



Telelogic

De los requisitos al código DOORS/Analyst y TAU G2

Antonio Rodríguez Perales

Telelogic

Antes de comenzar ...

- **¿Cómo están realizando la captura de sus requisitos?**
 - ¿Qué es captura de requisitos?
 - No lo hacemos
 - Documento Word
 - Herramienta de gestión de requisitos: Telelogic DOORS,...
- **¿Cómo están realizando el análisis de requisitos?**
 - ¿Qué es análisis de requisitos?
 - No tenemos requisitos
 - De forma textual en WORD o herramientas especializadas de gestión de requisitos
 - Utilizamos herramientas de modelado no UML
 - Utilizamos herramientas de modelado UML
- **¿Cómo editan aquellos documentos que combinan requisitos textuales y diagramas visuales?**
 - Copy & Paste
- **¿Cómo mantienen actualizada la documentación con los modelos visuales?**
- **¿Qué nivel de trazabilidad consiguen/buscan en sus desarrollos?**
- **¿Cómo especificáis los planes de prueba? ¿Están trazados a los requisitos?**

Desafíos del desarrollo de software

Aumento de la cantidad de Software

- Complejidad de los Sistemas
 - Movilidad e Interoperabilidad
- Software proporciona un forma de diferenciación y mejora de la competitividad



¿Por qué la productividad no aumenta tan rápido?

- Barreras de Comunicación – diferentes tecnologías, técnicas y notación en cada fase
- Entornos de trabajo distribuidos
- Desarrollo basado en componentes (reutilización): fácil de proponer, difícil de implementar
- El desarrollo de software es una forma de artesanía

3

© Telelogic AB

Telelogic

Desafíos del desarrollo de software: Calidad (→ Productividad)

- La falta de calidad erosiona la productividad y los beneficios
 - El coste para la industria de EEUU en corrección de errores de software se estima entre 50 KM€ y **175 KM€ cada año.**
Fuente: NIST, USA Today, Standish Group, Forrester
- Testing es importante pero no es la solución
 - El coste de encontrar defectos en la fase de integración es muy caro (aunque mejor, en cuanto a coste e imagen, que entregar una versión con ese defecto)
- “Mejor prevenir que curar”
 - Establecer un proceso que facilite un producto sin defectos



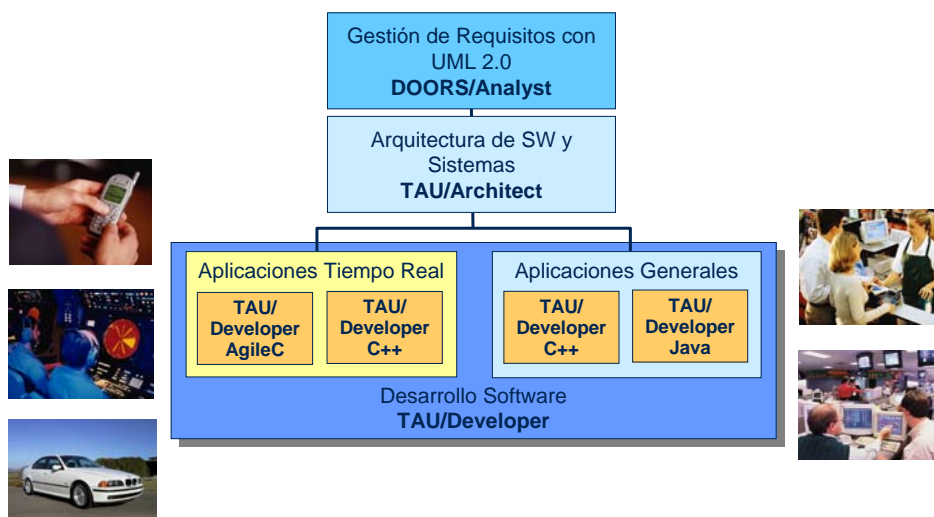
Source “Out of the Crisis” by W. Edwards Deming.

4

© Telelogic AB

Telelogic

Desarrollo a partir de los Requisitos



5

© Telelogic AB

Telelogic

Telelogic DOORS – El Líder contrastado

- **Líder mundial de mercado**
 - DOORS tiene el 41 % del mercado (Fuente: Informe Ovum 2004)
 - DOORS es el líder del mercado por **7º año consecutivo**
- **Líder a nivel tecnológico**
 - Valorada como la mejor herramienta de gestión de requisitos por Yphise, Octubre 2002.
 - “DOORS es el líder de mercado y el **producto mas maduro** con la mas amplia variedad de imprescindibles funcionalidades”
“Software Assessment Report – Requirements Management Tools” – Yphise
- “Telelogic DOORS es la herramienta líder en gestión de requisitos y está completamente integrada a lo largo del ciclo de vida”. *Thomas Murphy, Analyst, META Group*

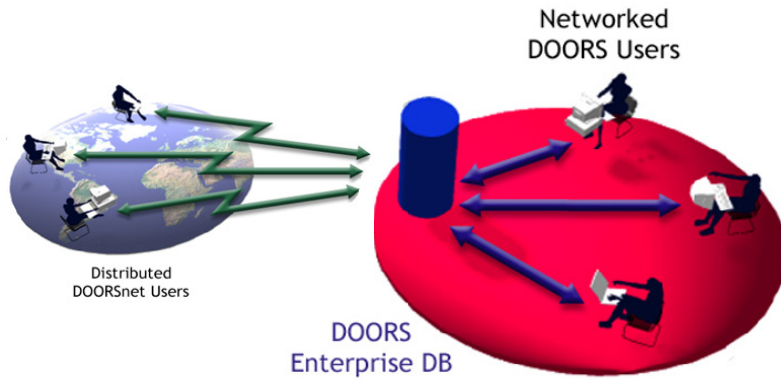
6

© Telelogic AB

Telelogic

Arquitectura Telelogic DOORS

DOORS/ERS



Vista de la base de datos DOORS

The screenshot shows the 'DOORS Database: /Company Programs/Vehicle projects/Light Trucks - DOORS' window. The left pane shows a tree view of the database structure. The right pane shows a table of data. Green arrows point from labels to specific elements in the interface.

Name	Type	Description
Sports utility vehicle 4x4	Project	New 4 wheel drive Sports Utility
Over sized truck	Project	
Sports utility vehicle 4x2	Project	New 2 wheel drive Sports Utility
Prototypes	Folder	
Flat bed trucks	Folder	
Common components	Folder	
Project definitions	Formal	
Truck Definitions	Formal	Common descriptions

Labels in the image:

- Carpetas**: Points to the 'Light Trucks' folder in the tree view.
- Proyectos**: Points to the 'Sports utility vehicle 4x4' project in the table.
- Documentos DOORS**: Points to the 'Project definitions' and 'Truck Definitions' rows in the table.
- Carpeta Borrada**: Points to the 'Prototypes' folder in the table.

Vistas “todo en una”

The screenshot displays the DOORS software interface with three main windows:

- Left Window:** A tree view showing a hierarchical structure of requirements, including sections like '3 Target Users' and '4 Requirements'.
- Center Window:** A text editor showing a requirement description for '3 Target Users' with a 3D pie chart visualization. Below the text is a table with columns 'High', 'Med', 'Low', and 'Nil'.
- Right Window:** A table listing requirements with columns for ID, description, and Test Method. The table includes sections like '1 Introduction', '1.1 Schedule', '2 User types', '2.1 Nationalities', '2.2 User sizes', and '3 Requirements'.

- **Objetos OLE**
- **Múltiples atributos**
- **Múltiples vistas**

9

© Telelogic AB

Telelogic

DOORS/Analyst: Expandiendo las posibilidades de DOORS

- **DOORS permite el tratamiento de requisitos textuales o imágenes embebidas como objetos OLE (solo lectura, editables dentro de DOORS a través de la tecnología OLE,...)**
- **DOORS/Analyst extiende DOORS con capacidades de análisis basada en UML 2.0:**
 - Representación visual de los requisitos, modelos de los sistemas y software
 - Trazabilidad desde los requisitos a diagramas o a los elementos de los diagramas
 - Creación automática de elementos visuales desde requisitos textuales y viceversa.
 - Mejora la comunicación entre analistas y arquitectos (diseñadores) software
 - diagramas DOORS/Analyst pueden ser impresos, visualizados en DOORS o DOORSnet y exportados a cualquier formato.

10

© Telelogic AB

Telelogic

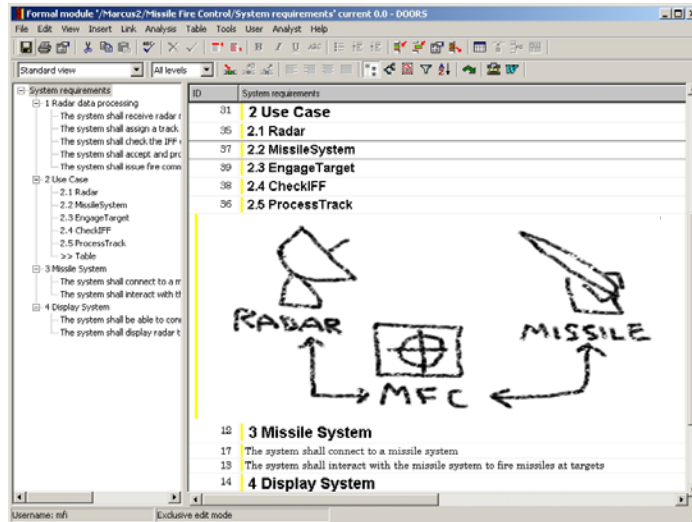
¿Por qué necesito modelos en UML 2.0?

- Una forma de introducir formalidad en la definición de los sistemas
 - Los modelos son diagramas, no solo imágenes
- Beneficios del análisis UML 2.0
 - Potencia la utilización de un único y preciso vocabulario para todo el sistema
 - Proporciona un medio para el estudio de un problema y sus potenciales soluciones
 - Permite visualizar en diagramas las especificaciones y el diseño
 - Permite refinar los modelos según se avanza en el diseño
 - Permite la validación de ciertos aspectos del diseño (arquitectura, comportamiento,...) mediante la simulación de los modelos
 - Potencia la comunicación entre equipos, tecnologías y empresas mediante la utilización de una notación estándar (UML 2.0)

Gestión de Requisitos y Análisis UML

- **Ambas son Técnicas Complementarias**
- Gestión de requisitos no es suficiente
 - Gestión de requisitos se centra en la captura y la trazabilidad entre requisitos atómicos textuales
 - Análisis UML permite la creación de múltiples vistas de esa información estructurada, facilitando la comprensión de los requisitos y mejorando la consistencia
- Análisis UML no es suficiente
 - **El modelo no son los requisitos**
 - Requisitos no-funcionales no son capturados en un modelo
 - Un modelo gráfico es insuficiente desde el punto de vista contractual
 - Se pierde la visión de contexto del sistema

DOORS/Analyst De la Teoría a la Práctica



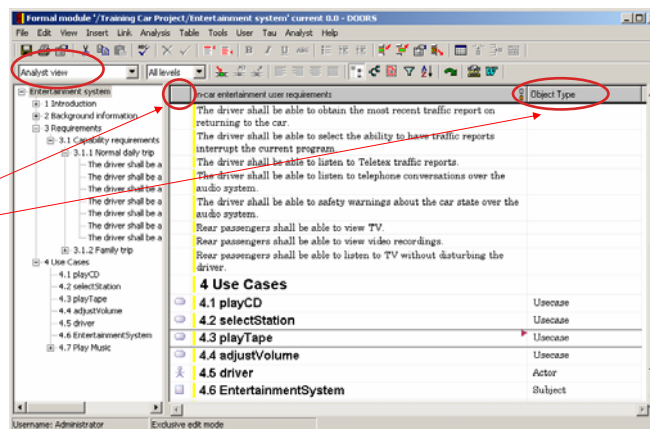
13

© Telelogic AB

Telelogic

DOORS/Analyst: Vistas Especiales

- DOORS/Analyst añade los elementos y diagramas UML como nuevos objetos del módulo. (Un objeto UML puede tener los mismos atributos que un objeto textual)
- Además añade el atributo "UML type" el cual presenta de forma textual y mediante iconos visuales



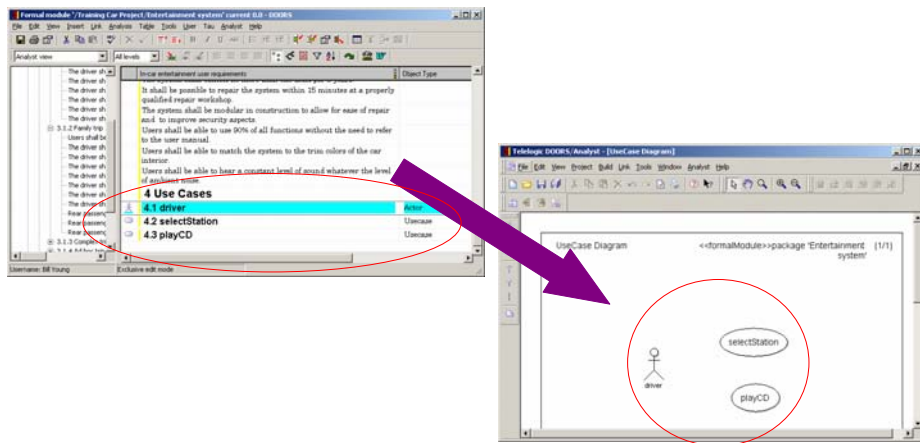
14

© Telelogic AB

Telelogic

DOORS → DOORS/Analyst

- Al ejecutar DOORS/Analyst desde DOORS, todos los objetos marcados como elementos UML son transferidos y generados automáticamente en el proyecto UML dentro de DOORS/Analyst



15

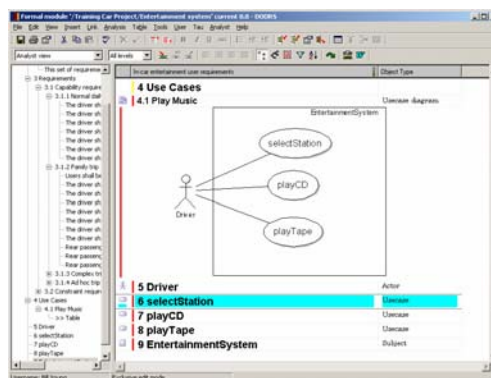
© Telelogic AB

Telelogic

DOORS ← DOORS/Analyst

- Cuando el diseño UML se guarda en DOORS/Analyst, automáticamente la información UML se sincroniza con la información textual y se añaden los diagramas:

- Elementos añadidos
- Elementos borrados
- Elementos renombrados
- Diagramas creados
- Diagramas borrados
- Diagramas modificados



16

© Telelogic AB

Telelogic

Beneficios de DOORS/Analyst

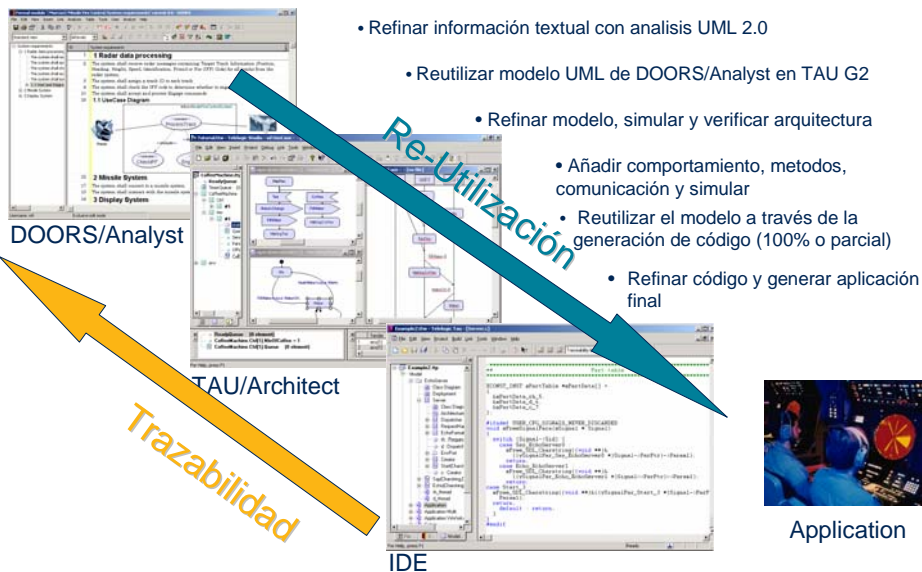
- **Facilita la comunicación en la gestión de requisitos**
 - Requisitos menos ambiguos
 - Mas facil demostrar los requisitos a los usuarios finales
- **Refuerza la colaboración con otros miembros del equipo**
 - Rapidamente enriquece los requisitos con conceptos gráficos
 - Asegura que los requisitos son completamente entendidos por todo el equipo, independientemente de su ambito
 - Los modelos visuales proveen un "lenguaje común" para todos
- **Asegura la validación de los requisitos**
 - Trazabilidad se extiende a analisis, diseño y finalmente a código
- **Disminuye la curva de aprendizaje y facilita empezar a usarlo**
 - Gestión de requisitos en ún unico entorno integrado
 - Los simbolos y diagramas usados son conocidos por la mayoría de los desarrolladores de software y sistemas
 - Basado en paradigmas de usabilidad familiares y diagramas estandarizados
- **Asegurar la trazabilidad de los requisitos**
 - Rapidamente encontrar los elementos relevantes del modelo via enlaces bidireccionales desde los requisitos textuales
- **Facilita el Mantenimiento de la Documentación**
 - No es necesario preocuparse de tener actualizadas las diferentes representaciones (textual y visual)

17

© Telelogic AB

Telelogic

Derriba las barreras de comunicación



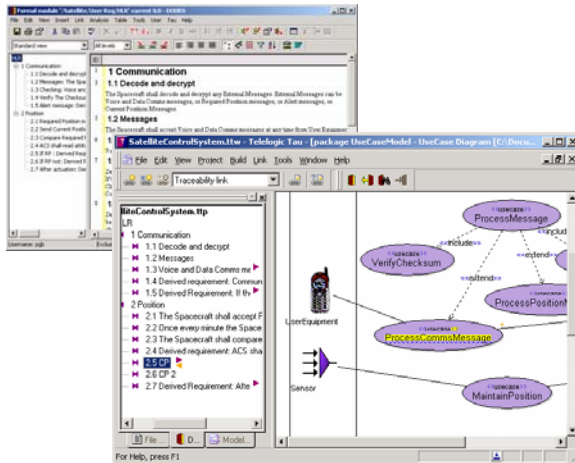
18

© Telelogic AB

Telelogic

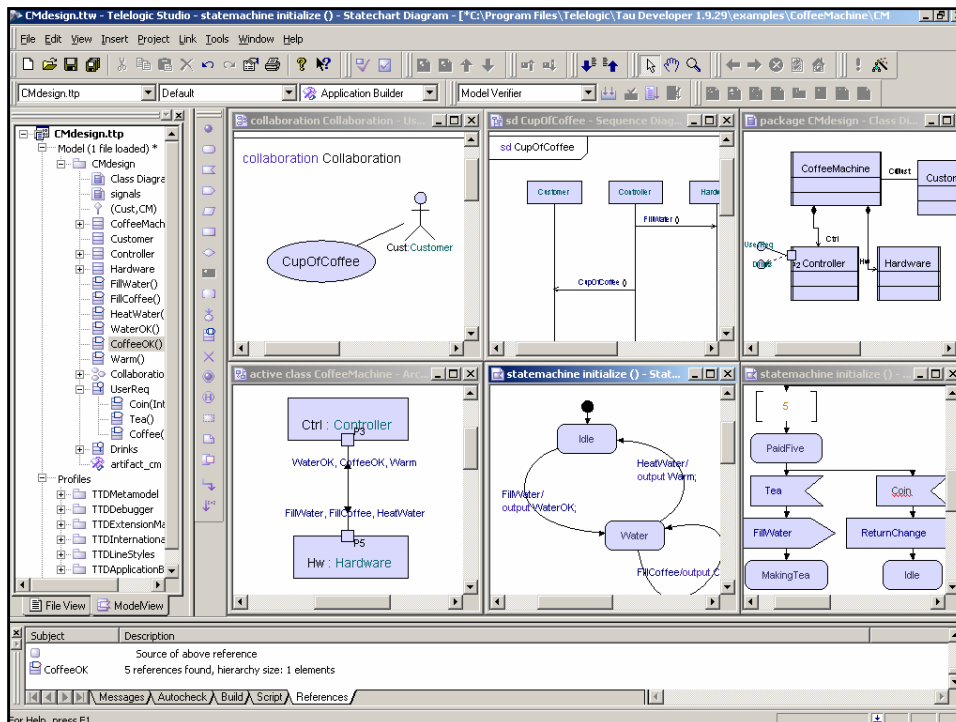
Extensión de la trazabilidad al desarrollo UML

DOORS



Tau/Architect & Tau/Developer

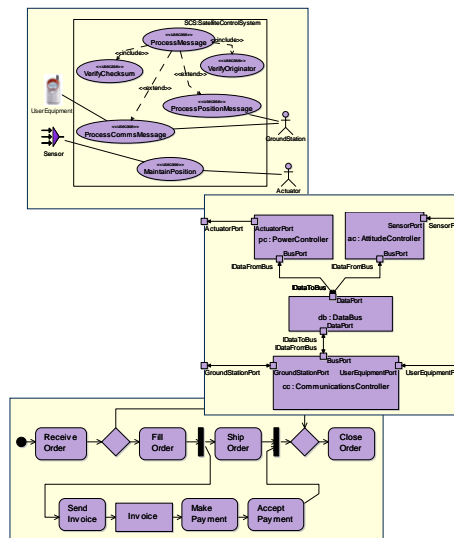
- Conexión de los requisitos a los elementos de análisis y diseño
- Asegura el cumplimiento de los requisitos en el diseño
 - Lectura y creación de enlaces sin utilizar DOORS
- Afina la granularidad en la trazabilidad a los modelos UML
- Proporciona navegación bi-direccional entre DOORS y TAU



Soporte UML 2.0

- Diagrama de Actividad
- Diagrama de Arquitectura
- Diagrama de Clases
- Diagrama de Componente
- Diagrama de Despliegue
- Diagrama de Interacción
- Diagrama de Paquetes
- Diagrama de Secuencia
- Diagrama de Estados
- Diagrama de Casos de Uso

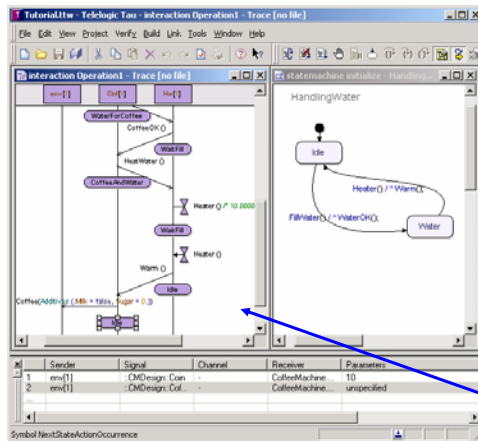
* Más información al final de la presentación



Modelado Activo de Sistemas

- Comenzar diseñando diagramas de arquitectura de alto nivel, diagramas de caso de uso y diagramas de secuencia
- El modelador (Active Modeler) pro-activamente generará las clases, operaciones, señales, puertos,...
- Reducción de tiempos y esfuerzo en la construcción del modelo completo

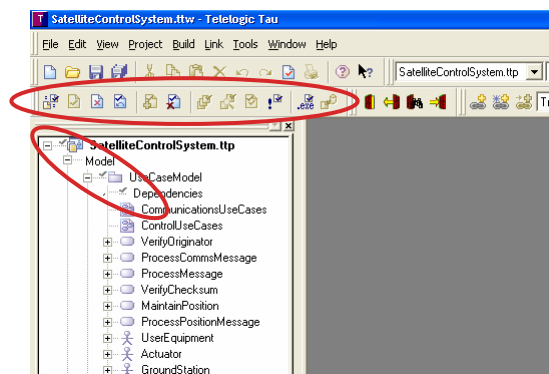
Verificación Dinámica de Modelos



- Modelos ejecutables
 - Aún no completamente acabados
- Verificación Temprana
 - Encontrar errores antes
- Facilita el chequeo de sistemas con comportamiento complejos

Generación automática de diagramas de secuencia

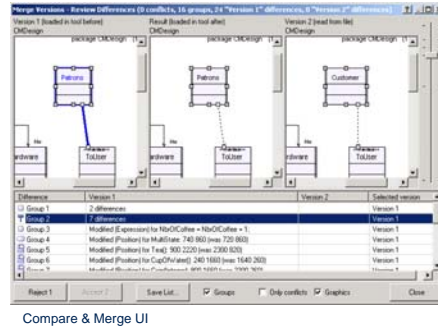
Tau/Developer – Trabajo en equipo



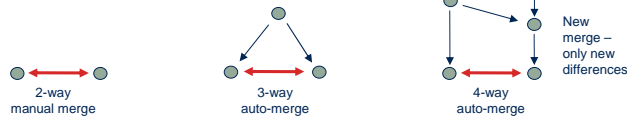
- Soporta SCCI
- Gestiona el trabajo en equipo
 - Gestiona configuraciones
 - Asegura la consistencia entre componentes
 - Soporta equipos distribuidos
- “Compare and Merge”

Compare/Merge

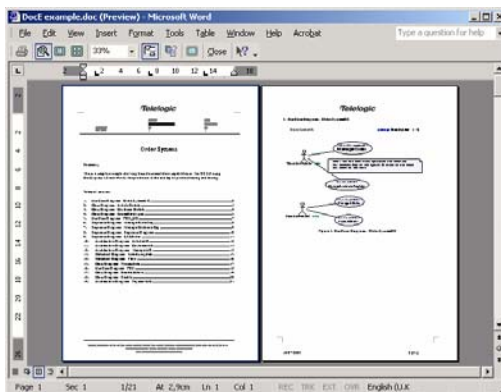
- Basado en modelos gráficos
 - Reconoce cambios semánticos y gráficos
 - Detecta movimientos y cambios en los nombres de los componentes
- Utilización gráfica o mediante línea de comandos
 - Resalta gráficamente los cambios
 - 2, 3 and 4-way compare & merge
 - 'Batch auto-merge' en la integración con sistemas CM



Compare & Merge UI



Generación de Documentación



- Multi-fuente desde varias herramientas
 - Ejem: correlaciona modelos UML en Tau/Architect con los requisitos textuales en DOORS
- Documentación Formal
 - Generación basada en plantillas basadas en los estándares de la empresa
- Actualización automática



Telelogic

UML 2.0

Telelogic

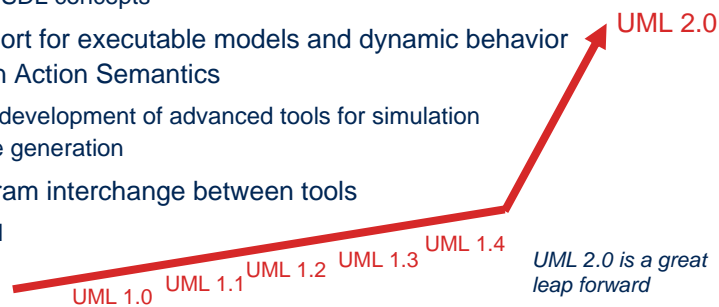
What is UML?

- Unified Modeling Language is a visual language for specifying, constructing, and documenting systems
- Derived from synthesis of Booch, OMT and Objectory modeling languages
 - roots can be traced to ER data modeling, Specification and Description Language (SDL), Message Sequence Charts (MSCs), Statecharts, etc.
- Adopted by OMG in November 1997 as UML 1.1
 - most recent minor revision is UML 1.4, adopted May 2001
 - next planned major revision is UML 2.0, submitted for approval Q1/2002



What Does UML 2.0 Improve?

- Internal UML cleanup
- Better support for Use scenario modeling
 - Sequence diagrams extended with Z.120 MSC
- Better support for component-based and architecture-centric development
 - Includes SDL concepts
- Better support for executable models and dynamic behavior including an Action Semantics
 - Enables development of advanced tools for simulation and code generation
- Better diagram interchange between tools
 - XML/XMI



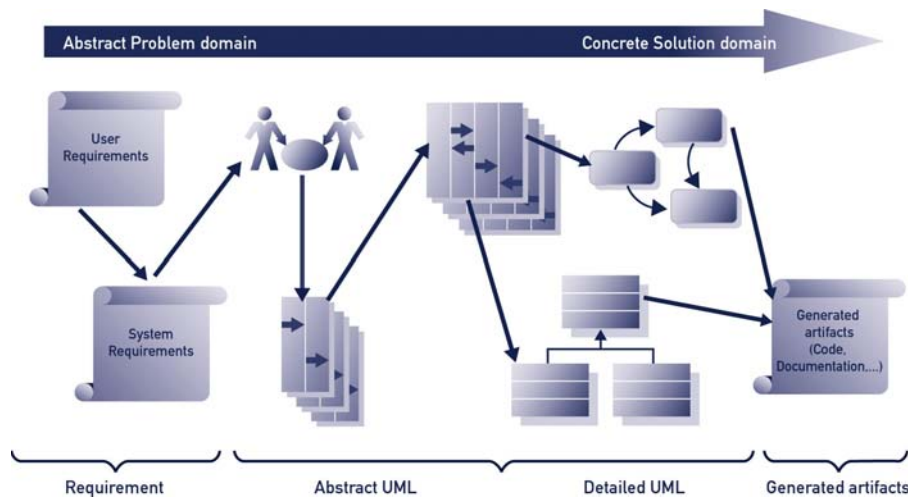
UML 2.0 Is Adopted

“Needham, MA, USA - June 12, 2003 - Members of the Object Management Group™ (OMG™) gathered last week in Paris at a meeting sponsored by Compuware Corporation, THALES and Sun Microsystems. In a key development, the Analysis and Design Task Force voted to recommend adoption of the Unified Modeling Language™ (UML™) 2.0 Superstructure specification, completing the definition of this major upgrade to the industry's main software modeling notation.”

<http://www.omg.org/news/releases/pr2003/06-12-03.htm>

Only minor changes can happen with the UML 2.0 from now on.

UML 2.0 Across the Lifecycle



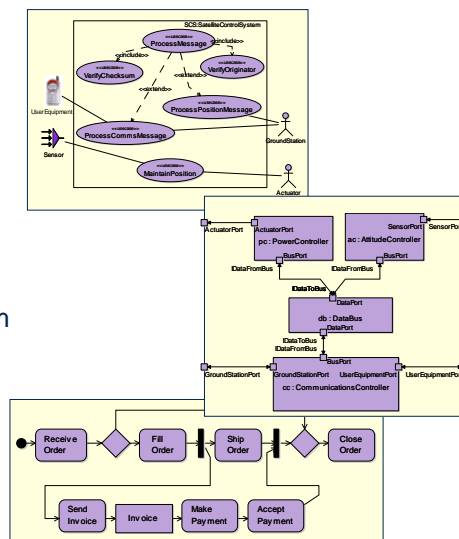
31

© Telelogic AB

Telelogic

Telelogic UML 2.0 Support – from Definition to Implementation

- Activity Diagram
- Architecture Diagram (a.k.a. Composite Structure Diagram)
- Class Diagrams
- Component Diagram
- Deployment Diagram
- Interaction Overview Diagram
- Package Diagram
- Sequence Diagram
- State Machine Diagram
- Use Case Diagram



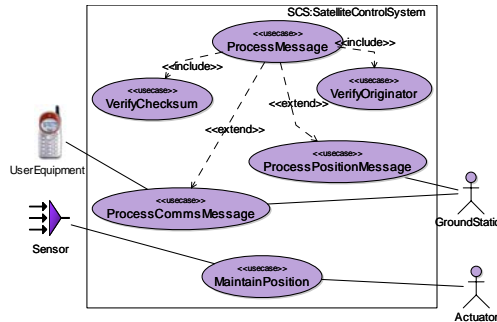
32

© Telelogic AB

Telelogic

Use Case Diagrams

- Describe the functionality of a system or part of a system
 - highlights the entities that interact with the system
- Visualization of systems requirements
 - promote understanding between different stakeholders
- Icons to illustrated real life components
 - instead of the standard symbols



33

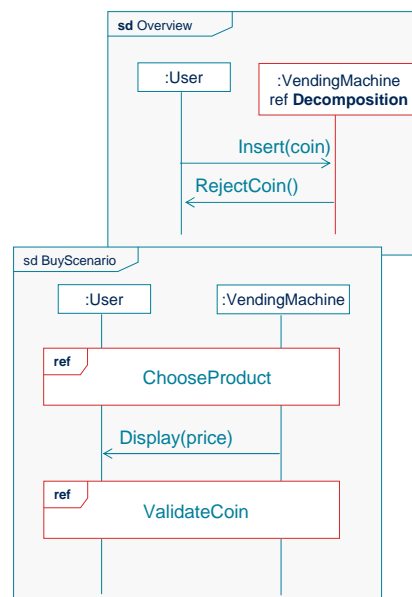
© Telelogic AB

Telelogic

Sequence Diagrams

(1/2)

- Extended and improved in UML 2.0
- Sequence Diagrams for extensive and scalable modeling of complex interactions
- Supports specification of real-life complexity
- Avoiding duplication of information ensures consistency and reduces amount of work

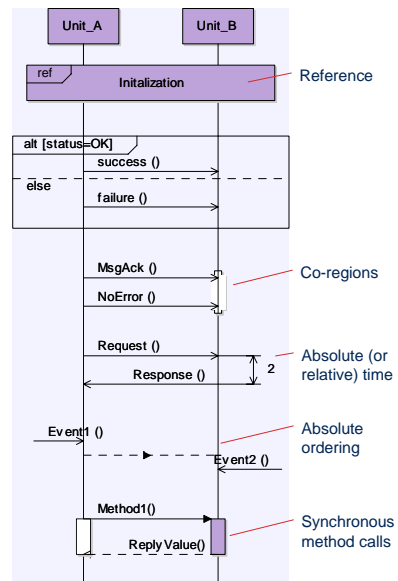
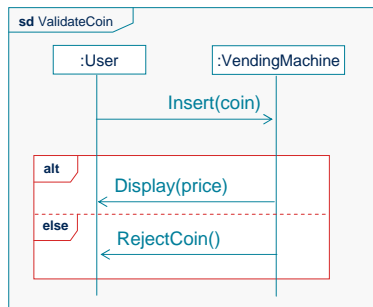


34

© Telelogic AB

Telelogic

Sequence Diagrams (2/2)



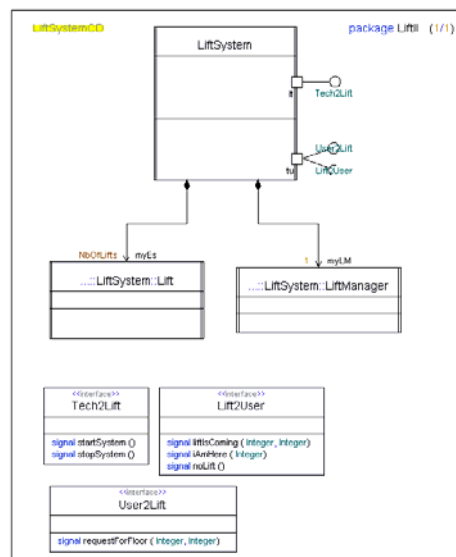
35

© Telelogic AB

Telelogic

Class Diagrams

- Packages
- Active and passive classes
 - Scoping and information hiding
 - Composition link
- Provided and Required Interfaces, ports and signals
 - Reactive systems



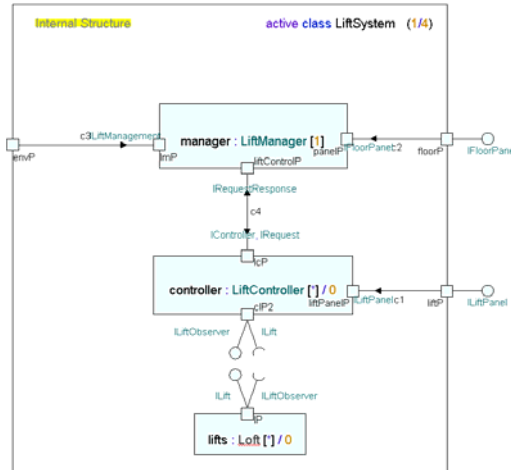
36

© Telelogic AB

Telelogic

Composite Structure Diagrams (also called Architecture Diagrams)

- A class can be used as a part of an internal structure of another class
 - Hierarchical decomposition
- Connectors are used as contextual associations
 - Connect ports & interfaces



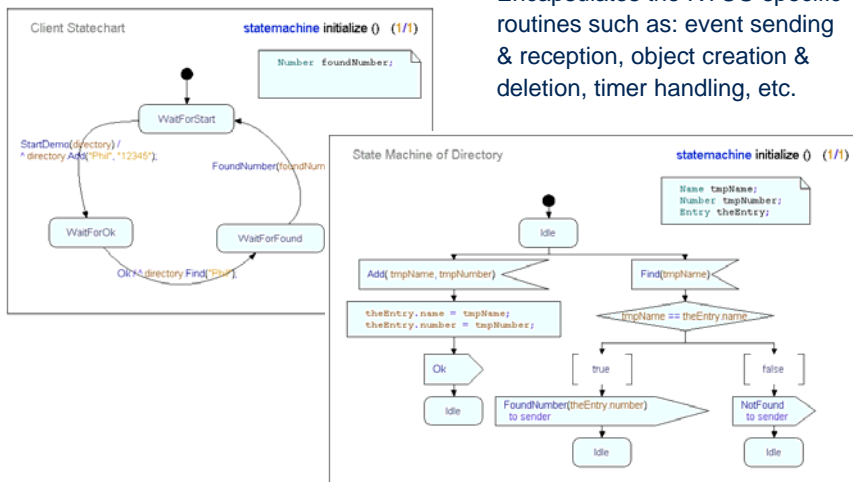
37

© Telelogic AB

Telelogic

Statecharts in Text and Graphic Forms

- Encapsulates the RTOS-specific routines such as: event sending & reception, object creation & deletion, timer handling, etc.



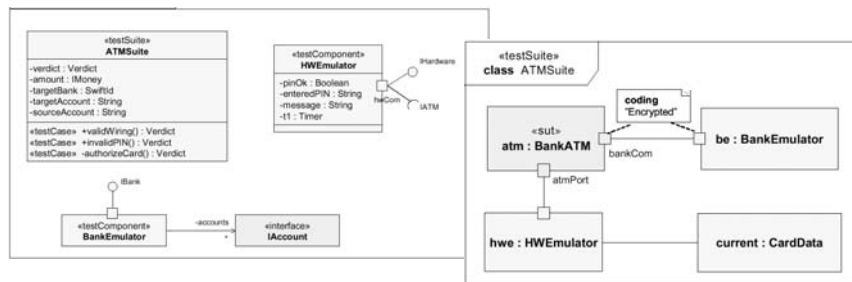
38

© Telelogic AB

Telelogic

Testing (1/2)

- Based on UML Testing Profile
- Focused on system features
 - Described by examples; System as a whole
- Construct the test-bench architecture and describe expected sequences
 - Class diagrams; Architecture diagrams; Sequence diagrams



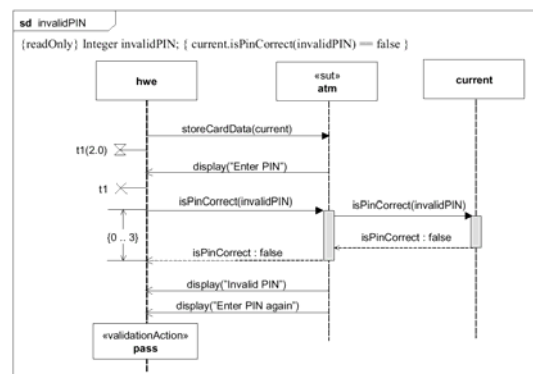
39

© Telelogic AB

Telelogic

Testing (2/2)

- Specify the expected services using Sequence diagrams



40

© Telelogic AB

Telelogic