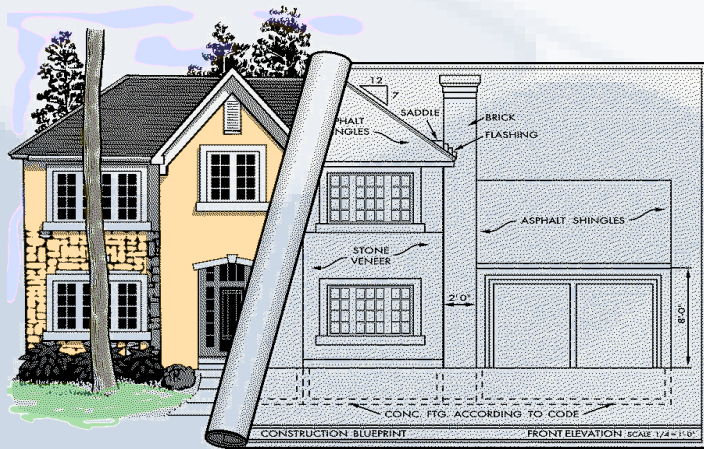


La Necesidad de Modelar

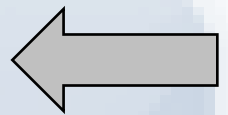
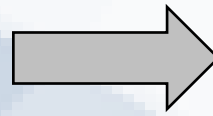


Analogía Arquitectónica

- ¿Tiene sentido poner ladrillos sin hacer antes los planos?
- El modelo, los planos, ayuda a afrontar la **complejidad del proyecto**.
- ¿Cuál es el **lenguaje adecuado** para representar los planos?
- Ingeniería directa e ingeniería inversa: una casa, un coche, un virus...



Ingeniería
directa



Ingeniería
inversa



Comunicación y Representación del Conocimiento



- Para representar el conocimiento hace falta un **lenguaje adecuado**.
- El conocimiento bien representado ayuda a hacerse las preguntas oportunas: ¿qué falta aquí? ¿qué pasaría si...? ¿por qué no se puede...?

¿Qué es un Modelo?

■ Abstracto

- Simplificación de la realidad: divide y vencerás.

■ Comprensible

- Expresado de tal forma que se pueda entender fácilmente.

■ Preciso

- Representa fielmente el sistema modelado.

■ Predictivo

- Se puede utilizar para obtener conclusiones correctas sobre el sistema.

■ Barato

- Más económico que construir y estudiar el propio sistema.

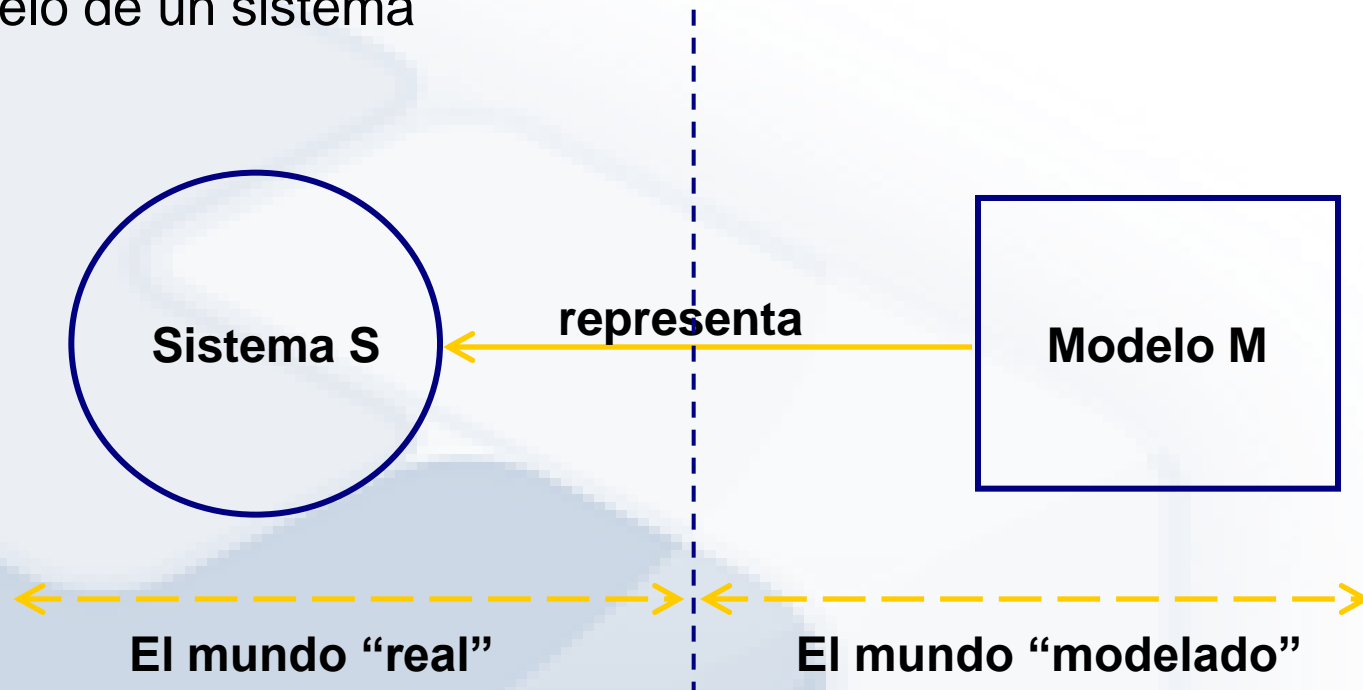
Bran Selic, "The Pragmatics of Model-Driven Development", IEEE Software, 20(5), 2003.

¿Qué es un Modelo? (II)

- Diversos tipos de modelos:
 - Estructura, electricidad, saneamiento...
 - Estático, dinámico...
 - ¿cómo se relacionan entre sí?
- Modelos formales y modelos informales.
 - **Modelos informales**: ad hoc, sin lenguaje común.
 - **Modelos formales**: lenguaje universal, precisión, rigor, coherencia.
- Modelado y lenguaje.
 - El lenguaje es vehículo del pensamiento: ayuda a **pensar con claridad**.
 - El modelado es un **elemento esencial** del proceso de desarrollo de software.
 - El modelado requiere un **lenguaje adecuado**.
- Metamodelo.
 - Los metamodelos **definen lenguajes** que permiten expresar modelos.
 - Un metamodelo describe un conjunto de elementos del modelo, su disposición, sus relaciones y sus restricciones.
 - Un modelo sólo es **válido** si es conforme a su metamodelo.

¿Qué es un Modelo? (III)

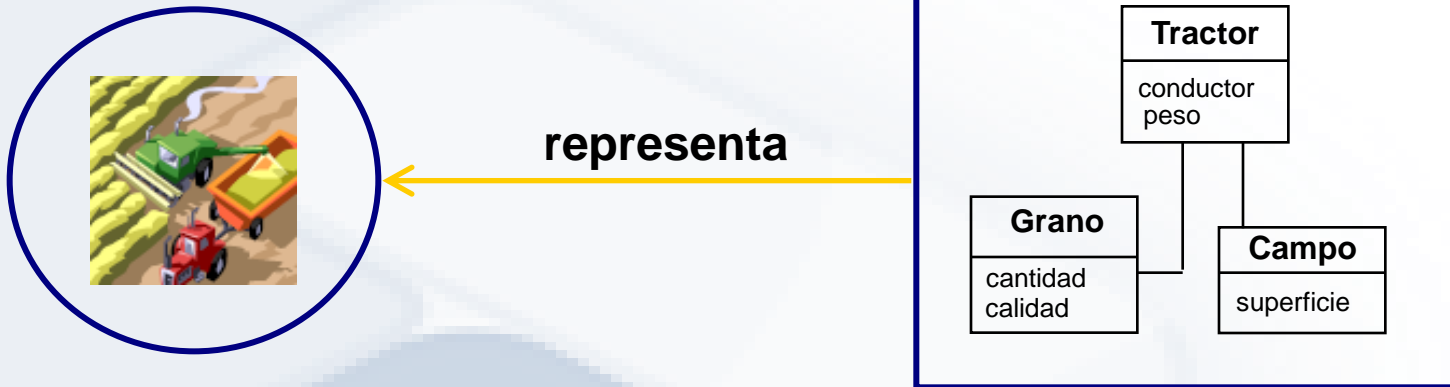
Modelo de un sistema



Jean Bézivin, "Model Engineering for Software Modernization",
The 11th IEEE Working Conference on Reverse Engineering, 2004.

¿Qué es un Modelo? (IV)

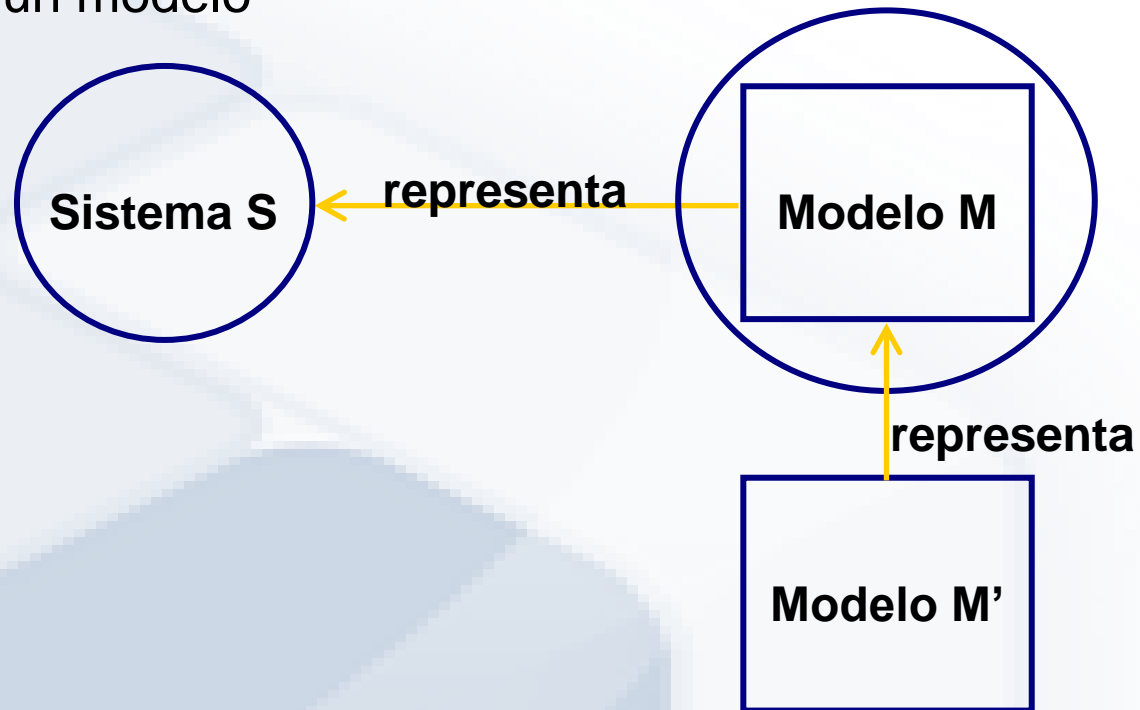
Modelo de un sistema



Jean Bézivin, “Model Engineering for Software Modernization”,
The 11th IEEE Working Conference on Reverse Engineering, 2004.

¿Qué es un Modelo? (V)

Modelo de un modelo



Jean Bézivin, "Model Engineering for Software Modernization",
The 11th IEEE Working Conference on Reverse Engineering, 2004.

¿Qué es un Modelo? (y VI)

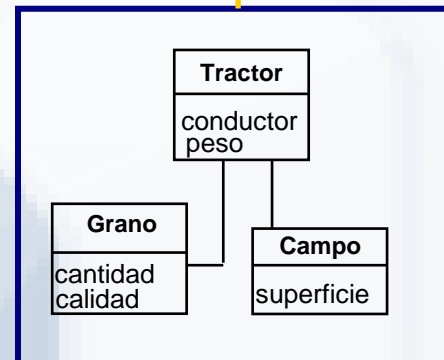
Modelo de un modelo



representa

```
351 ENTRANCE SECTION
352 I-BEGIN.
353 *
354 ACCEPT W-DATE1 FROM DATE.
355 ACCEPT W-DATE2 FROM DATE.
356 MOVE ZER0 TO Z2-PARAM.
357 MOVE ZER0 TO Z-EXT I-CALL.
358 MOVE ZER0 TO FEND-FLAG.
359
360 *
361 *
362 *
363 *
364 *
365 *
366 *
367 *
368 *
369 *
370 *
371
372 MOVE 'I-TABLE1' TO W-VAL.
373
374 IF (Z-EXT = SPACE OR = LOW-VALUE
375 GO TO W-EXIT-FORE.
376 MOVE Z-EXT TO W-FLAG.
377 MOVE ZER0 TO FEND-FLAG.
378 FEND ANY RUTA.
379 IF OR GET RUTA.
380 IF INEQ MOVE '01' TO Z-EXT
381 GO TO W-EXIT-FORE.
```

representa



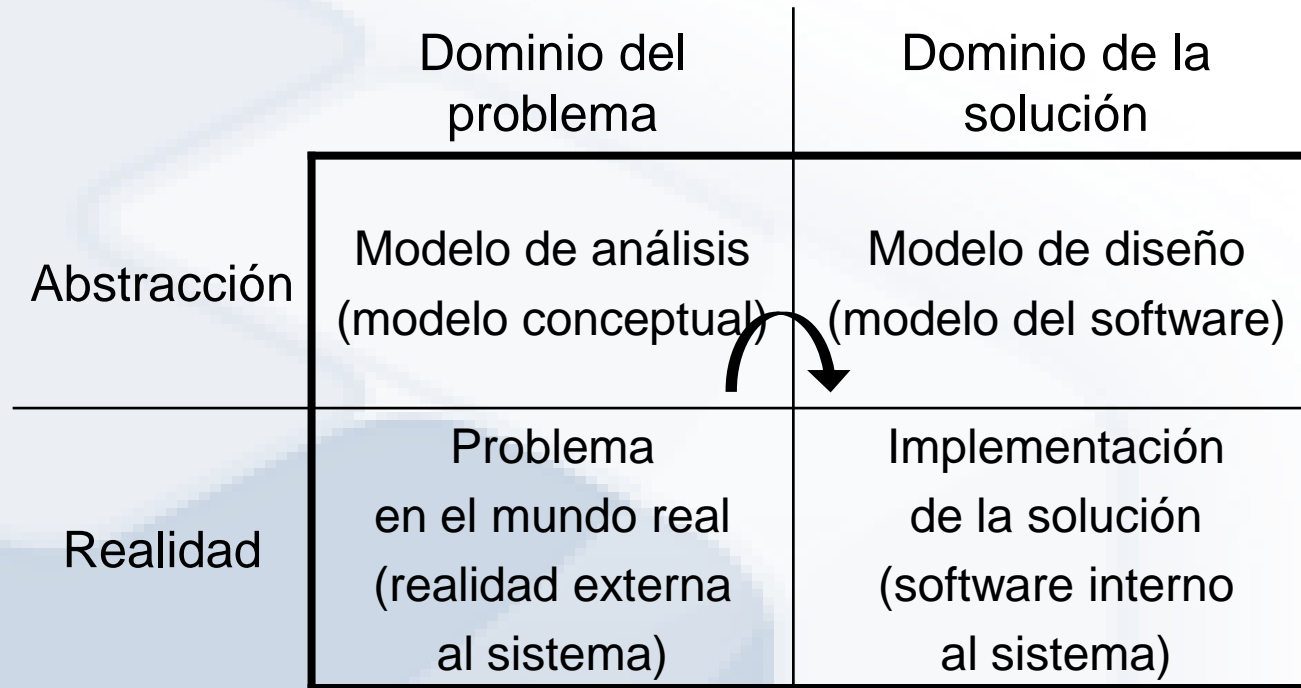
Jean Bézivin, “Model Engineering for Software Modernization”,
The 11th IEEE Working Conference on Reverse Engineering, 2004.

Modelos de Análisis y de Diseño

- Partes integrantes esenciales de la **documentación** de cualquier proyecto.
- A pesar de emplear una notación similar, los modelo de análisis y diseño son **abstracciones (modelos) de cosas distintas**.
 - **Análisis** (modelo conceptual): abstracción del problema, el **mundo real** tal como es antes, o como será después, de que se construya el sistema propuesto.
 - **Diseño** (modelo del software): abstracción de la construcción interna del **sistema** propuesto, que da solución al problema planteado en el mundo real.
- Por tanto, tienen diferente **propósito**.
 - **Problema**: estudiar los requisitos sin tomar decisiones de implementación.
 - **Solución**: establecer cómo debe construirse el sistema antes de construirlo.
- No hay **precedencia temporal** estricta análisis-diseño.
 - El análisis es anterior al diseño sólo dentro de cada iteración.
 - Son modelos interdependientes, pueden evolucionar en paralelo.
- La **transición** no es simple, inmediata ni automática (problema → solución).
 - Un buen diseño no se logra meramente “añadiendo detalles” al análisis.
 - El modelo de análisis no puede transformarse simplemente en una parte del modelo de diseño: representan realidades distintas.

Modelos de Análisis y de Diseño (II)

- Representan realidades distintas



H. Kaindl. "Difficulties in the Transition from OO Analysis to Design". IEEE Software, 16(5), 1999.

Modelos de Análisis y de Diseño (III)

- Análisis: creación de una especificación del problema y de los requisitos.
 - Exploración y esclarecimiento de los **requisitos** del sistema.
 - Construcción de un **modelo del mundo real** basado en los conceptos del dominio (modelo conceptual).
 - **Qué** debe hacer (o hace: ingeniería inversa) el sistema, pero no **cómo**.
 - No introduce artefactos de diseño o implementación.
- Es difícil determinar dónde acaba el análisis y dónde comienza el diseño.

Modelos de Análisis y de Diseño (y IV)

- Diseño: definición de una solución software que satisfaga los requisitos.
 - **Soluciones tecnológicas** para implementar los requisitos del sistema.
 - Construcción de un **modelo del sistema** antes de proceder a su construcción efectiva (modelo del software).
 - Incluye **aspectos de implementación**: patrones de diseño, librerías de clases, mecanismos de persistencia, componentes, etc.
 - Introduce **nuevos artefactos**: un objeto de análisis puede ser implementado por un conjunto de objetos de diseño.
 - Tiene en cuenta la **plataforma de implementación** (máquinas, sistemas operativos, lenguajes, etc.), así como consideraciones de eficiencia, rendimiento, optimización de recursos, etc.

El Lenguaje Unificado de Modelado

- Un lenguaje gráfico y formal para el modelado de sistemas
 - Visualizar, especificar, construir y documentar los artefactos de un sistema.
 - Lenguaje, no método, ligado a UML, pero sirve para otros métodos.
 - Orientado a objetos, toma sus conceptos de los lenguajes de programación orientados a objetos.
- Origen histórico
 - Antes de 1994: la “guerra de los métodos”.
 - 1994: **Booch** y **Rumbaugh** en Rational
 - 1995: **Jacobson** se une.
 - 1997: primera versión estándar certificada por el OMG.
 - Actualmente: la “guerra de las herramientas”.
- Documentación sobre UML
 - Libros de los Tres “Amigos”: Guía del Usuario, Manual de Referencia.
 - El Estándar (v2.2): www.omg.org
 - Revistas: Software and Systems Modeling, Journal of Object Technology.
 - Congresos: MODELS
 - Libros de otros autores, foros en Internet...

Elementos Principales de UML

- Cosas – representan distintos aspectos del modelo:
 - Estructura: son los “sustantivos” de UML, tales como clase, interfaz, atributo, componente, nodo...
 - Comportamiento: son los “verbos” de UML, tales como acción, actividad, interacción, estado, mensaje...
 - Agrupamiento: son los “paquetes”, que se usan para agrupar elementos relacionados semánticamente en unidades coherentes.
 - Anotación: son las “notas”, que pueden añadirse en cualquier parte del modelo para capturar información no gráfica.
- Relaciones – representan conexiones entre las cosas:
 - Asociación
 - Dependencia
 - Generalización
 - Realización
- Diagramas – representan un conjunto de cosas y relaciones de manera gráfica.

Tipos de Diagramas en UML 2

Modelado de requisitos	Diagrama de casos de uso	
Modelado estático	Diagrama de clases	
	Diagrama de objetos	
	Diagrama de paquetes	
	Diagrama de estructura compuesta	
Modelado dinámico	Interacción	Comunicación
		Secuencia
		Temporización
		Visión global de interacción
	Diagrama de estados	
	Diagrama de actividad	
Modelado físico	Diagrama de componentes	
	Diagrama de despliegue	

Sistema, Modelo y Diagrama

- Un **sistema informático** es una colección de elementos (hardware y software) organizados para cumplir una **finalidad** concreta.
 - Un sistema puede estar dividido en subsistemas.
- Un **modelo** es una **abstracción** de un sistema, es decir, una simplificación (completa y consistente) del sistema real, que sirve para comprenderlo mejor.
 - Provisionalmente, un modelo puede ser incompleto (faltan elementos) o inconsistente (contiene contradicciones).
 - Un sistema puede estar modelado desde distintos puntos de vista complementarios, según lo que se considere relevante en cada caso.
- Un **diagrama** es la representación gráfica de un conjunto de elementos interconectados, una **vista** parcial de un modelo.
 - Un modelo no es meramente una colección de diagramas.
 - Un modelo puede contener elementos no representados en un diagrama.
 - Un modelo puede contener especificaciones textuales esenciales.

Herramientas de Modelado

■ ¿Qué puede ofrecer una herramienta CASE para UML?

- Dibujo
- Corrección sintáctica
- Coherencia entre diagramas
- Integración con otras aplicaciones
- Trabajo multiusuario
- Reutilización
- Generación de código...

■ ¿Dónde encontrarlas?

- www.objectsbydesign.com

■ Ejemplos

- Altova UModel
- Visual UML
- TAU UML Suite
- MagicDraw UML

