

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA



BUAP

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

**APUNTES DE LA MATERIA DE INGENIERÍA DE SW II
OTOÑO 2022**

VI.4. INTRODUCCIÓN AL MODELADO Y UML

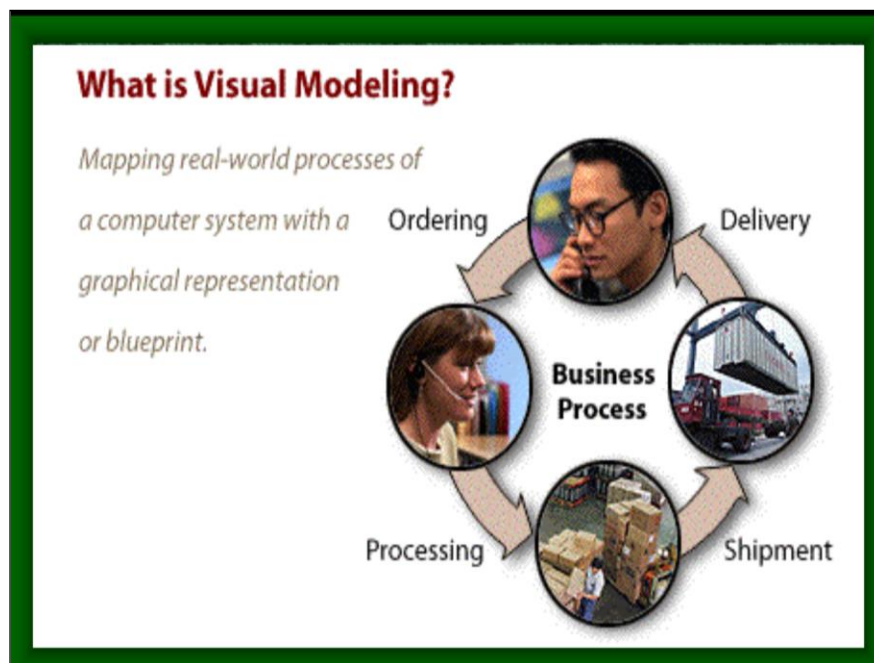
VI.4.1. EL MODELADO VISUAL Y UML

Hoy en día muchos de los procesos que intervienen en un negocio o empresa y que resuelven un determinado problema, requieren ser automatizados. Como ejemplo podemos citar a toda aquella empresa o negocio de ventas donde el proceso consistente en que un cliente de dicha empresa llame por teléfono para solicitar la compra de un producto y éste le sea llegado por medio de un determinado medio de transporte; pueda ser automatizado. Si nosotros logramos de alguna forma la visualización de los componentes de dicho proceso así como de las relaciones de esta simple transacción, podremos entonces comprender el funcionamiento del proceso de una manera mucho más simple y sencilla que si lo describiéramos en palabras.

El mapeo de los procesos del mundo real de un sistema software a una representación gráfica (blueprint) nos proporciona una visión más clara de ese sistema computarizado. A este mapeo se le conoce como **el modelado visual**.

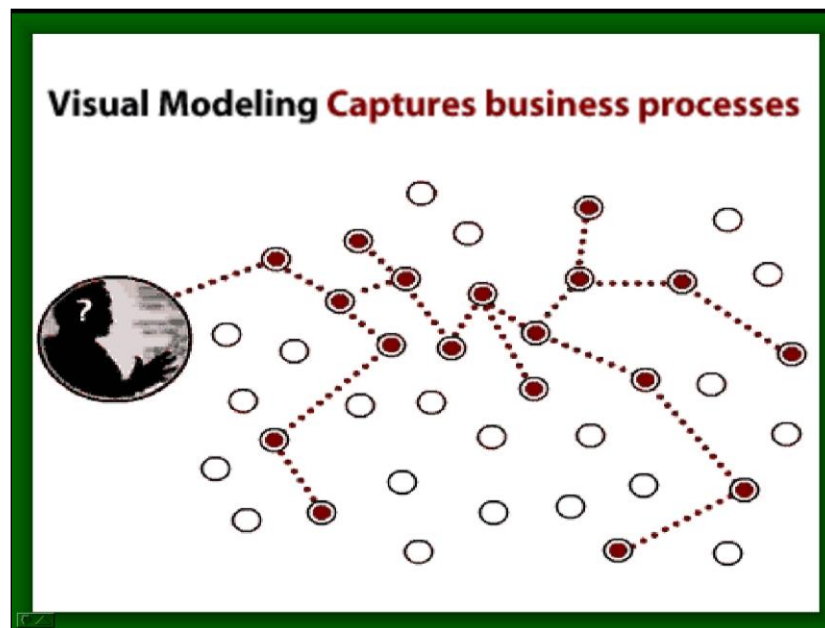
Son muchos los beneficios del modelado visual, entre ellos están:

- La captura e identificación de procesos
- Los enlaces de Comunicación
- El manejo de la Complejidad
- La definición de arquitecturas software
- La reusabilidad disponible



VI.4.2. CAPTURA DE PROCESOS

En la actualidad los procesos computacionales engloban una gran variedad de procesos que se dan en las empresas o negocios. Es por eso que para entender un sistema software es necesario definir aquellos requerimientos inmersos en la operación de dichos procesos. Sin la definición de esos requerimientos el sistema o producto software no podrá construirse. Cuando se crean los casos de uso, el modelado visual que se define en esa parte captura los procesos de la empresa o negocio identificando los requerimientos del sistema software desde una perspectiva del usuario. Esto precisamente da pauta a la construcción del diseño y desarrollo de los procesos implicados.




VI.4.3. COMUNICACIONES

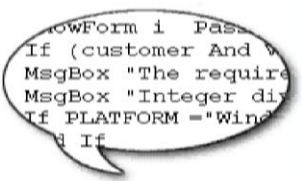
Los analistas y expertos en dominios son comúnmente quienes definen los requerimientos del sistema y las arquitecturas software. Por otro lado los desarrolladores son quienes construyen el sistema basándose en esos requerimientos. Estos dos grupos pueden y deben trabajar en equipo pero pueden existir diferencias, malos entendidos o desacuerdos entre ellos en la terminología que puedan emplear por separado.

El modelado visual tiene una comunicación estándar y el UML en particular proporciona una suave transición entre el dominio del problema y el dominio de la computadora. Mediante la comunicación a través de un lenguaje común de modelado visual, todos los grupos y miembros de un equipo de trabajo, en particular basados en el diseño, operan y se comunican con la misma terminología minimizando así los desacuerdos o diferencias e incrementando la eficiencia.

Visual Modeling **Enhances communications**

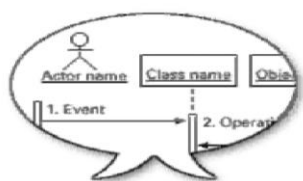


Business domain
Capture business objects and logic.



Computer domain
Analyze and design the application.


Visual Modeling **Enhances communications**



Business domain
Capture business objects and logic.

Computer domain
Analyze and design the application.

Visual Modeling **Enhances communications**

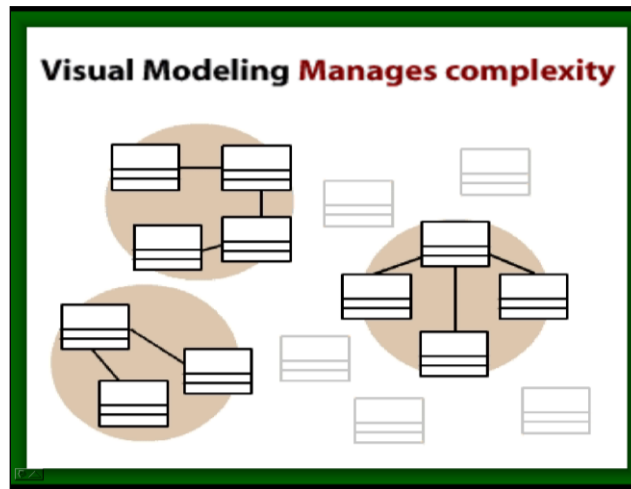


Business domain
Capture business objects and logic.

Computer domain
Analyze and design the application.

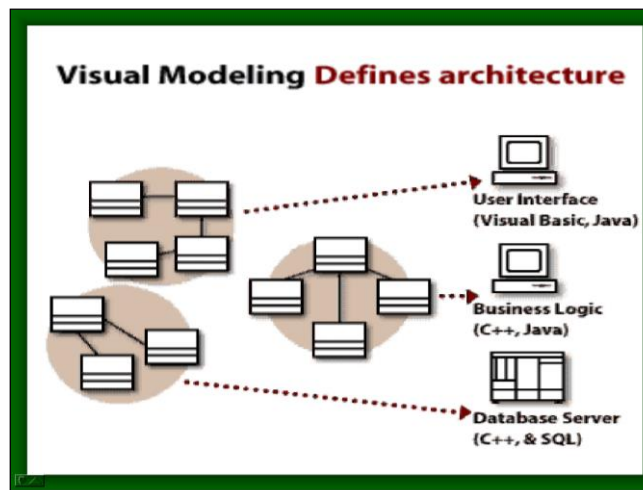
VI.4.4. MANEJO DE LA COMPLEJIDAD

Es común que en empresas donde se realiza ingeniería de software los sistemas estén conformados de cientos de clases. Dichas clases se organizan entonces para ser vistas por muchos grupos diferentes de personas en base a las necesidades de cada uno de ellos con una visión por tanto diferente. El modelado visual proporciona la capacidad de mostrar elementos de modelado de varias formas de tal manera que dichos elementos pueden ser vistos en diferentes niveles de abstracción. Un mismo modelo puede tener varias vistas diferentes dependiendo de las necesidades o intereses del visor.



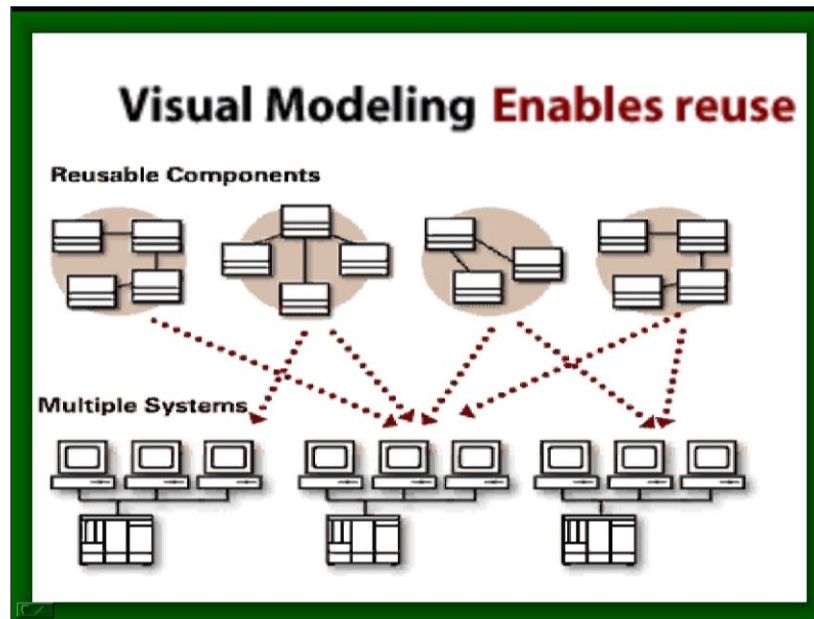
VI.4.5. DEFINICION DE ARQUITECTURAS SOFTWARE

El modelado visual proporciona la capacidad de capturar la arquitectura lógica del software independientemente del lenguaje de implementación que se quiera usar. En un sistema donde se diseñan procesos, el lenguaje de implementación es una herramienta determinada mientras que la arquitectura lógica de software es mapeada en una arquitectura física. Esta forma de mapeo proporciona una gran flexibilidad en el diseño de un producto software. Así por ejemplo si se decide cambiar la arquitectura del lenguaje C++ al lenguaje JAVA, con el modelado visual la arquitectura lógica es la misma y sólo necesita ser mapeada a la nueva implantación física.



VI.4.6. REUSABILIDAD DISPONIBLE

El modelado visual proporciona el uso de la reusabilidad de partes de un producto software o aplicación mediante la creación de componentes en el diseño del sistema. Estos componentes pueden ser compartidos y reutilizados por equipos de proyectos y los cambios pueden ser incorporados fácilmente dentro de aquellos proyectos existentes que están siendo desarrollados aun. La reusabilidad de componentes hace más flexibles los diseños.




VI.4.7. QUE ES UML?

Usando UML se puede llevar a cabo el modelado y obtener los beneficios enunciados en las secciones anteriores de éste capítulo. UML es el lenguaje estandar para la *visualización, especificación, construcción y documentación* de los artefactos o componentes de un producto o sistema software. Con UML:

- Se incrementa la productividad de una empresa.
- Se acorta el desarrollo de los ciclos de vida de un sistema software.
- Se adquiere una alta calidad del software que se construye.

How can you realize the benefits of visual modeling?

Use the Unified Modeling Language. *The UML is the standard language for visualizing, specifying, constructing, and documenting the artifacts of a software system. If you put the UML to work at your organization, increased productivity, shorter development lifecycles, and higher quality systems will become a reality.*



VI.4.8. EVOLUCION DEL UML

