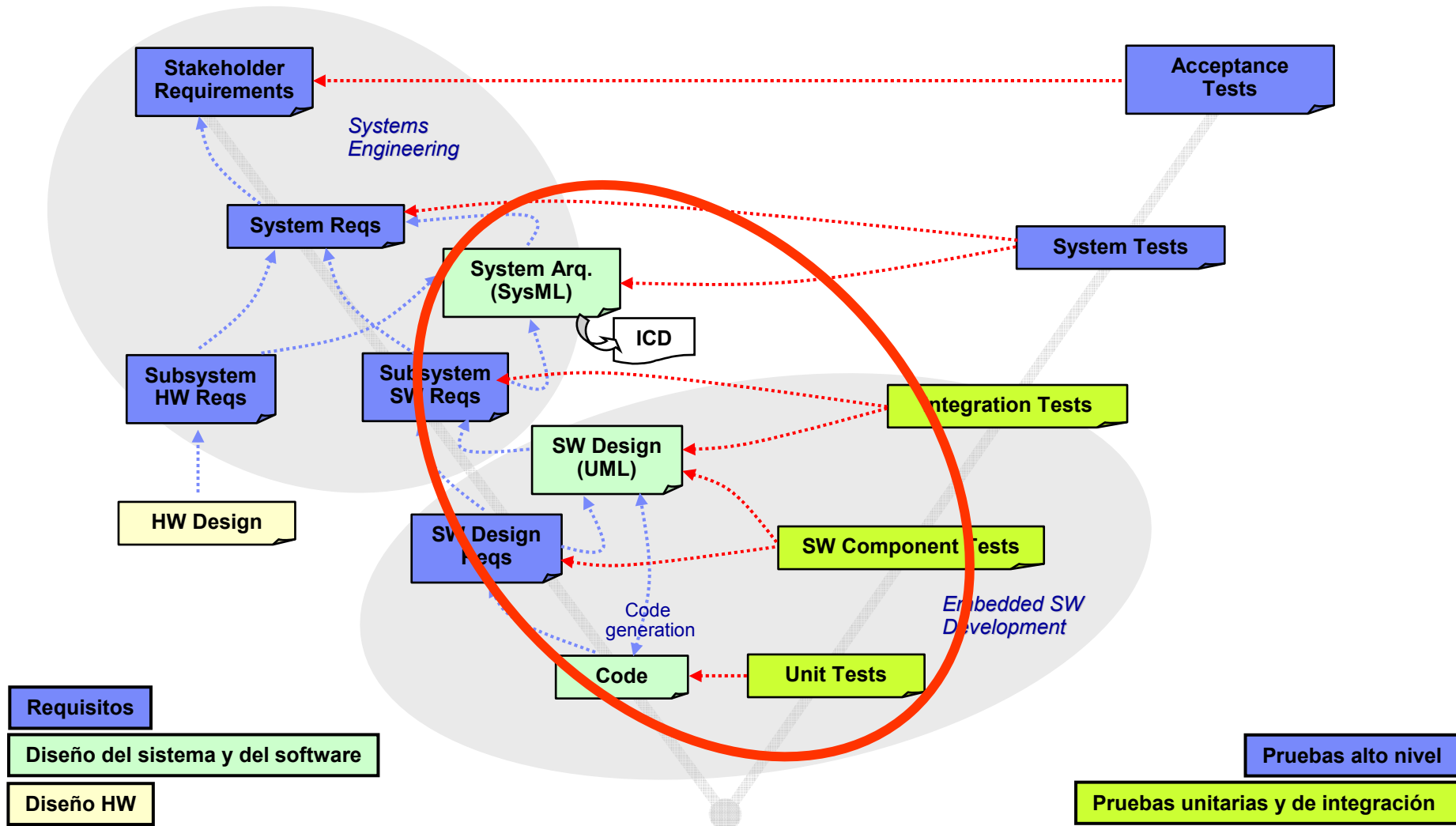


IBM Telelogic Rhapsody®

- **Entorno MDD para el diseño y desarrollo de sistemas**
- **Trazabilidad a los requisitos asegura que el diseño cumple los requisitos del cliente**
- **Incrementa la productividad del desarrollador.** Simulación del sistema mediante ejecución de modelos. La verificación temprana y continua ayuda a detectar errores en el momento de introducirse.
- **Mejora la comunicación.** Visualización de los requisitos, la estructura y el comportamiento del sistema a través de UML/SysML
- **Mejora la colaboración** entre equipos de desarrollo distribuidos
- **Generación automática de código** a partir de los modelos
- **Trabajar desde el código o desde el modelo**
- **Incrementa la productividad del equipo de pruebas:** Pruebas dirigidas por el modelo (*Model Driven Testing - MDT*)
- **Generación automática de documentación**



Proceso de desarrollo genérico construcción y desarrollo de sistemas





IBM Software Group

Arquitectura del sistema



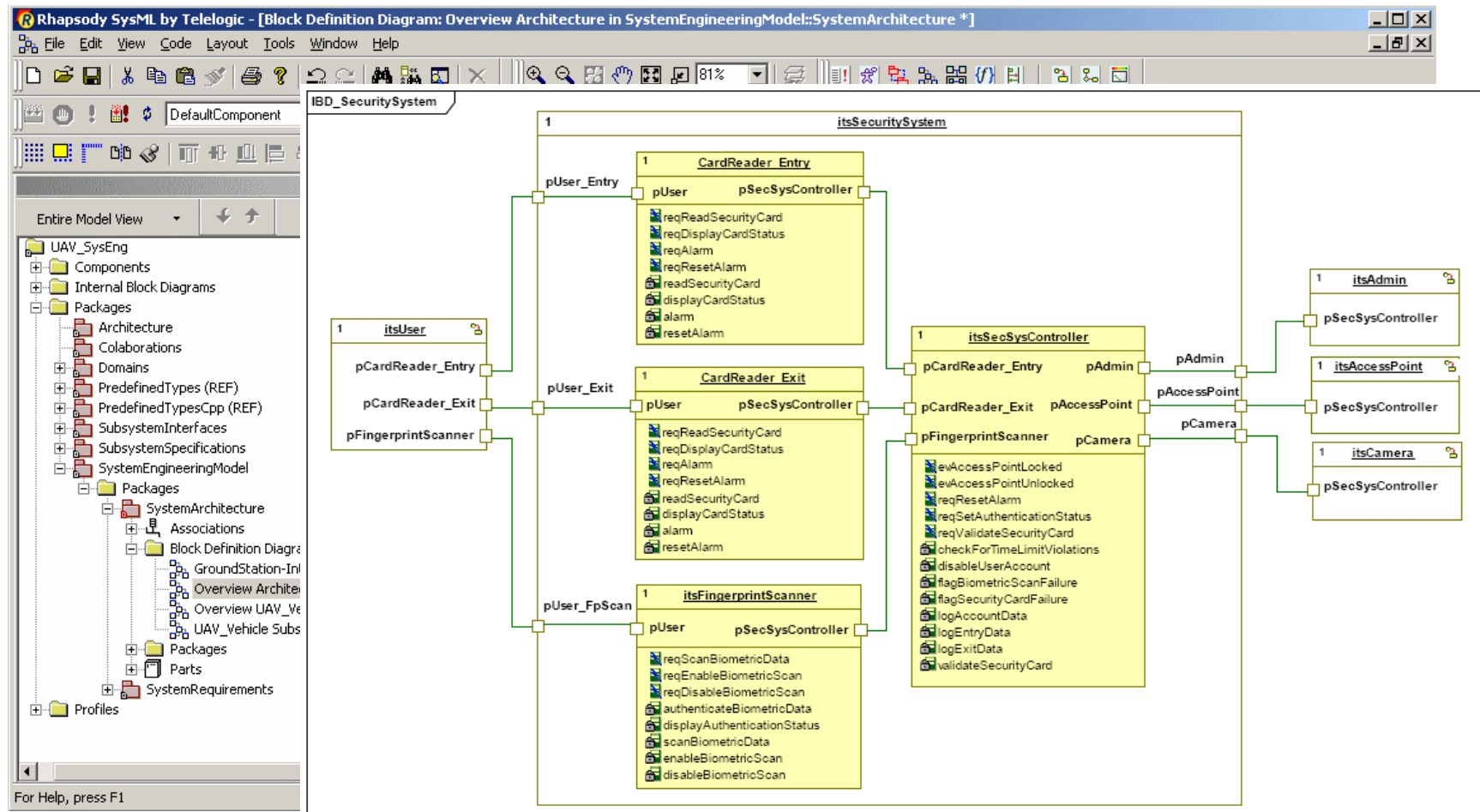
Rational software

→ Go to **IBM**



Diseño de la arquitectura del sistema

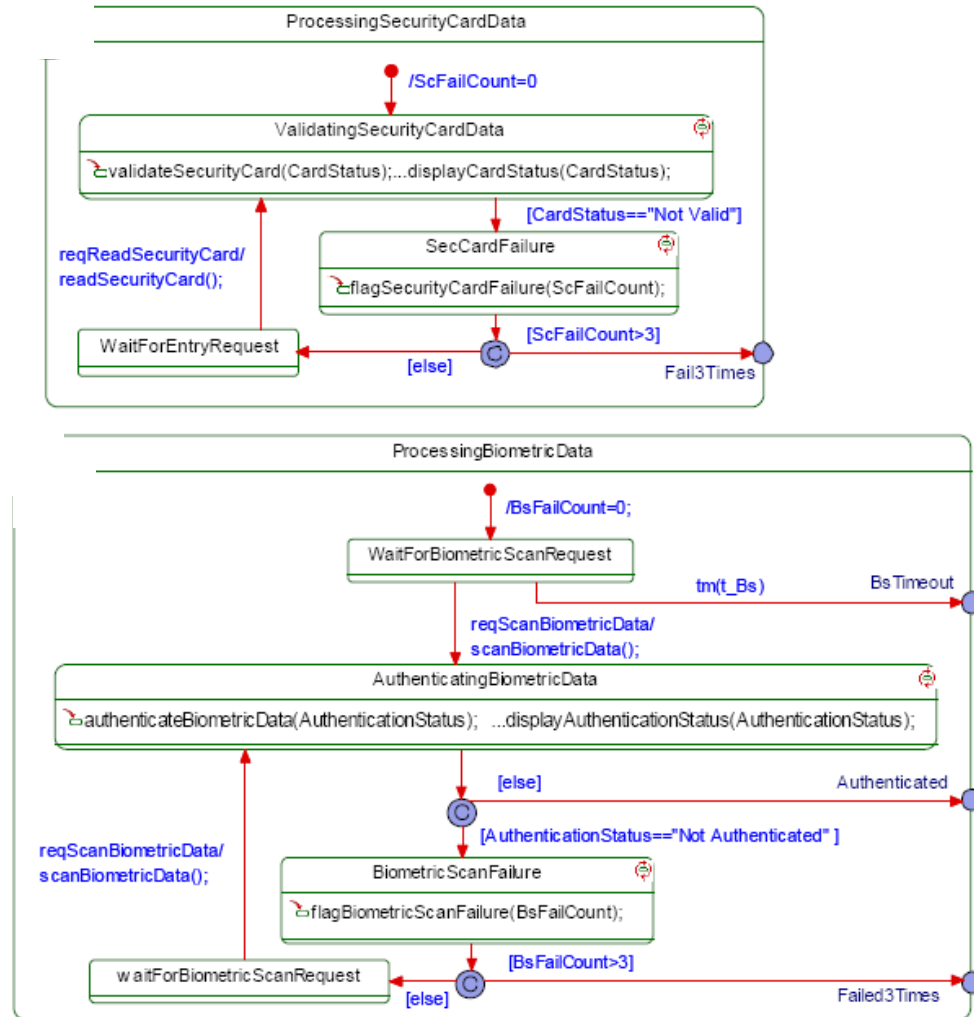
- Estructura: Block Definition Diagrams e Internal Block Diagrams**





Diseño de la arquitectura del sistema

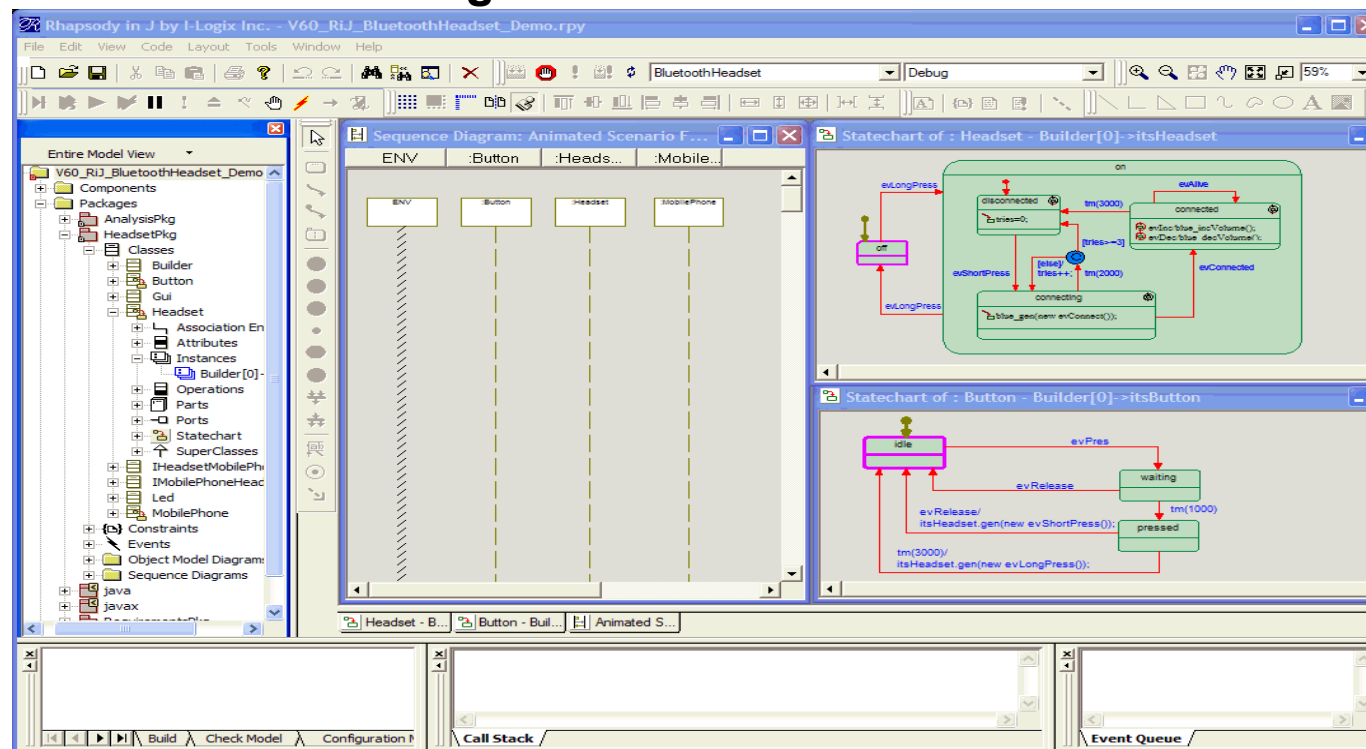
- **Comportamiento:**
 - ▶ Use Case Diagrams
 - ▶ Activity Diagrams
 - ▶ Sequence Diagrams
 - ▶ Statechart Diagrams





Verificación y validación temprana

- La ejecución del modelo ayuda a **detectar errores en las fases tempranas**
- Promueve el desarrollo ágil, permite **validar los requisitos en fases tempranas** de diseño
- Visualización de la ejecución:** Flujo de mensajes en diagrama de secuencia, y estado del sistema en diagrama de estados





| IBM Software Group

Trazabilidad requisitos del sistema y arquitectura

Rational. software

→ Go to **IBM**

Trazabilidad de los requisitos de sistema al modelo

Asegurar que la arquitectura cumple los requisitos del sistema...

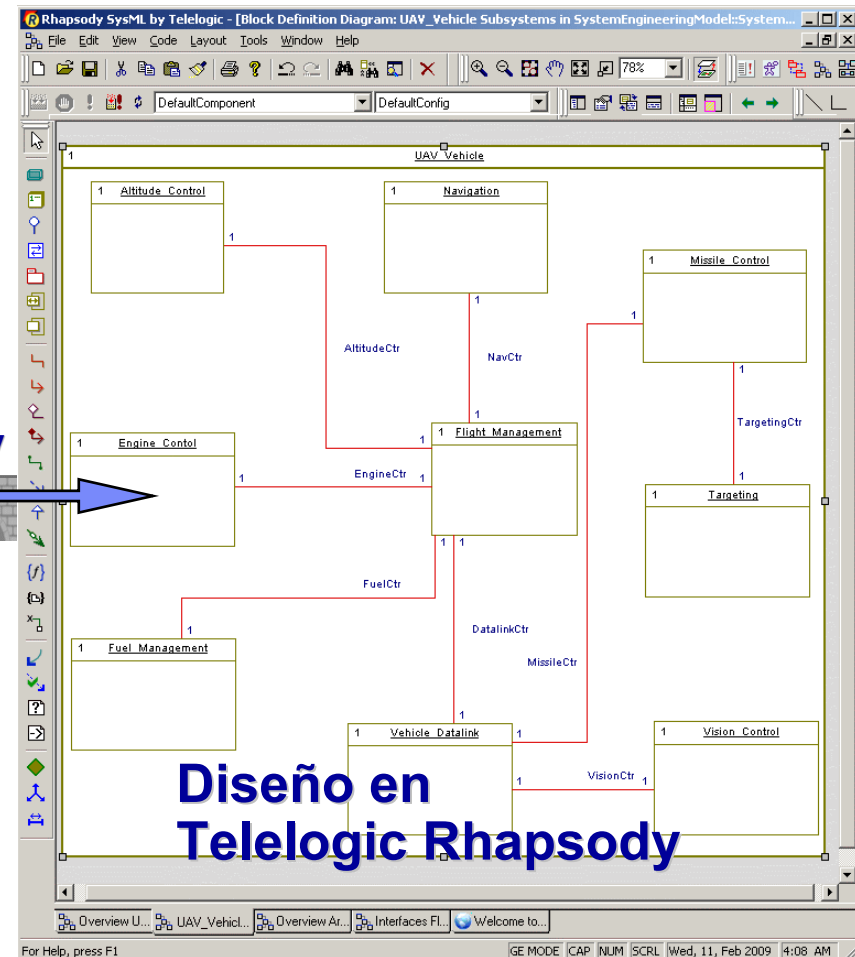
- ¿Cómo se establece la trazabilidad entre requisitos y elementos del diseño?

'System Requirements Specification' current 0.0 in /2. System Requirements Specifications (Formal modu...

ID	Object Type	System Requirements Specification
SRSUAV-1	*	1 Overview
SRSUAV-2	*	The Unmanned Air Vehicle System is a system solution to a medium-range reconnaissance in hostile environments with limited attack capability.
SRSUAV-3	*	It is a medium-range long endurance UAV system that can carry a variety of payloads to assist in ground, air and sea operations.
SRSUAV-4	*	A full UAVS consist of four UAVs and a ground Mission Planning and Control System.
SRSUAV-5	*	2 System Requirements
SRSUAV-74	*	2.1 UAV Vehicle
SRSUAV-6	Requirement	The CUAV is shall be a multipurpose and reusable UAV with multimission capability.
SRSUAV-78	Requirement	This is a new system requirement..
SRSUAV-7	Requirement	It shall operate at altitudes of up to 30,000 feet.
SRSUAV-8	Requirement	It shall reach ground speeds of up to 100 knots in cruise mode.
SRSUAV-9	Requirement	It shall reach ground speeds of up to 150 knots in dash mode.
SRSUAV-10	Requirement	It shall carry payloads up to 450 lbs for durations exceeding 24 hours.
SRSUAV-11	Requirement	The UAV shall fly unimpeded in low visibility environments while carrying reconnaissance or attack payloads.
SRSUAV-12	Requirement	The UAV shall be controllable from the ground station CMPCS.
SRSUAV-13	Requirement	The UAV shall be capable of flying complex flight plans with the operational goal of systematic area search.
SRSUAV-14	Requirement	The UAV shall be capable of flying complex flight plans with the operational goal of ground route or road based (synonym) search.
SRSUAV-15	Requirement	The UAV shall be capable of flying complex flight plans with the operational goal of orbit surveillance of point targets.
SRSUAV-16	Requirement	The UAV and manned control capability from the ground station shall provide sustained 24 hour flight with real-time visual, infra-red or radar telemetry with target recognition preprocessing.
SRSUAV-17	Requirement	The Communications shall be jam resistant in environments that are not high ECM.
SRSUAV-18	Requirement	Control commands shall be encrypted.
SRSUAV-19	Requirement	Control commands shall be processed and un-protected.
SRSUAV-20	Requirement	Telemetry rates for visual telemetry shall support 30 frames-per-second at 640 x 480 resolution.
SRSUAV-21	Requirement	Range of flight shall be fully supported within line of sight (LOS) range and be considerably less than vis. This is because the Coyote is capable of being passed among different CMPCSs.
SRSUAV-22	Requirement	For navigation the Coyote has on-board global positioning system based navigation and shall be directly controllable from the ground station.

Username: Administrator | Exclusive edit mode

¿Cómo?
Gateway



Requisitos en Telelogic DOORS

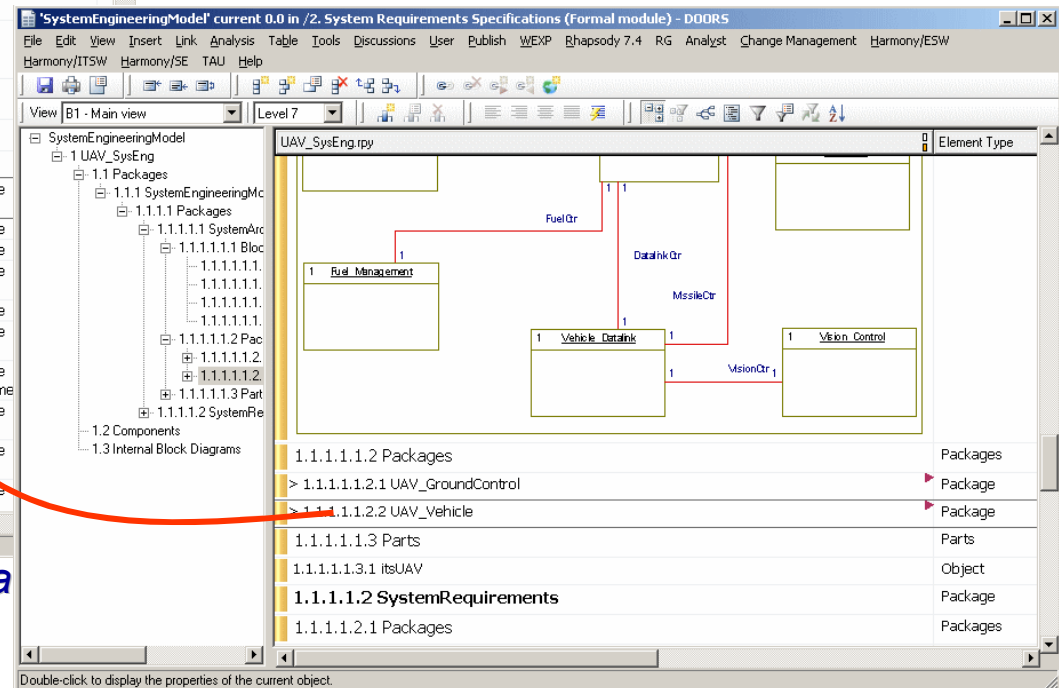
Trazabilidad de los requisitos de sistema al modelo

Asegurar que la arquitectura cumple los requisitos del sistema...

- Las dependencias pueden establecerse desde DOORS

ID	Object Type	System Requirements Specification	Allocated to (System Architecture)
SRSUAV-3	*	It is a medium-range long endurance UAV system that can carry a variety of payloads to assist in ground, air and sea operations.	
SRSUAV-4	*	A full UAVs consist of four UAVs and a ground Mission Planning and Control System.	
SRSUAV-5	*	2 System Requirements	
SRSUAV-74	*	2.1 UAV Vehicle	
SRSUAV-6	Requirement	The CUAV is shall be a multipurpose and reusable UAV with multimission capability.	Package: UAV_Vehicle
SRSUAV-78	Requirement	This is a new system requirement.	Package: UAV_Vehicle
SRSUAV-7	Requirement	It shall operate at altitudes of up to 30,000 feet.	Package: UAV_Vehicle
SRSUAV-8	Requirement	It shall reach ground speeds of up to 100 knots in cruise mode.	Package: UAV_Vehicle
SRSUAV-9	Requirement	It shall reach ground speeds of 150 knots in dash mode.	Package: UAV_Vehicle
SRSUAV-10	Requirement	It shall carry payloads up to 450 lbs for durations exceeding 24 hours.	Package: UAV_Vehicle
SRSUAV-11	Requirement	The UAV shall fly unimpeded in low visibility environments while carrying reconnaissance or attack payloads.	Package: UAV_Vehicle
SRSUAV-12	Requirement	The UAV shall be controllable from the ground station CMPCS.	Package: UAV_Vehicle
SRSUAV-13	Requirement	The UAV shall be capable of of flying complex flight plans with the operational goal of systematic area search.	Package: UAV_Vehicle
SRSUAV-14	Requirement	The UAV shall be capable of of flying complex flight plans with the operational goal of ground route or road based. (synonym)	Package: UAV_Vehicle

Módulo DOORS con una copia de la información del modelo



Crear relación desde el modelo a los requisitos

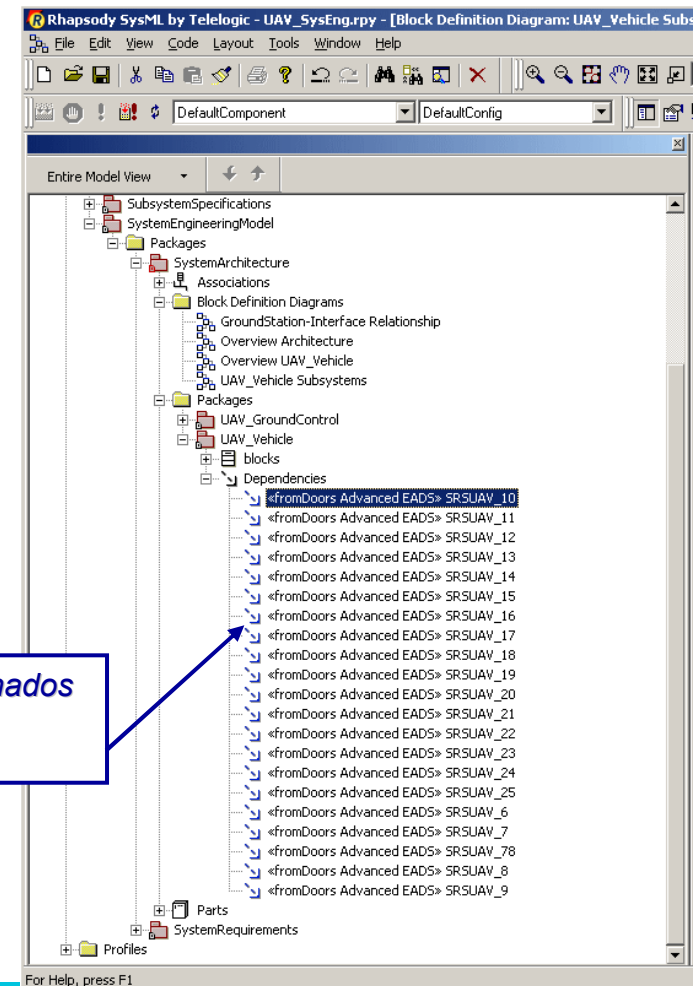
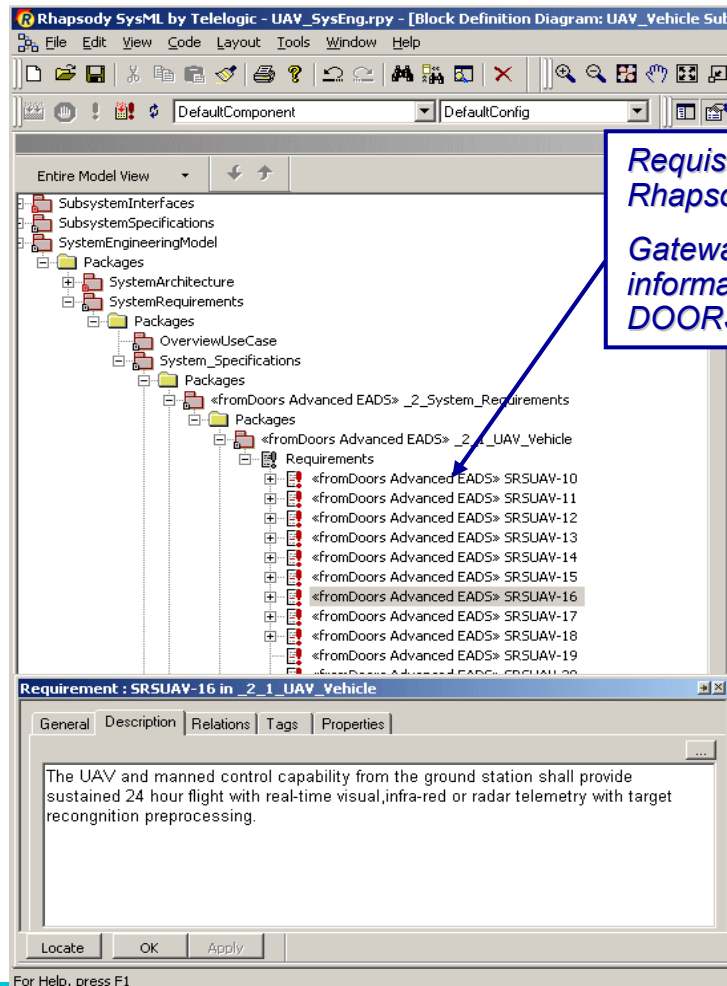
Gateway mantiene actualizada la información del modelo, incluyendo relaciones con reqs



Trazabilidad de los requisitos de sistema al modelo

Asegurar que la arquitectura cumple los requisitos del sistema...

- ...o desde Rhapsody



Trazabilidad de los requisitos de sistema al modelo

Asegurar que la arquitectura cumple los requisitos del sistema...

- Impacto y cobertura desde Gateway

The screenshot displays the Telelogic Rhapsody Gateway - UAV_SysEng interface. The window title is "Telelogic Rhapsody Gateway - UAV_SysEng". The menu bar includes File, Edit, View, Tools, Reports, and Help. The toolbar contains various icons for file operations and analysis. The main workspace is divided into three panes:

- Upstream Impact Information:** A tree view showing a hierarchy of system requirements (SRSUAV-6 to SRSUAV-15) and their associated UAV models (UAV-6 to UAV-15).
- Selection:** A central pane showing a "Rule check" for the "UML Model Rhapsody SysML". It lists "System Specifications" and "Doors Advanced EADS". The selected item is "2.1 UAV Vehicle" under "2 System Requirements".
- Downstream Impact Information:** A tree view showing the downstream impact of the selected requirement, including "UAV_Vehicle" and "Flight_Management" components.

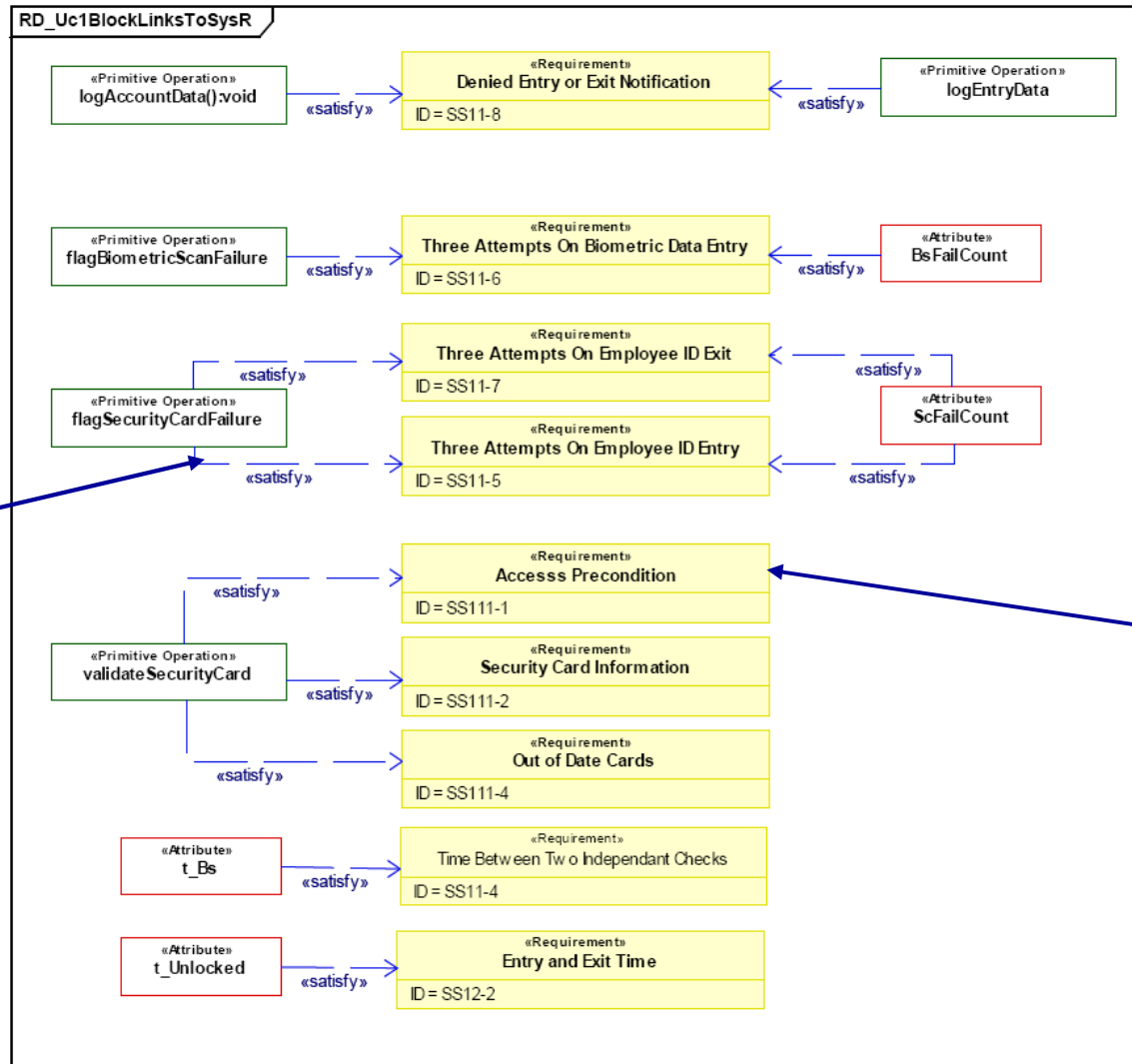
At the bottom of the interface, there are three tabs: "Texts and Reference Attributes", "Attributes", and "Messages". Below these tabs are three input fields for "Upstream", "Selection", and "Downstream" text and reference attributes. The status bar at the bottom indicates the current selection: "System Specifications Doors Advanced EADS/2 System Requirements/2.1 UAV Vehicle".





Visualización de los requisitos y su trazabilidad

Diagramas de Requisitos



Trazabilidad, relación de satisfacción

Requisito





IBM Software Group

Generación automática de documentación: ICDs, ...

Rational software

→ Go to **IBM**

Generación automática de documentación desde el modelo

- **Personalización de informes**
- **Diferentes formatos: HTML, Microsoft® Word®, PowerPoint®, Rich Text Format (RTF), etc. desde un modelo Rhapsody**
 - ▶ Crea enlaces para una navegación rápida del informe
- **Amplia variedad de plantillas predefinidas**
- **Fácil de utilizar**



Generación de documentación: diagramas, descripción textual..

2. Subsystem Interface Description

2.1 Interfaces of first Level of Subsystem division

2.1.1 Interface IDatalink

General Description

Datalink is an High level description of the communication protocol established to communicate the GroundStation and the UAV_Vehicle.

This GroundStation can control 2-5 UAV_Vehicles.

Both sides must implement several variables to make available the status of the datalink.

The physic connection types can change depends on the necessity of the enviroment, need of the war scenario and the strategic situation of the groundstation. This connection types must be able to perform a minimum capacities. These capacities are TBD.

Interface uses diagrams

This Diagram is created to describe the interfaces generated to the communication between the 2 first level Subsystem.

The Datalink Interface reflect the message that could be send/received among the 2 subsystems.

4

Interface : IDatalink in SubsystemInterfaces

General Description Attributes Flow Properties Operations Relations Tags Properties

Datalink is an High level description of the communication protocol established to communicate the GroundStation and the UAV_Vehicle.

This GroundStation can control 2-5 UAV_Vehicles.

Both sides must implement several variables to make available the status of the datalink.

The physic connection types can change depends on the necessity of the enviroment, need of the war scenario and the strategic situation of the groundstation. This connection types must be able to perform a minimum capacities. These capacities are TBD.

Block Definition Diagram : UAV Principal Subsystems in SubsystemInterfaces

General Description Relations Tags Properties

This Diagram is created to describe the interfaces generated to the communication between the 2 first level Subsystem.

The Datalink Interface reflect the message that could be send/received among the 2 subsystems.

Interface : IDatalink in SubsystemInterfaces

General Description Attributes Flow Properties Operations Relations Tags Properties

Name: IDatalink

Stereotype:

Main Diagram: UAV Principal Subsystems in SubsystemInterfaces

Concurrency: sequential

Defined In : SubsystemInterfaces



Generación de documentación: mensajes en tablas, tipos de datos...

2.1.1.1 Attributes in the interface IDatalink

Name	Type	Range
LinkType	LinkType	See LinkType
LinkStatus	LinkStatusType	See LinkStatusType
LinkSpeed	int	9800..34000000
CongestionFlag	int	0..100

2.1.1.2 Messages in the Interface IDatalink

In this section will be described the messages that each subsystem must implement to communicate with the rest of the Subsystems that expect to use this interface to communicate.

2.1.1.2.1 Message: InitComms

Description:
Initialize the communications with the requested address.

Fields in this message

Name	Type	Range	Length
OrgAddress	int	0..2e32	32 bits
OrgPort	int	0..2e32	32 bits
DestAddress	int	0..2e32	32 bits
DestPort	int	0..2e32	32 bits
Type	LinkType	See LinkType	4 bits
DLOperation	DataLinkOpType	See DataLinkOpType	3 bits

2.1.1.2.2 Message: CloseComms

Description:
Finalize the communications with the requested address.

Fields in this message

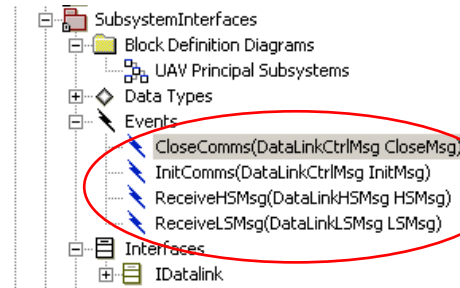
Name	Type	Range	Length
OrgAddress	int	0..2e32	32 bits
OrgPort	int	0..2e32	32 bits
DestAddress	int	0..2e32	32 bits
DestPort	int	0..2e32	32 bits
Type	LinkType	See LinkType	4 bits
DLOperation	DataLinkOpType	See DataLinkOpType	3 bits

2.1.1.2.3 Message: ReceiveHSMsg

Description:
A Message of High Speed and High priority is sent from an origin address-port pair to a destination address-port pair. These pairs represent the 2 subsystems that is connected.

Fields in this message

Name	Type	Range	Length
OrgAddress	int	0..2e32	32 bits
OrgPort	int	0..2e32	32 bits
DestAddress	int	0..2e32	32 bits
DestPort	int	0..2e32	32 bits



Event : CloseComms in SubsystemInterfaces

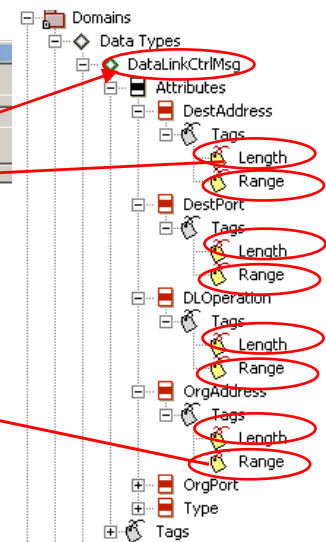
General Arguments Description Relations Tags Properties

Finalize the communications with the requested address.

Event : CloseComms in SubsystemInterfaces

General Arguments Description Relations Tags Properties

Name	Type	Value
CloseMsg	DataLinkCtrlMsg	<New>





IBM Software Group

Diseño y desarrollo de software



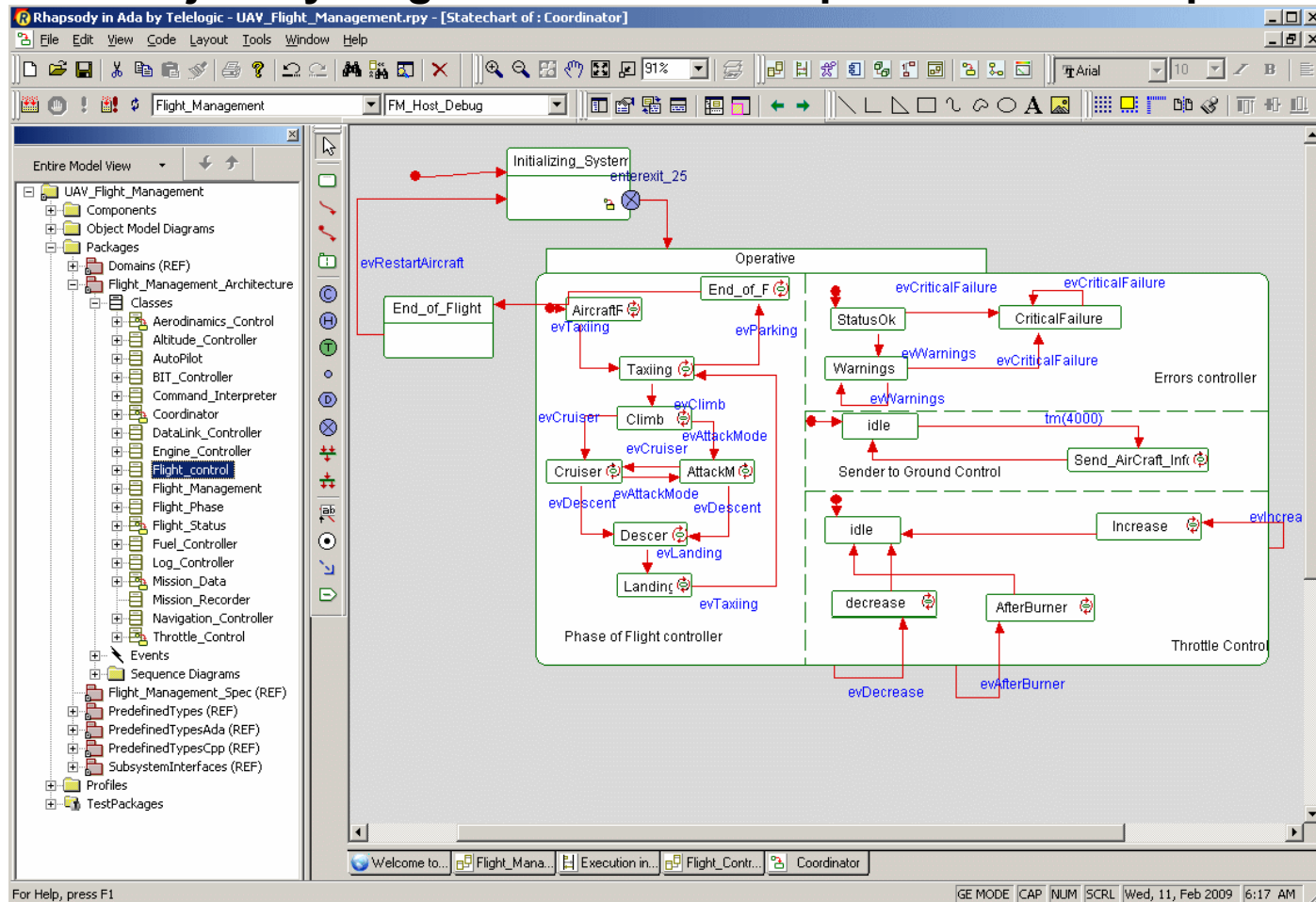
Rational software

→ Go to IBM



Diseño del software

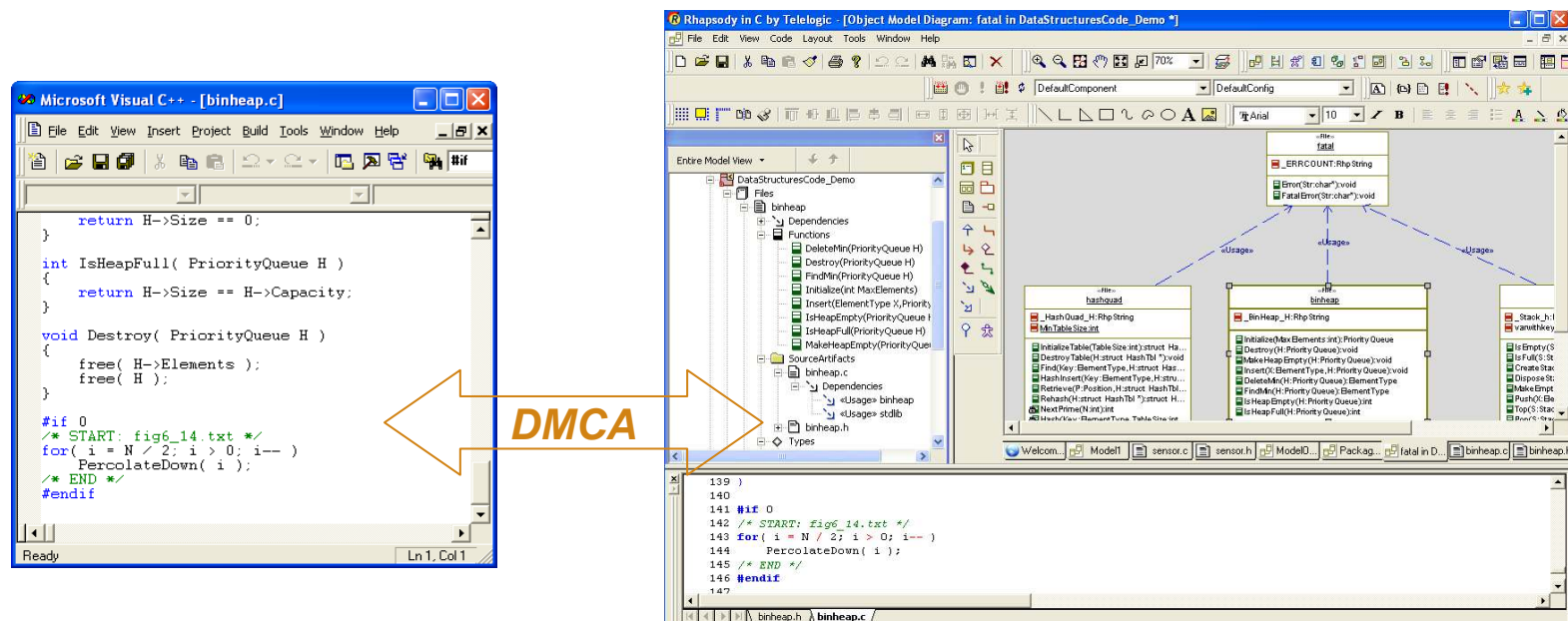
- Diagrama de objetos y diagrama de estados que define el comportamiento



Trabaje como usted quiera

Dynamic Model Code Associativity - DMCA

- **A nivel del modelo o el código**
 - ▶ Reduce la curva de aprendizaje
 - ▶ Incremento de la productividad
- **Diseño y código se mantienen sincronizados: “dynamic model code associativity” (DMCA)**
 - ▶ Cambie el modelo y el código cambiará automáticamente
 - ▶ Cambie el código y el modelo cambiará automáticamente



Si no quiere, no es necesario codificar...

- Añadir detalle al modelo a través de formularios

The screenshot displays the Rhapsody in Ada IDE interface. On the left, the 'Entire Model View' shows a hierarchical tree of components. Under 'Throttle_Control', the 'Attributes' folder is expanded, showing a 'currentPower' attribute. On the right, the code editor shows the Ada source code for 'Throttle_Control.ads'. The code defines a 'reactive_part_t' type and a 'Throttle_Control_t' tagged record. The 'currentPower' attribute is defined as a 'Float := 0.0;' with a comment '--++ attribute currentPower'. A red circle highlights this line, and a red arrow points from the 'currentPower' attribute in the model tree to this line in the code editor. The status bar at the bottom indicates 'GE MODE', 'NUM', and the date 'Wed, 11, Feb 2009 6:29 AM'.

```
type reactive_part_t is new Oxf.Reactive.Reactive_t with
record
  its_Parent : Throttle_Control_wide_acc_t;
end record;

end reactive_part;

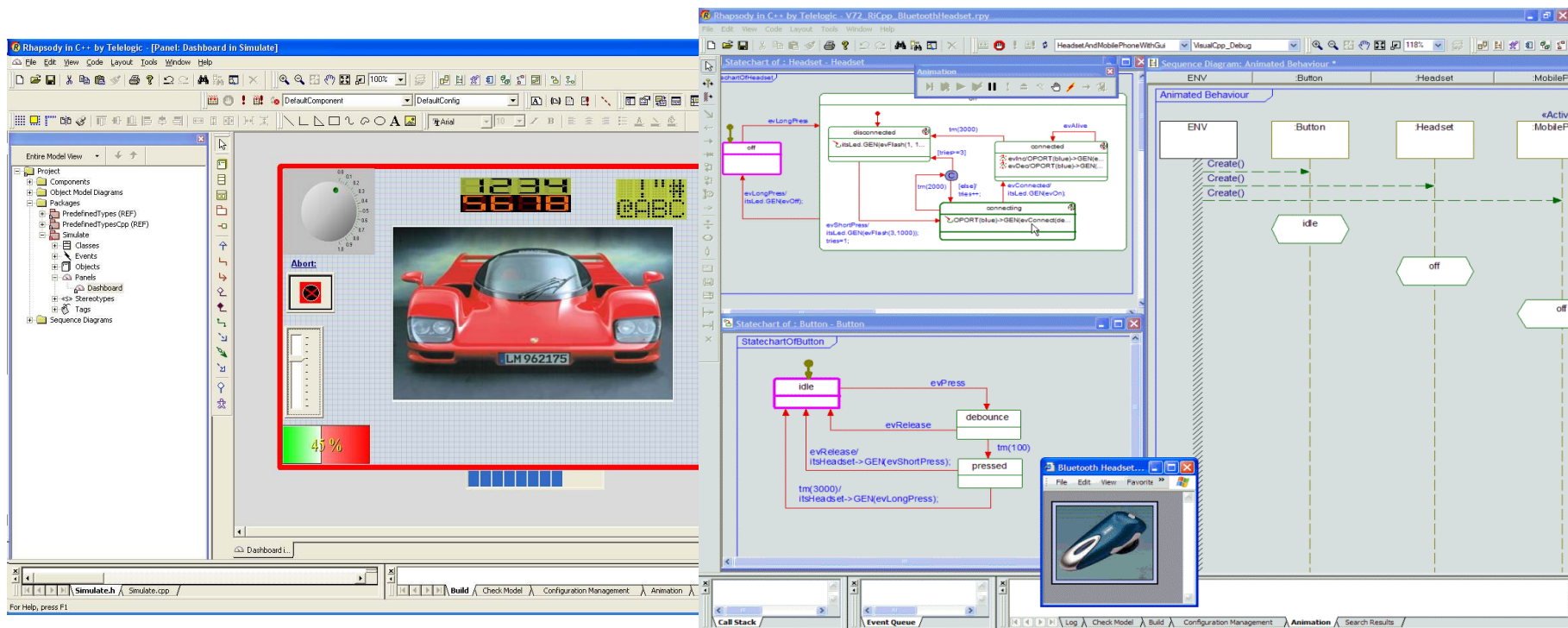
type Throttle_Control_t is tagged
record
  its_Reactive : aliased reactive_part.reactive_part_t;
  its_Reactive_acc : reactive_part.reactive_part_acc_t;
  Operative_Sub_State : Integer := Non_State;
  root_state_Sub_State : Integer := Non_State;

  -- Fields
  currentPower : Float := 0.0; --++ attribute currentPower
end record;

procedure Start_Behavior (this : access Throttle_Control_t;
success : out boolean
);
procedure Terminate_Behavior (this : access Throttle_Control_t);
```

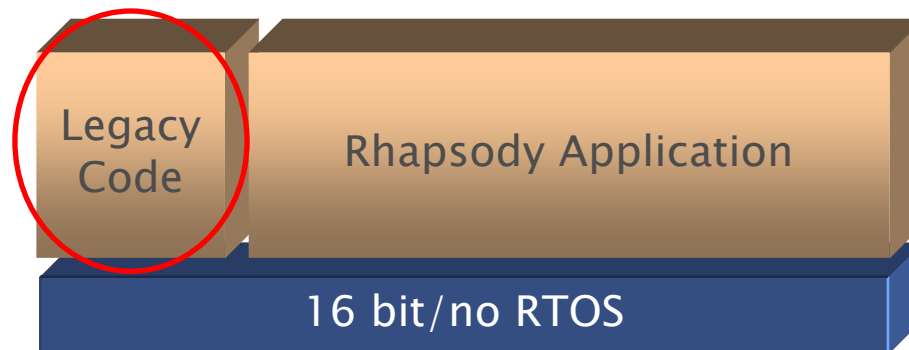
Paneles gráficos para la simulación y animación

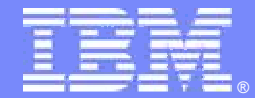
- Creación de **maquetas de la interfaz** para comunicar eficazmente el comportamiento de diseño destinado a los clientes
- Modificar, monitorizar y analizar los valores de datos durante la simulación para ayudar a garantizar que el **diseño es correcto** en una fase temprana del proceso



Generación automática de código desde el modelo

- **Generar código completo de aplicaciones C, C++, Java y Ada**
- **Desplegar rápidamente su diseño en cualquier plataforma de destino.**
- **Reutilizar código externo, referenciado en el modelo o integrado (ingeniería inversa)**





IBM Software Group

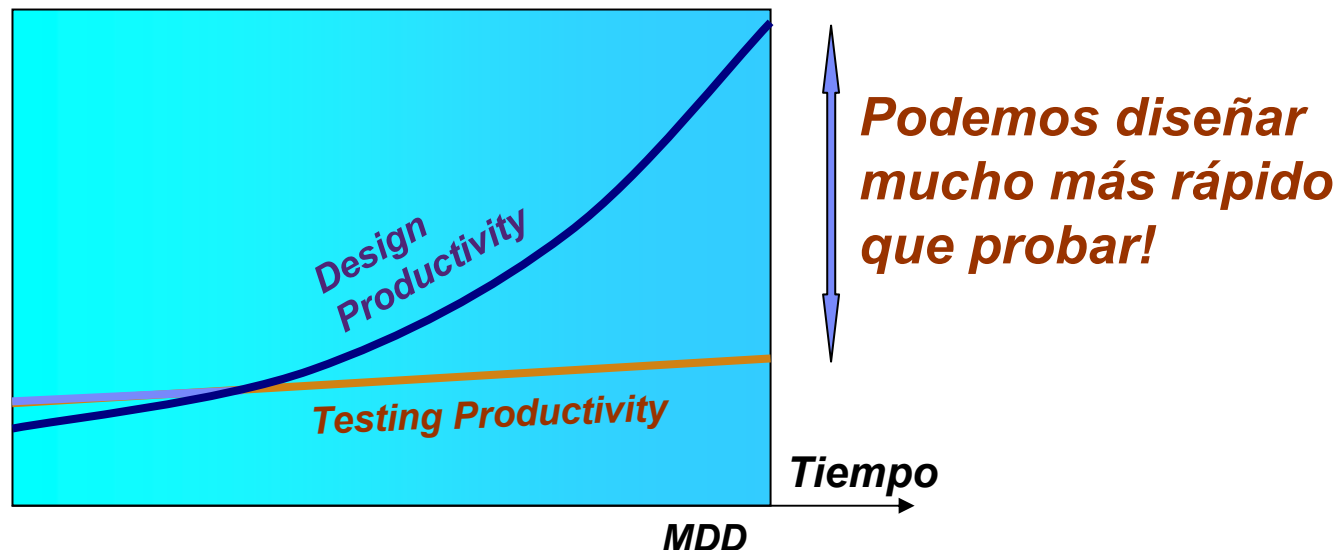
*Pruebas dirigidas por modelos
(Model Driven Testing – MDT)*

Rational software

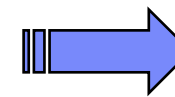
→ Go to **IBM**

Proceso de pruebas poco productivo...

- Está comprobado que la adopción de MDD mejora la productividad de los desarrolladores
- El equipo de pruebas sigue siendo mucho menos productivo que el de desarrollo: pruebas a nivel de código!!!



- ¿Por qué no trasladar la idea de MDD a Testing?



**Model
Driven
Testing**



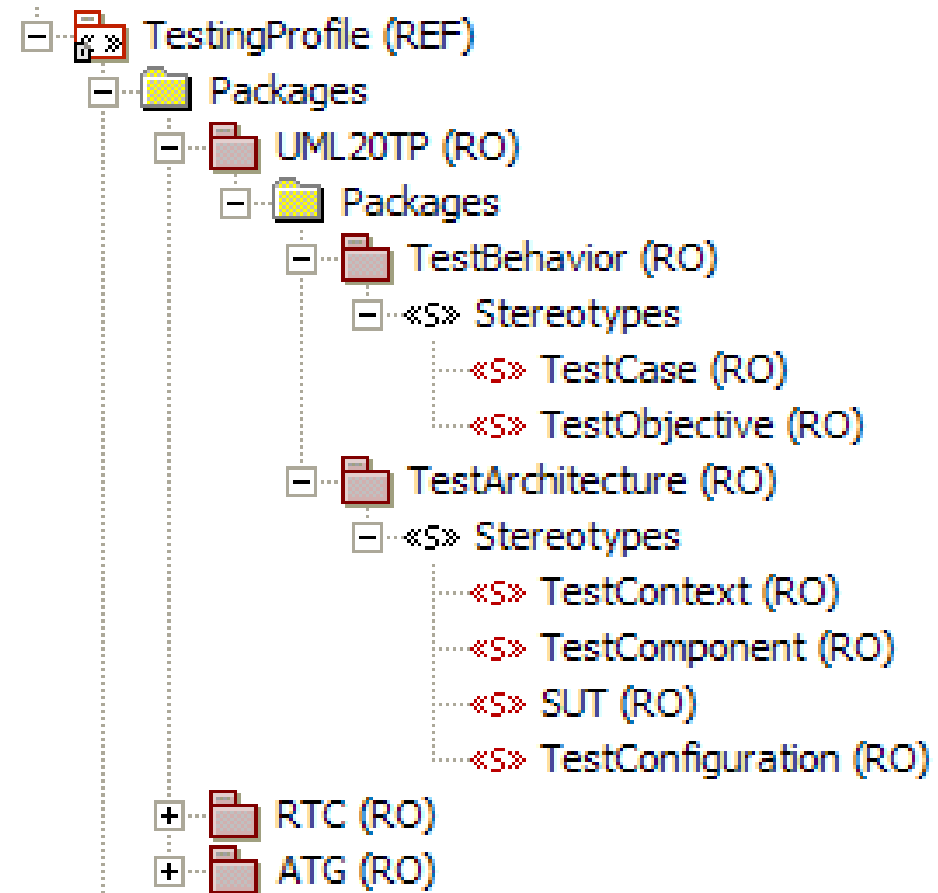
Pruebas dirigidas por modelos (Model Driven Testing - MDT)

- Wikipedia, la enciclopedia libre en la World Wide Web (www), se refiere a *model driven testing* como “*software testing where test cases are derived in whole or in part from a model that describes some (if not all) aspects of the system under test (SUT)*”
- UML es el lenguaje para modelado, mientras que los tests derivados del modelo se representan utilizando **UML Testing Profile (UTP)**
- UTP se puede considerar como un lenguaje para **visualizar, especificar, analizar, construir** y **documentar** todos los artefactos involucrados en un proceso de pruebas.



UML Testing Profile

- **El UML 2 Testing Profile extiende UML con conceptos específicos de pruebas:**
 - ▶ **Test Behavior** : actividades a realizar durante la prueba y resultados
 - ▶ **Test Architecture** : elementos del modelo a probar y sus relaciones con otro elementos



Procesos de diseño y pruebas completamente integrados

Diseño

- Object Model Diagrams
- Domains Overview
- Packages
 - «External» CppLibrary
 - «External» RhapsodyFramework
 - AnalysisPkg
 - CashRegisterPkg
 - Classes
 - BuyOneGetOneFree
 - BuyThreeGetOneFree
 - CashRegister
 - CountedProduct
 - Product
 - ProductDatabase
 - TenPercentOff
 - ThreeForOneEuro
 - Dependencies
 - Interfaces
 - Object Model Diagrams
 - CashRegister Overview
 - Product Overview
 - ProductDatabase Overview
 - Special Offers Overview
 - Sequence Diagrams
 - HardwarePkg
 - InterfacesPkg
 - PredefinedTypes (REF)
 - PredefinedTypesCpp (REF)
 - RequirementsPkg
 - TestConductorPkg
 - Profiles
 - TestPackages

Pruebas

- CashRegister
 - Profiles
 - TestPackages
 - demo
 - TPkg_CashRegister_0
 - Packages
 - TestComponents
 - TestContexts
 - TCon_CashRegister
 - Attributes
 - Dependencies
 - Links
 - SUTs
 - Test Context Diagrams
 - TestCases
 - atg_tc_002()
 - atg_tc_003()
 - atg_tc_004()
 - atg_tc_006()
 - atg_tc_008()
 - atg_tc_009()
 - tc_activity_diagram()
 - tc_adding_removing_products()
 - tc_code()
 - tc_regression_test()
 - TestComponentInstances
 - TestConfigurations
 - TestResults
 - TCon_CashRegister_1.html
 - TCon_CashRegister_1.xml
 - TestScenarios

Test Context Result

Informe de resultados

Environment Info	
Test executed on machine:	NBOSC-21-1
Test executed by user:	ubrockmeyer
Used OS version:	Windows 2000 / Windows XP
Used Rhapsody version:	Aries, build 799102
Used TestConductor version:	2.0, build 530

Tested Project	
Project:	CashRegister
Active Component:	TCon_CashRegister_5
Active Configuration:	DefaultConfig

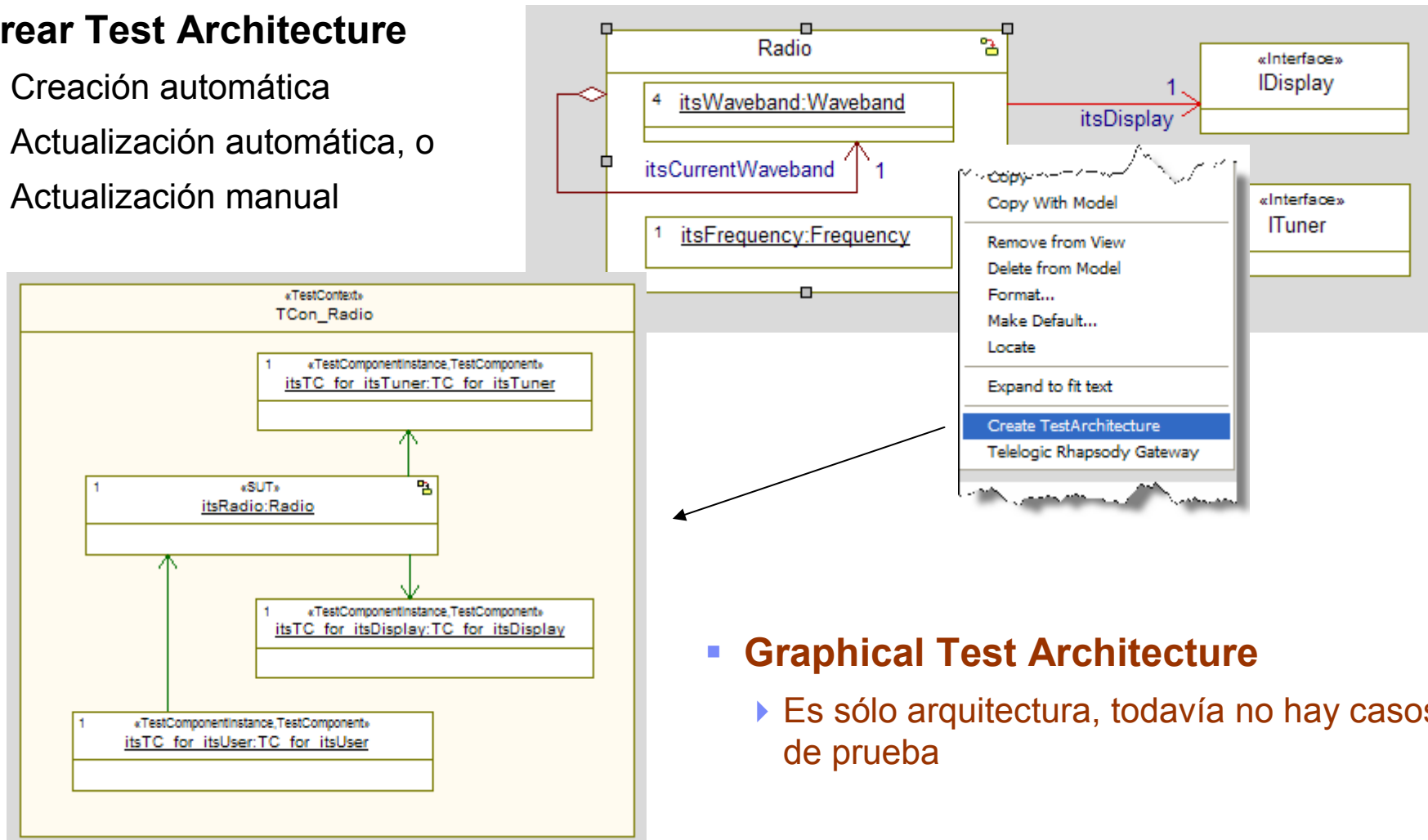
Test Context: TCon_CashRegister	Summary: PASSED
tc_code	PASSED
tc_activity_diagram	PASSED
tc_adding_removing_products	PASSED
tc_regression_test	PASSED
atg_tc_008	PASSED
atg_tc_009	PASSED
atg_tc_006	PASSED
atg_tc_002	PASSED
atg_tc_003	PASSED
atg_tc_004	PASSED

- Fácil navegación entre el diseño y los artefactos de prueba, son parte del mismo modelo
- Trazabilidad de requisitos a test cases
- Diseño y pruebas siempre sincronizados y actualizados
- Generación automática de informes de resultados

Proceso de pruebas: Crear arquitectura de pruebas

■ Crear Test Architecture

- ▶ Creación automática
- ▶ Actualización automática, o
- ▶ Actualización manual



■ Graphical Test Architecture

- ▶ Es sólo arquitectura, todavía no hay casos de prueba



Proceso de pruebas: creación de casos de prueba

- Crear Test Architecture

- Casos de prueba como diag de secuencia

- ▶ Manualmente

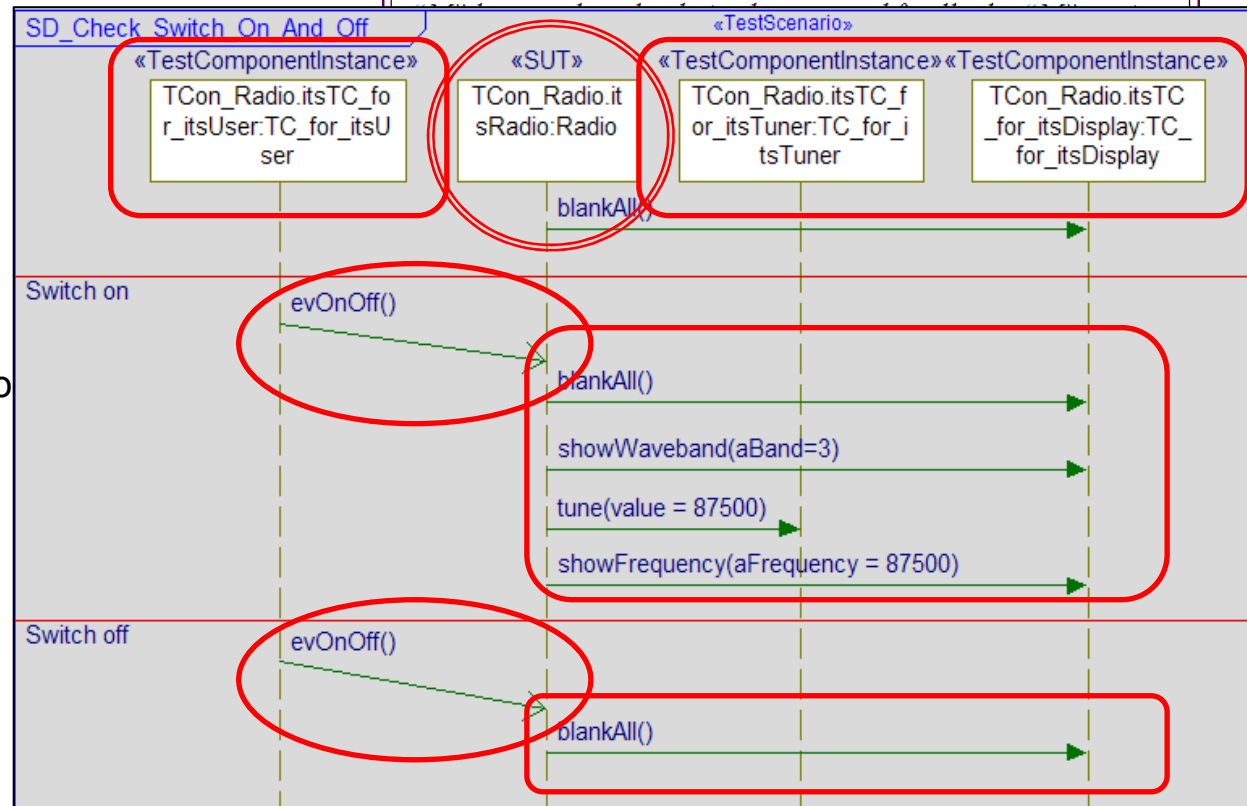
- En base a

- Requisitos Textuales
 - Experiencia previa o en el dominio

- Se define el comportamiento esperado del elemento a probar o SuT (System under Test)

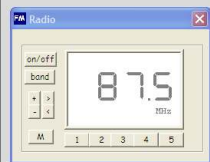
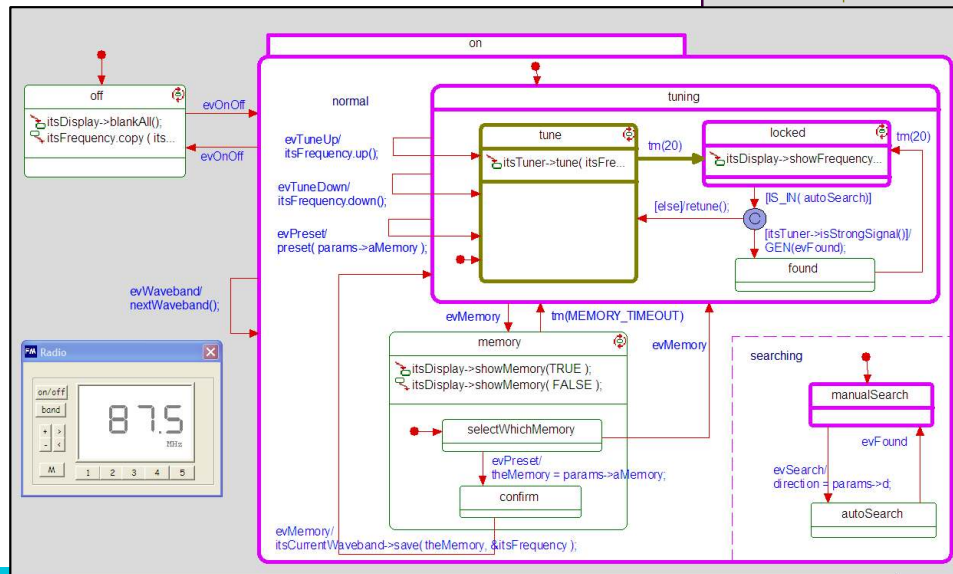
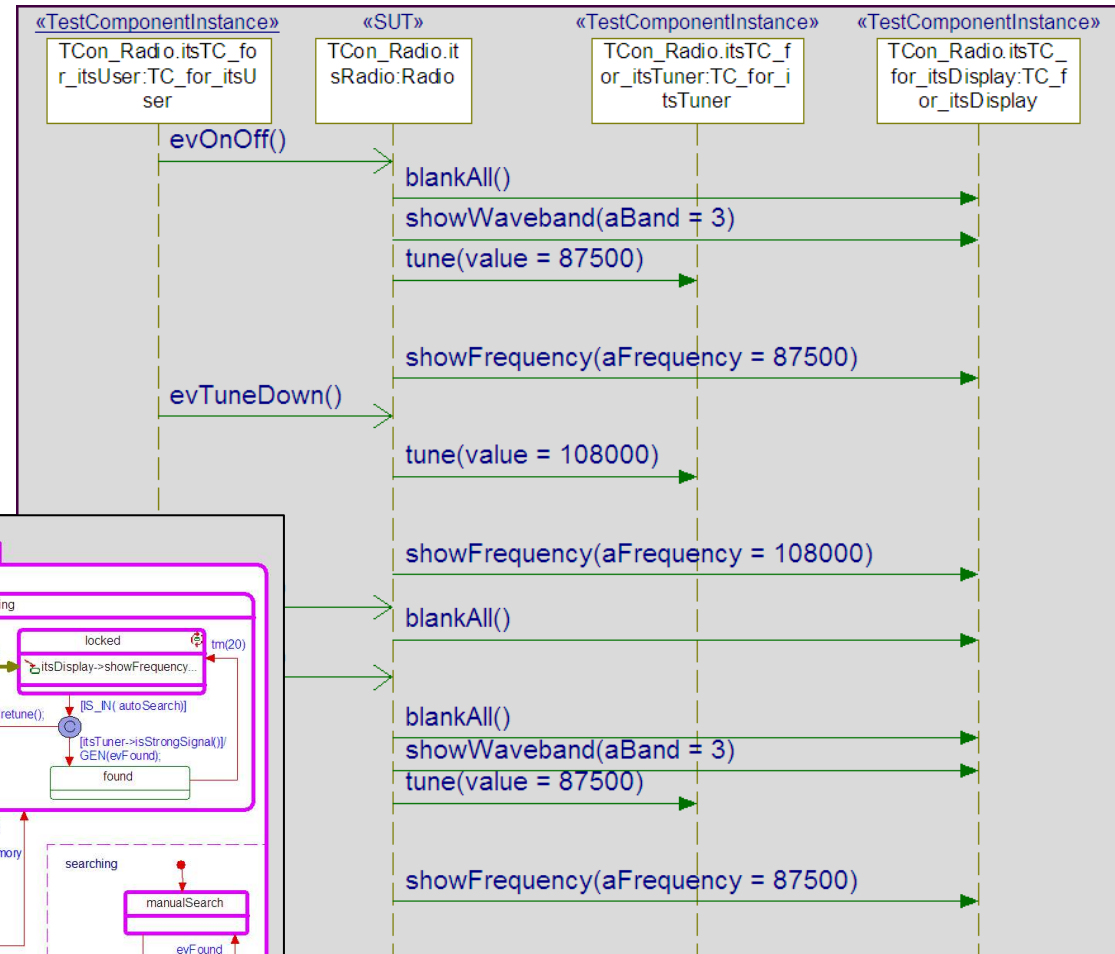
REQ014: There are 5 memory preset buttons. When pressed, they recall a frequency. This frequency depends on the selected waveband. Thus it is possible to have 20 frequencies memorized, five for each waveband.

REQ015: There is an "M" button allowing the presets to be set to the current frequency. This is done by pressing the



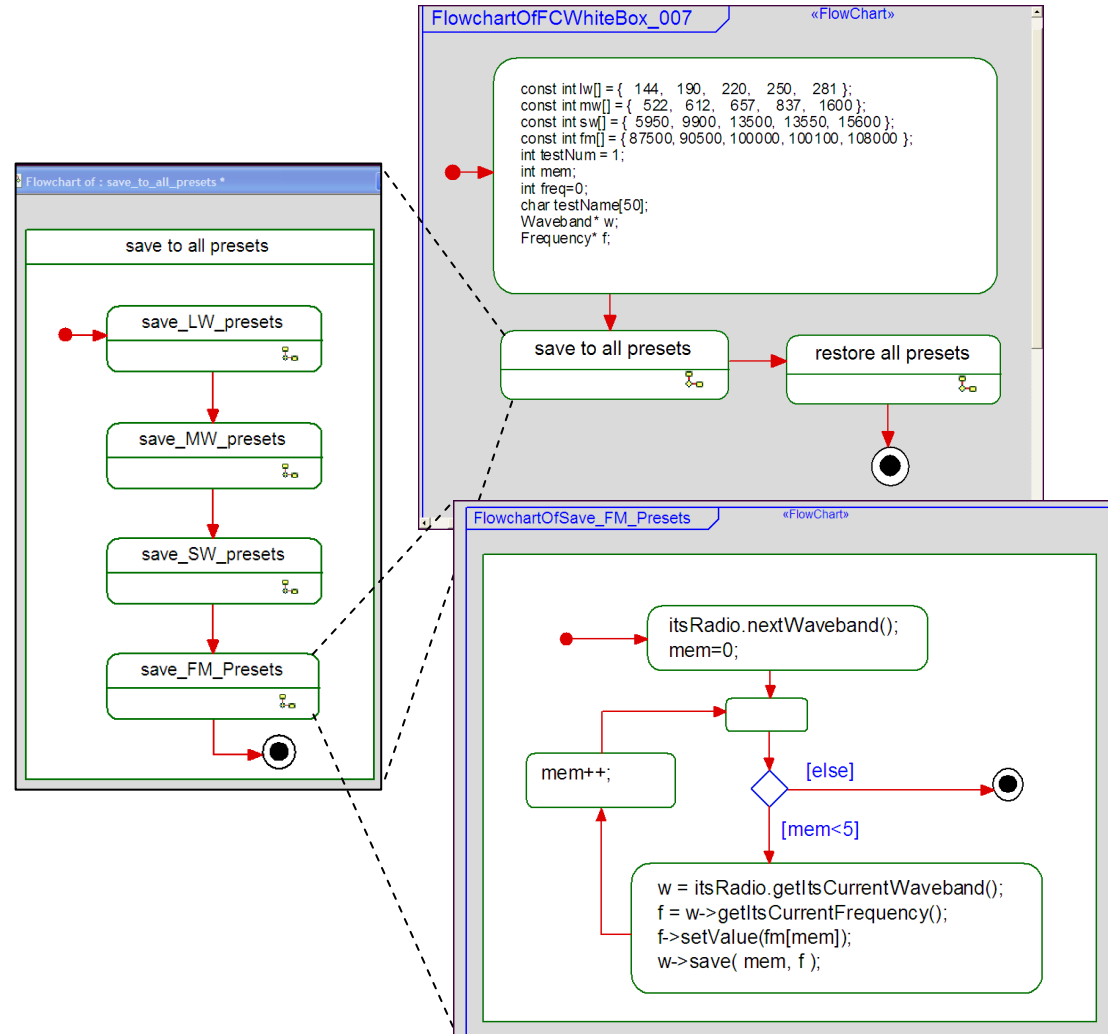
Proceso de pruebas: creación de casos de prueba

- Crear Test Architecture
- Casos de prueba como diag de secuencia
 - ▶ Manualmente
 - ▶ Grabación de simulaciones (diagramas de secuencia animados)



Proceso de pruebas: creación de casos de prueba

- **Crear a Test Architecture**
- **Casos de prueba como diag de secuencia**
 - ▶ Manualmente
 - ▶ Grabación de simulaciones
- **Casos de prueba como diagramas de flujo**
 - ▶ “Programación gráfica”
 - ▶ Intuitivos, potentes y fáciles de usar



Proceso de pruebas: creacion de casos de prueba

- Crear Test Architecture

- Casos de prueba como diag de secuencia

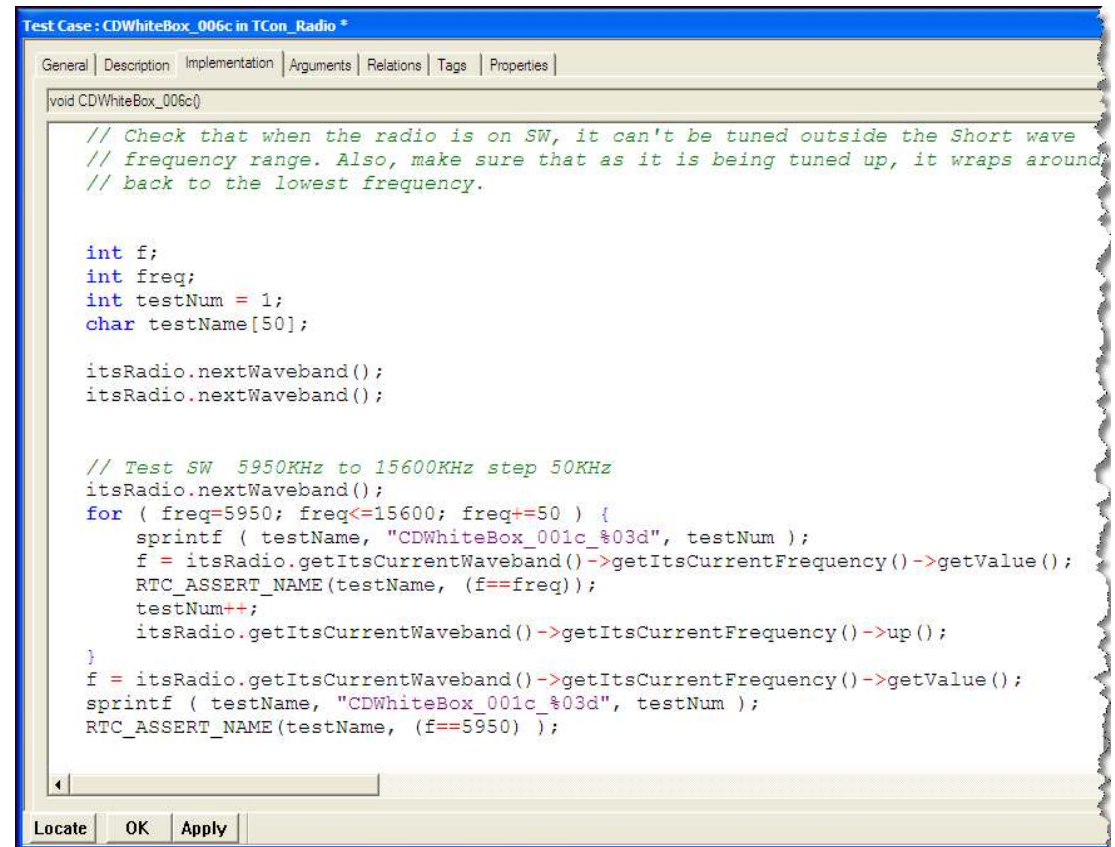
- ▶ Manualmente
- ▶ Grabación de simulaciones

- Casos de prueba como diagramas de flujo

- ▶ “Programación gráfica”
- ▶ Intuitivos, potentes y fáciles de usar

- Casos de prueba como código

- ▶ El comportamiento del caso de prueba se puede introducir directamente codificando



```
Test Case : CDWhiteBox_006c in TCon_Radio *
General | Description | Implementation | Arguments | Relations | Tags | Properties
void CDWhiteBox_006c()
// Check that when the radio is on SW, it can't be tuned outside the Short wave
// frequency range. Also, make sure that as it is being tuned up, it wraps around
// back to the lowest frequency.

int f;
int freq;
int testNum = 1;
char testName[50];

itsRadio.nextWaveband();
itsRadio.nextWaveband();

// Test SW 5950KHz to 15600KHz step 50KHz
itsRadio.nextWaveband();
for ( freq=5950; freq<=15600; freq+=50 ) {
    sprintf ( testName, "CDWhiteBox_001c_%03d", testNum );
    f = itsRadio.getItsCurrentWaveband()->getItsCurrentFrequency()->getValue();
    RTC_ASSERT_NAME(testName, (f==freq));
    testNum++;
    itsRadio.getItsCurrentWaveband()->getItsCurrentFrequency()->up();
}
f = itsRadio.getItsCurrentWaveband()->getItsCurrentFrequency()->getValue();
sprintf ( testName, "CDWhiteBox_001c_%03d", testNum );
RTC_ASSERT_NAME(testName, (f==5950) );
```



Proceso de pruebas: ejecución y resultados

Execute TestCase

Name	File	Line	Result
CDWhiteBox_001b_114	TCon_Radio.cpp	143	PASSED
CDWhiteBox_001b_115	TCon_Radio.cpp	143	PASSED
CDWhiteBox_001b_116	TCon_Radio.cpp	143	PASSED
CDWhiteBox_001b_117	TCon_Radio.cpp	143	PASSED
CDWhiteBox_001b_118	TCon_Radio.cpp	143	PASSED
CDWhiteBox_001b_119	TCon_Radio.cpp	143	PASSED
CDWhiteBox_001b_120	TCon_Radio.cpp	143	PASSED
CDWhiteBox_001b_121	TCon_Radio.cpp	149	FAILED

Buttons: Activate, Show Assertion, Quit

Casos de prueba como un flujo de flujo

Casos de prueba como código

Ejecución e informe de resultados

- Comportamientos inesperados se resaltan

Table of Contents:

- Test Report of Model V71_RICpp_Radio
 - TCon_Radio
 - System Under Test (SUT)
 - Test Component Instances
 - Test Context Diagrams
 - Test Cases
 - Test Case CDWhiteBox_006a
 - Test Objectives
 - Test Objective WB_TST006
 - Test Case SDWhiteBox_006b
 - Test Objectives
 - Test Case CDWhiteBox_006c
 - Test Objectives
 - Test Case CDWhiteBox_006d
 - Test Objectives
 - Test Case FCWhiteBox_007
 - Test Objectives
 - Test Case SDWhiteBox_001
 - Test Scenario check radio can be
 - Test Objective WB_TST007
 - Test Case SDWhiteBox_002a
 - Test Objectives
 - Test Case SDWhiteBox_002b
 - Test Objectives
 - Test Case SDWhiteBox_003
 - Test Objectives
 - Test Case SDWhiteBox_004
 - Test Objectives

Test Context Result

Test Context: TCon_Radio

Fri Sep 21 22:01:37 2007

Environment Info	
Test executed on machine:	TEMPRANILLO
Test executed by user:	ukmari
Used OS version:	Windows 2000 / Windows XP
Used Rhapsody version:	7.1, build896637
Used TestConductor version:	2.0, build 646

Tested Project	
Project:	V71_RICpp_Radio
Active Component:	TPkg_Radio_Comp
Active Configuration:	TC

Test Context: TCon_Radio	Summary: FAILED
CDWhiteBox_006a	PASSED
CDWhiteBox_006b	PASSED
CDWhiteBox_006c	PASSED
CDWhiteBox_006d	PASSED
FCWhiteBox_007	PASSED
SDWhiteBox_001	PASSED
SDWhiteBox_002a	PASSED
SDWhiteBox_002b	FAILED
SDWhiteBox_003	PASSED
SDWhiteBox_004	PASSED





| IBM Software Group

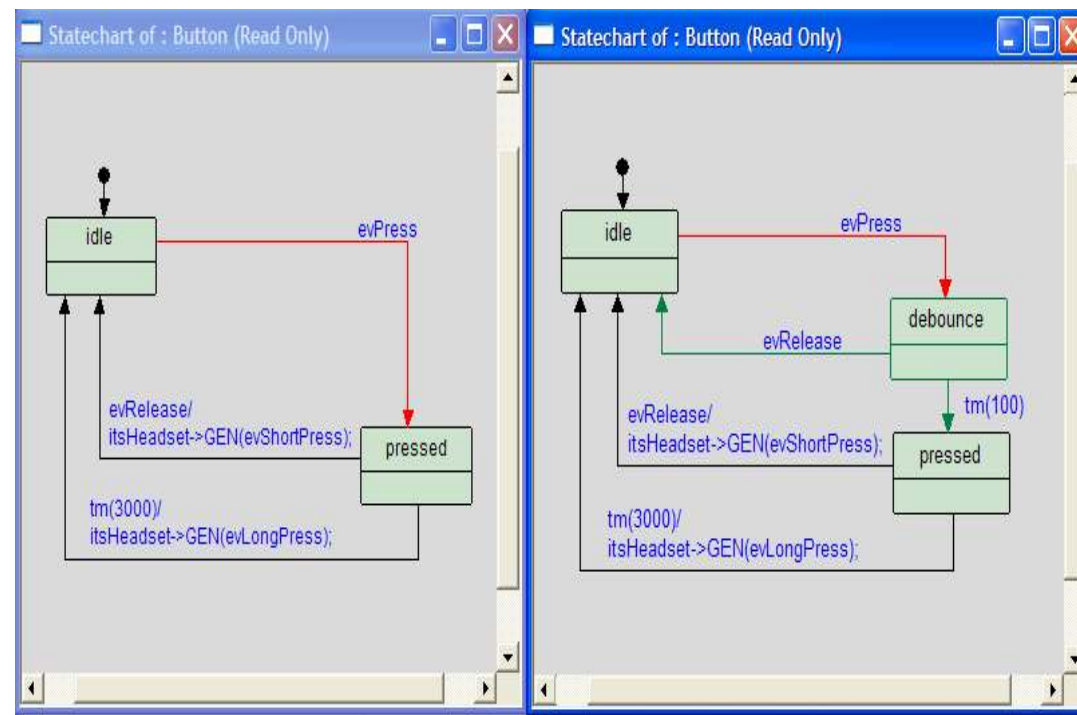
Colaboración, equipos distribuidos

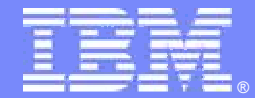
Rational. software

→ Go to **IBM**

Colaboración

- Soporte a equipos de desarrollos de cualquier tamaño
- Gestión del desarrollo en paralelo con comparador gráfico de diferencias entre modelos (también merge)
- Integración con herramientas de configuración de software como Rational ClearCase®, Rational Team Concert y Telelogic Synergy™





IBM Software Group

Resumen

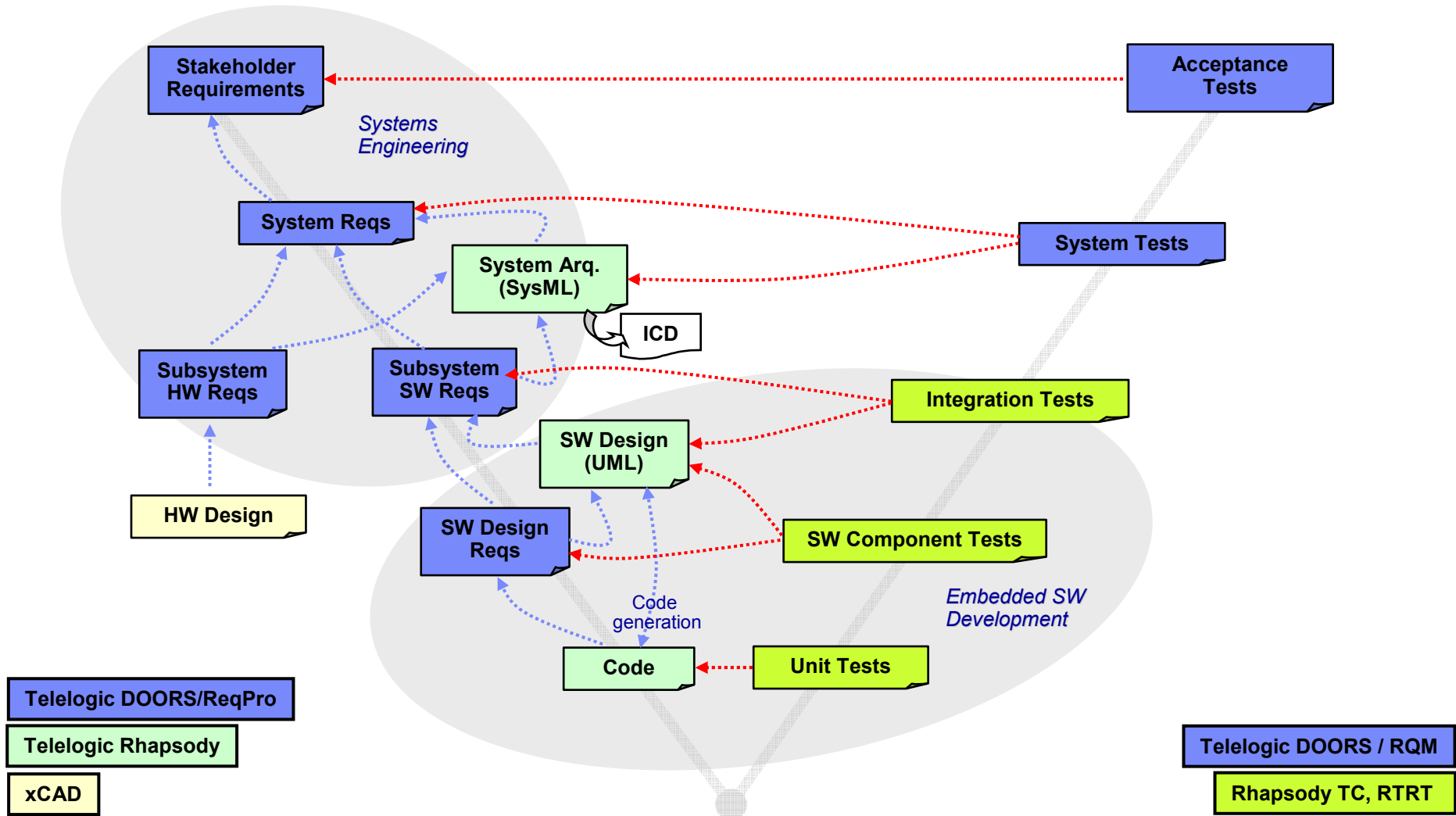
The Rational software logo, featuring the word 'Rational' in white on a blue rectangular background, followed by the word 'software' in a smaller, black, sans-serif font.A button with a right-pointing arrow and the text 'Go to IBM' in a bold, sans-serif font.

Resumen - IBM Telelogic Rhapsody ®

- IBM Telelogic Rhapsody® es el entorno de diseño y desarrollo de sistemas que **permite llevar a la práctica de forma productiva y eficiente los procesos de diseño y desarrollo de sistemas.**
- Desarrollo dirigido por modelos (MDD). **Mayor productividad desarrolladores**
- Pruebas dirigidas por modelos (MDT). **Mayor productividad ing. Pruebas**
- Trazabilidad a requisitos, **diseño cumple los requisitos.**
- Verificación temprana mediante simulación, detección de errores antes de las pruebas, **ahorro de costes.**
- Mejora la **comunicación** entre disciplinas. Uso de lenguajes estándar SysML y UML
- Facilita la **colaboración** a equipos distribuidos
- Trabaja como quieras: en el modelo o en el código (DMCA). **Disminuye el tiempo de aprendizaje**



Proceso, herramientas...



... y Telelogic Harmony

Harmony es la librería de mejores prácticas para el desarrollo de software y sistemas

- **Harmony/SE**
 - ▶ Systems Engineering
- **Harmony/ESW**
 - ▶ Embedded Software
- **Harmony/IT**
 - ▶ IT Software





Francisco J. López Minaya
Rational Technical Solution Architect
francisco.lopezminaya@es.ibm.com

© Copyright IBM Corporation 2008. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, Rational, the Rational logo, Telelogic, the Telelogic logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.

