

## Índice

- Introducción
- UML WAE (Web Application Extension)
- UWE (UML-Based Web Engineering)
- Ejemplo de MDWE (Model Driven Web Engineering)

## Introducción

- Podemos definir la *Ingeniería Web* como la aplicación de un método sistemático, disciplinado y cuantificable, al desarrollo, operación y mantenimiento de aplicaciones web, o la aplicación de la ingeniería al software web

## Introducción

- La ingeniería web está formada por muy diversas disciplinas:
  - Ingeniería del software
  - Sistemas de información
  - Interacción Persona-Computadora
  - Procesado de documentos y textos
  - e-learning
  - Agentes

## Introducción

- Hipermedia e hipertexto
- Comercio electrónico
- Seguridad
- Accesibilidad
- Fundamentos de ingeniería web
- Lenguajes de programación
- Otras

## Introducción

- De éstas, ingeniería del software y sistemas de información son dos de las más relevantes
- Nosotros nos vamos a centrar en ingeniería web, y en particular en modelado de aplicaciones
- Nótese que los temas de arquitectura multicapa y SOA también encajan en este apartado

## Introducción

- Hay muchas notaciones de modelado que pueden encontrarse
- vamos a centrarnos en dos muy vinculadas a UML:
  - UML WAE
  - UWE

## UML WAE

- UML Web Application Extension es una aproximación presentada por Jim Conallen
- Para Conallen, una aplicación web es un sistema web (servidor web, red, HTTP, navegador) en el que la interacción del usuario (navegación y entrada de datos) afecta el estado del negocio
- Conallen distingue entre aplicación web y sitio web

## UML WAE

- Para Conallen, una *página web* es básicamente cualquier cosa que puede ser servida por un servidor web
- Parte de la *separación de intereses*: el comportamiento de una *página web* en el servidor es distinto que en el cliente

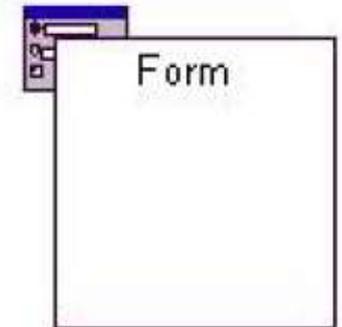
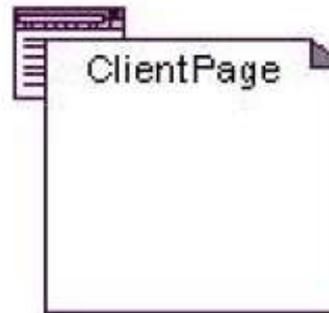
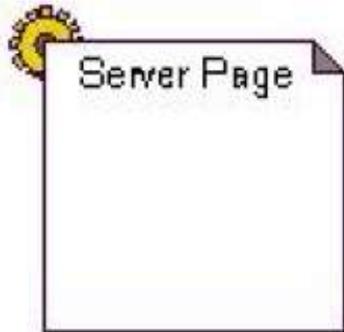
## UML WAE

- Conallén opta por asimilar páginas web a clases UML estereotipadas:
  - `client page`: página de cliente (p.e. HTML)
  - `server page`: página de servidor (p.e. Servlet)
  - `form`: formulario de entrada de datos

## UML WAE

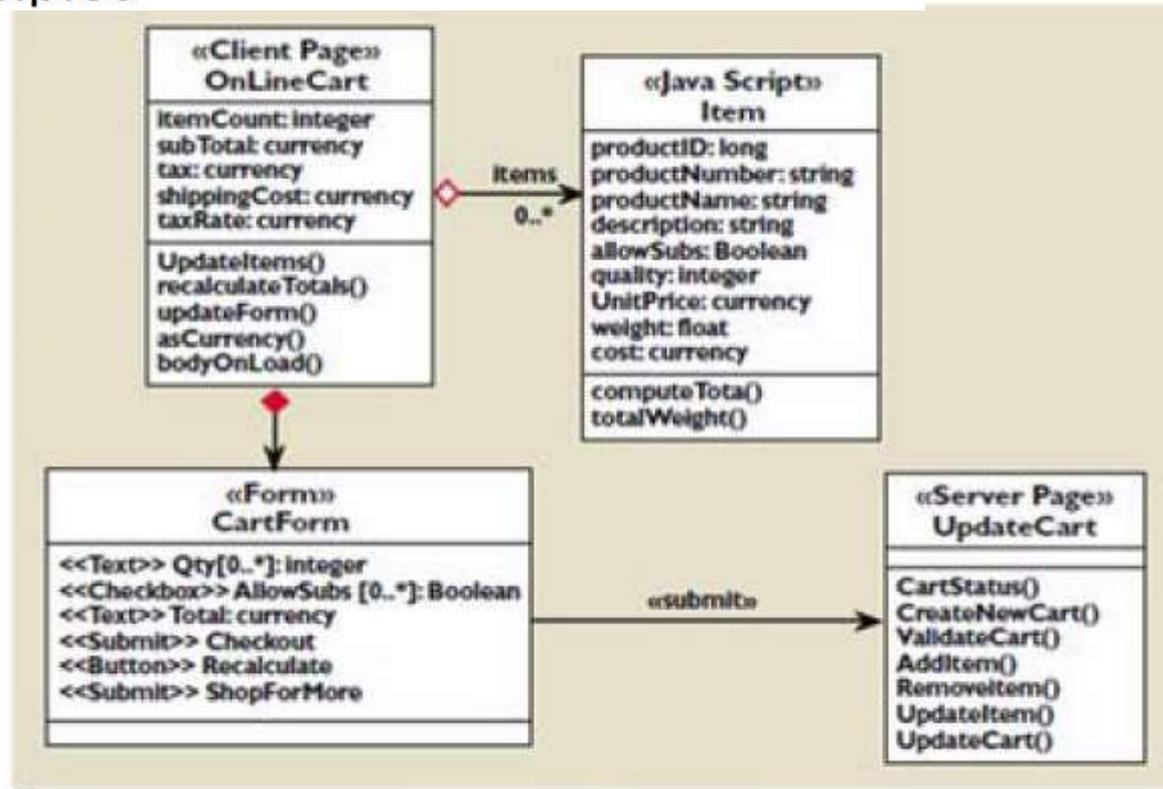
- Las relaciones entre páginas se establecen con asociaciones navegadas estereotipadas:
  - link: navegación entre páginas
  - build: construcción de una client page a partir de una server page
  - submit: envío de información de un form

## Clases estereotipadas de UML WAE



# UML WAE

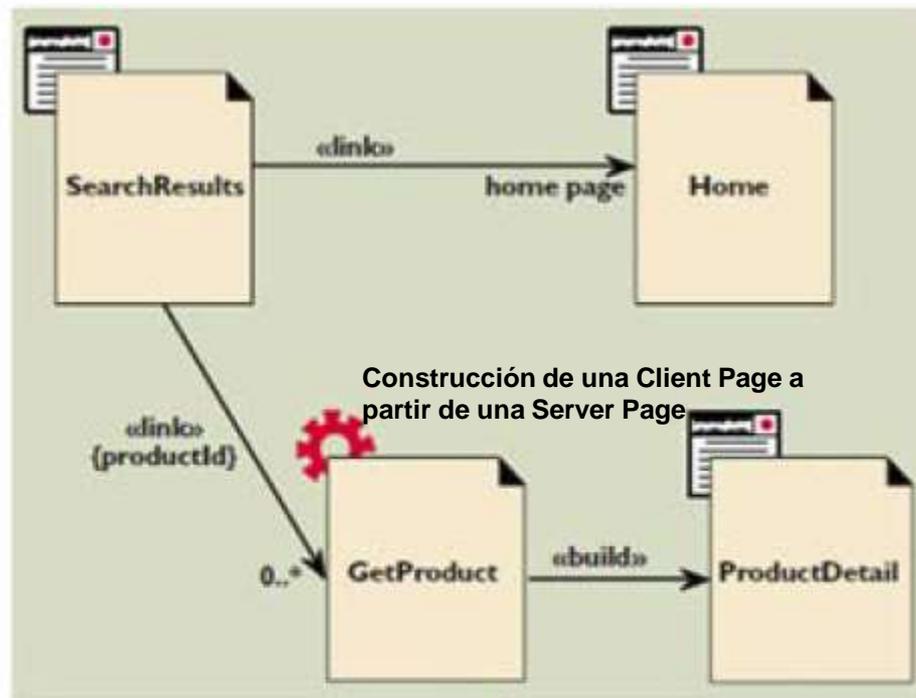
- Ejemplos



Notación UML WAE

## UML WAE

- Ejemplos



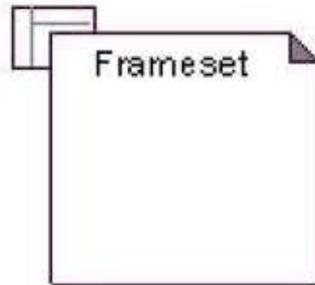
Notación UML WAE

## UML WAE

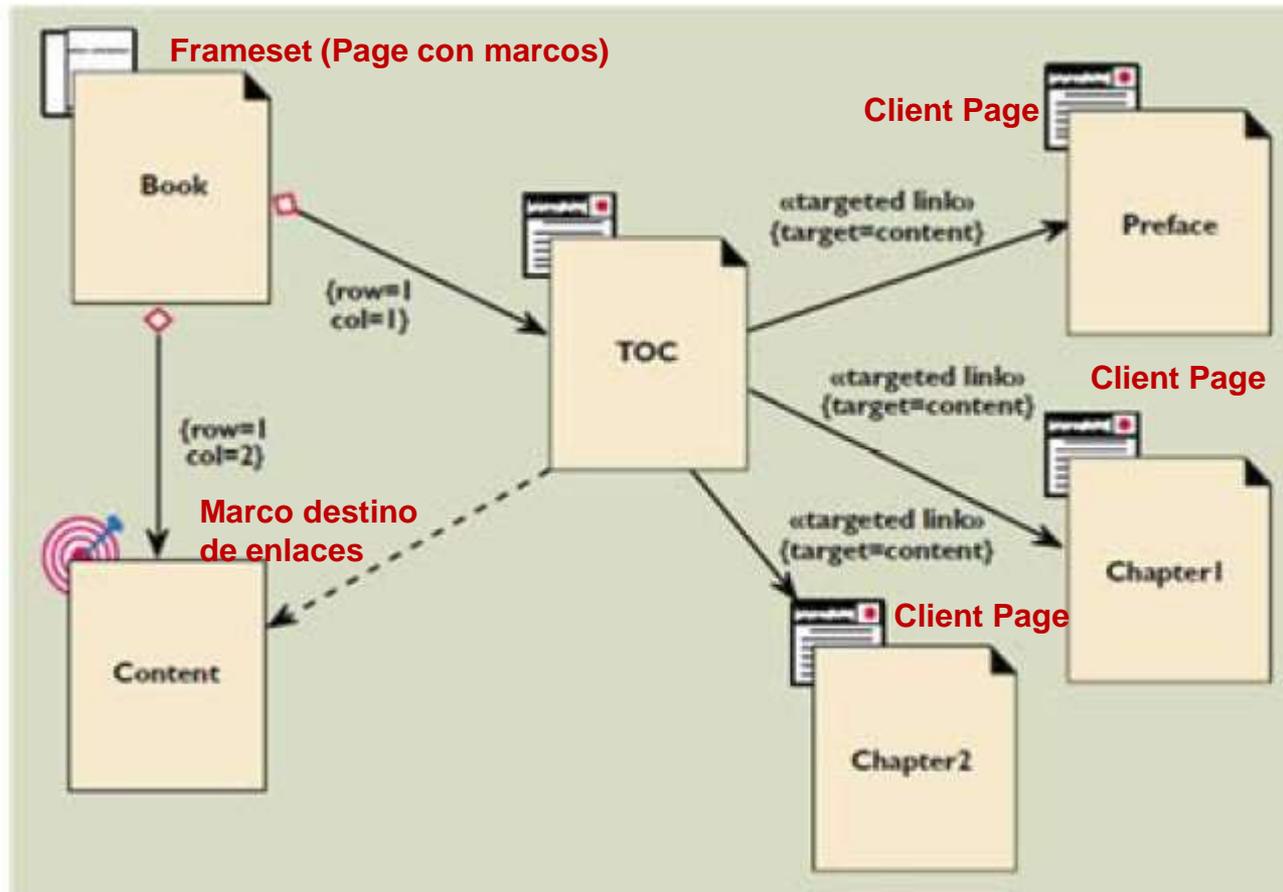
- También utiliza varios estereotipos para caracterizar marcos:
  - `frameset`: una página con marcos
  - `target`: un marco destino de enlaces
  - `targeted link`: el marco destino de un enlace

---

## Clases estereotipadas de UML WAE



# UML WAE



Marcos en UML WAE

## UML WAE

- Nótese que UML WAE es dependiente de la arquitectura
  - Model 1
  - MVC
- En el caso de una arquitectura MVC, UML WAE está restringido a la capa de presentación

## UML WAE

- Por tanto, para el modelado de una aplicación multicapa simplemente necesitamos
  - UML WAE para la capa de presentación
  - UML para el resto de capas
- UML WAE básicamente modela el código de los elementos software involucrados
- Normalmente las notaciones de diseño web se centran el modelado de las aplicaciones, en lugar del modelado del código

## UWE (UML-Based Web Engineering)

UWE es un método de ingeniería WEB orientada a objetos basada en UML, que puede ser utilizado para la especificación de aplicaciones WEB.

La aproximación propuesta por UWE provee:

- Una notación específica de dominio
- Un proceso de desarrollo basado en el modelo, y
- Una herramienta de soporte para la ingeniería de aplicaciones WEB.

## UWE Componentes

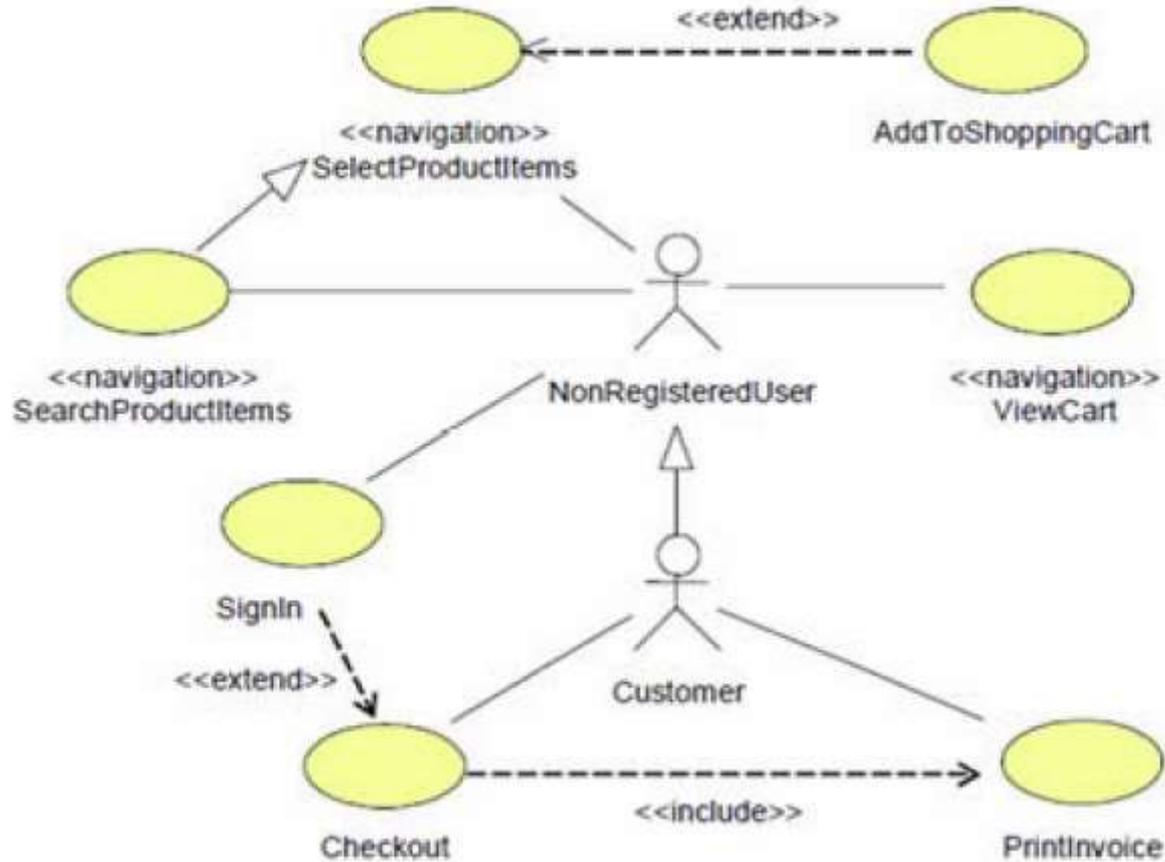
1. Modelo de casos de uso para capturar los requisitos del sistema.
2. Modelo conceptual para el contenido (modelo del dominio).
3. Modelo de procesos
4. Modelo de espacio de navegación.
5. Modelo de estructura de navegación.
6. Modelo estructural del proceso.
7. Modelo de flujo de proceso
8. Modelo de estructura de presentación

## UWE

### 1 • Modelo de casos de uso

- Es un modelo UML de casos de uso
- Los casos de uso que simplemente acceden a información se estereotipan como navigation

# UWE



Modelo UWE de casos de uso

## UWE

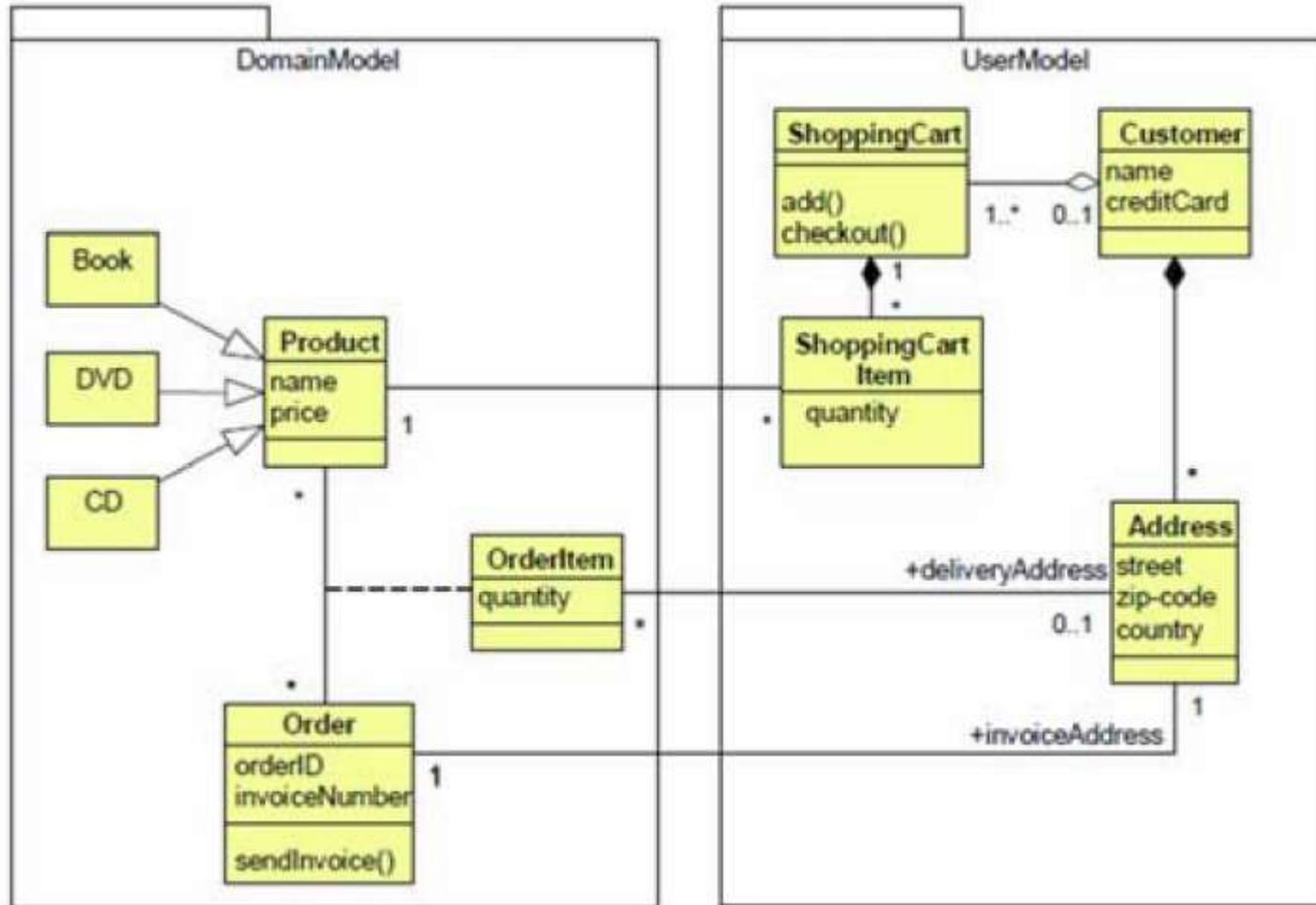
### 2• Modelo conceptual

– Representa los elementos persistentes de la aplicación

**-Su objetivo es reflejar un modelo conceptual del dominio de la aplicación considerando los requisitos consignados en los casos de uso.**

-Da como resultado un diagrama de clases de dominio.

UWE



Modelo conceptual UWE

## UWE

### 3 • Modelo de procesos

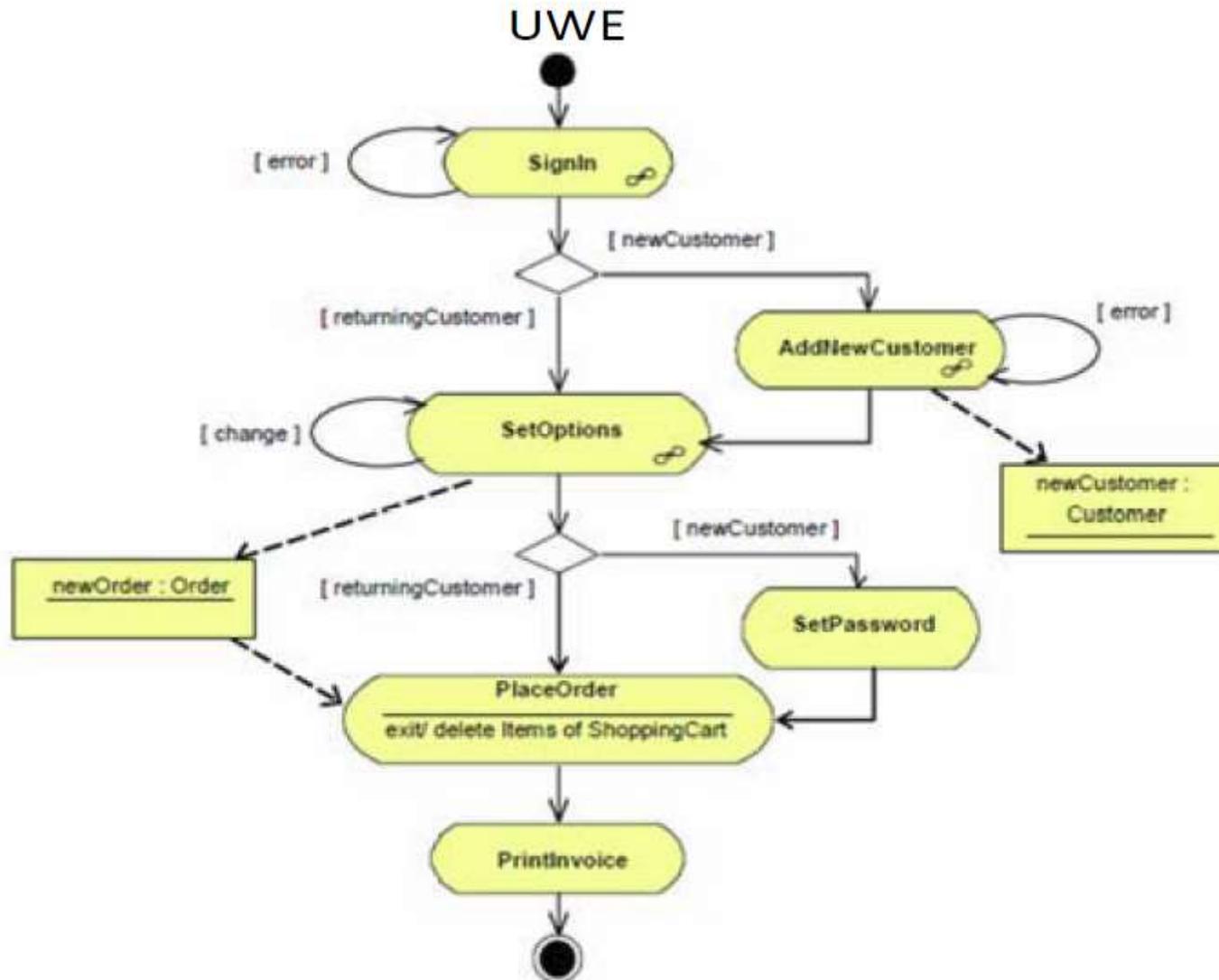
– Caracteriza las actividades llevadas a cabo en los casos de uso que no son de navegación

- Comprende:

1. **El Modelo Estructural de Proceso**, que describe las relaciones entre las diferentes clases de proceso y

2. **El Modelo de Flujo del Proceso**, que especifica las actividades conectadas con cada clase de proceso.

Ingeniería Web



Modelo de proceso para el checkout

## UWE

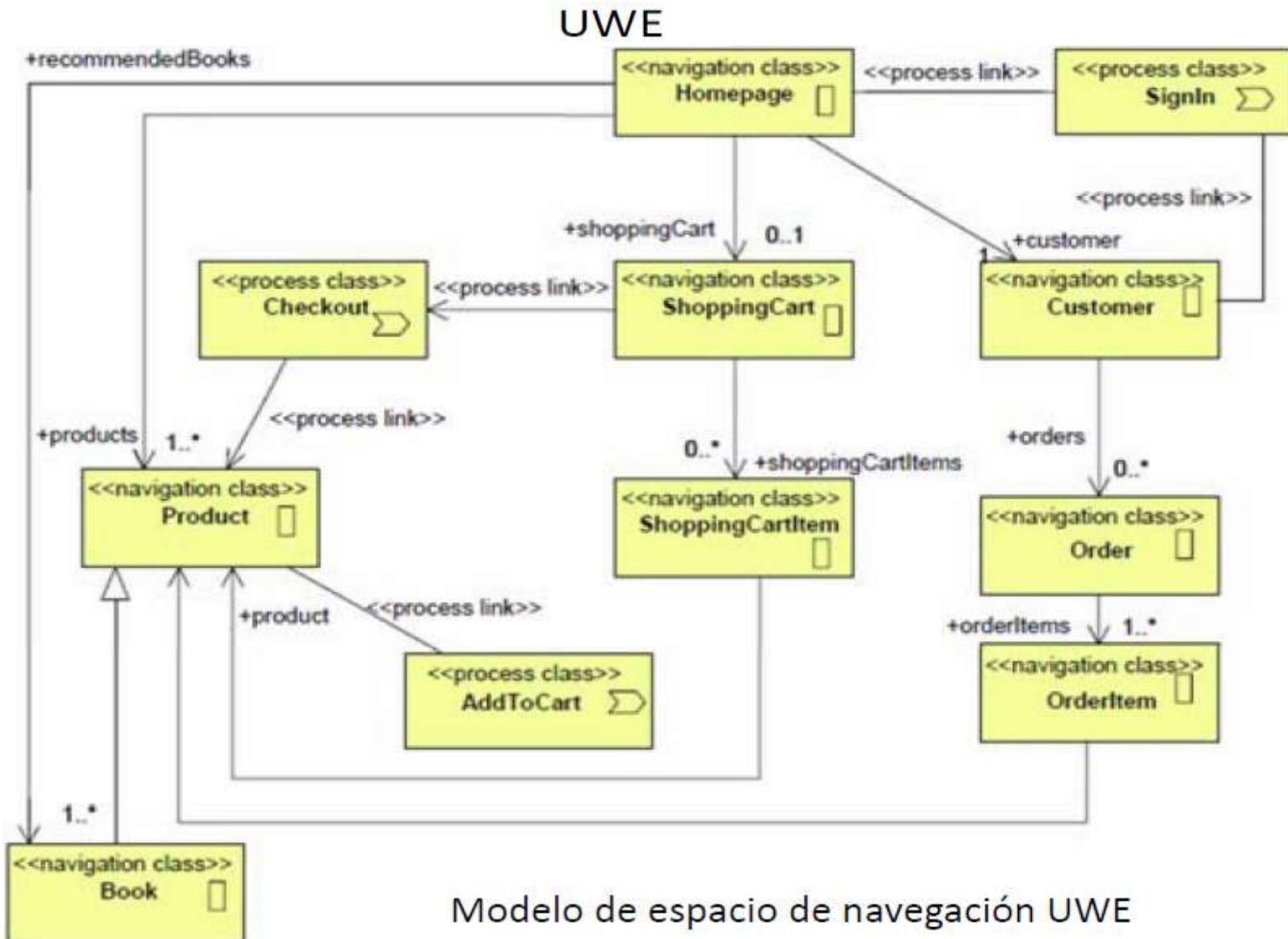
### 4 • Modelo del espacio de navegación

– Especifica qué objetos pueden ser visitados en la aplicación web

– Esterotipa dos tipos de clases:

- `navigation class`: vistas navegacionales de las clases del dominio
- `process class`: clase que implementa un caso de uso de proceso

Ingeniería Web



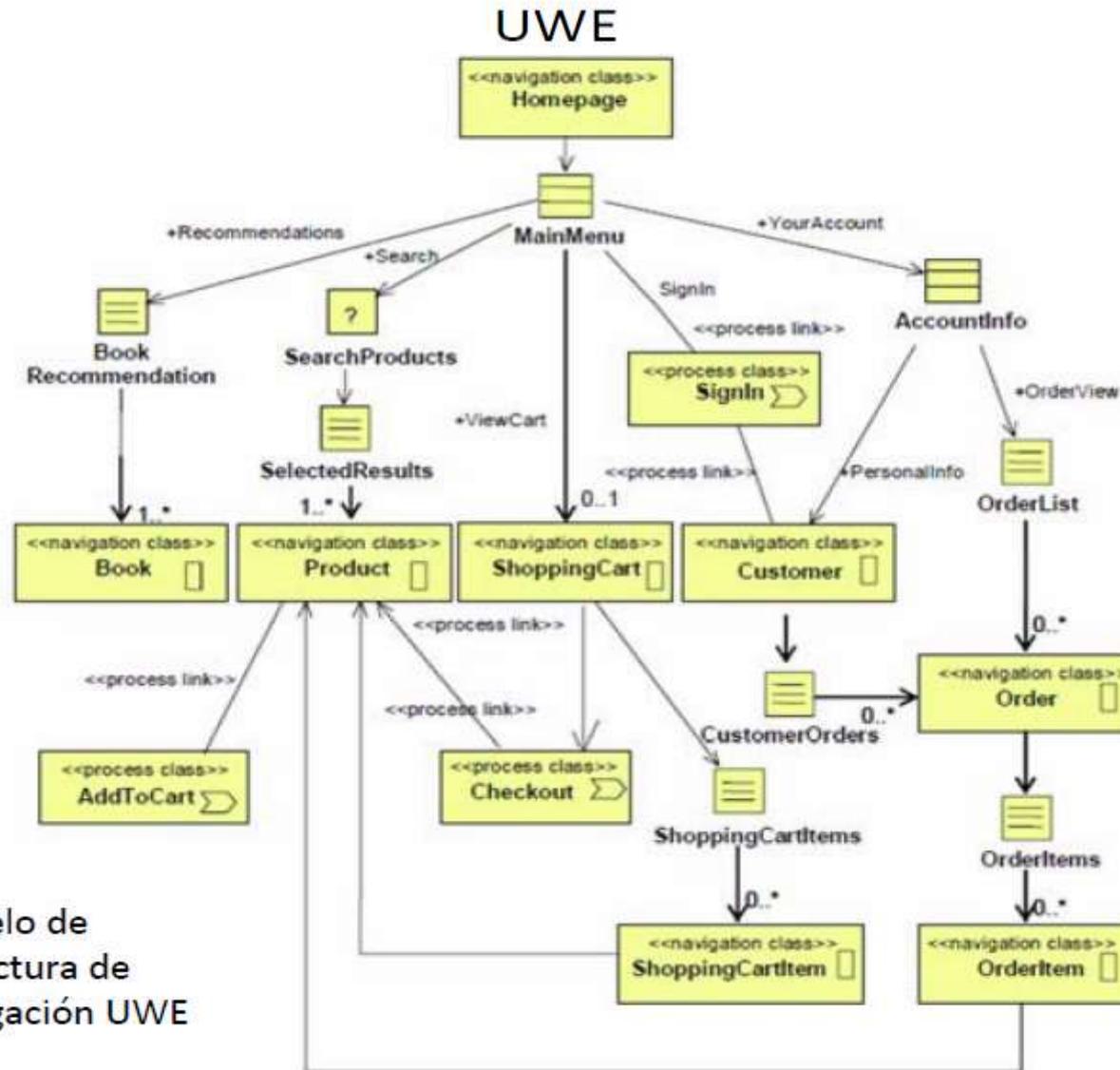
Modelo de espacio de navegación UWE

## UWE

### 5 • Modelo de la estructura de navegación

- Define cómo se pueden alcanzar los elementos del modelo del espacio de navegación
- Utiliza varias primitivas de acceso
  - Índices
  - Tours guiados
  - Querys

Ingeniería Web



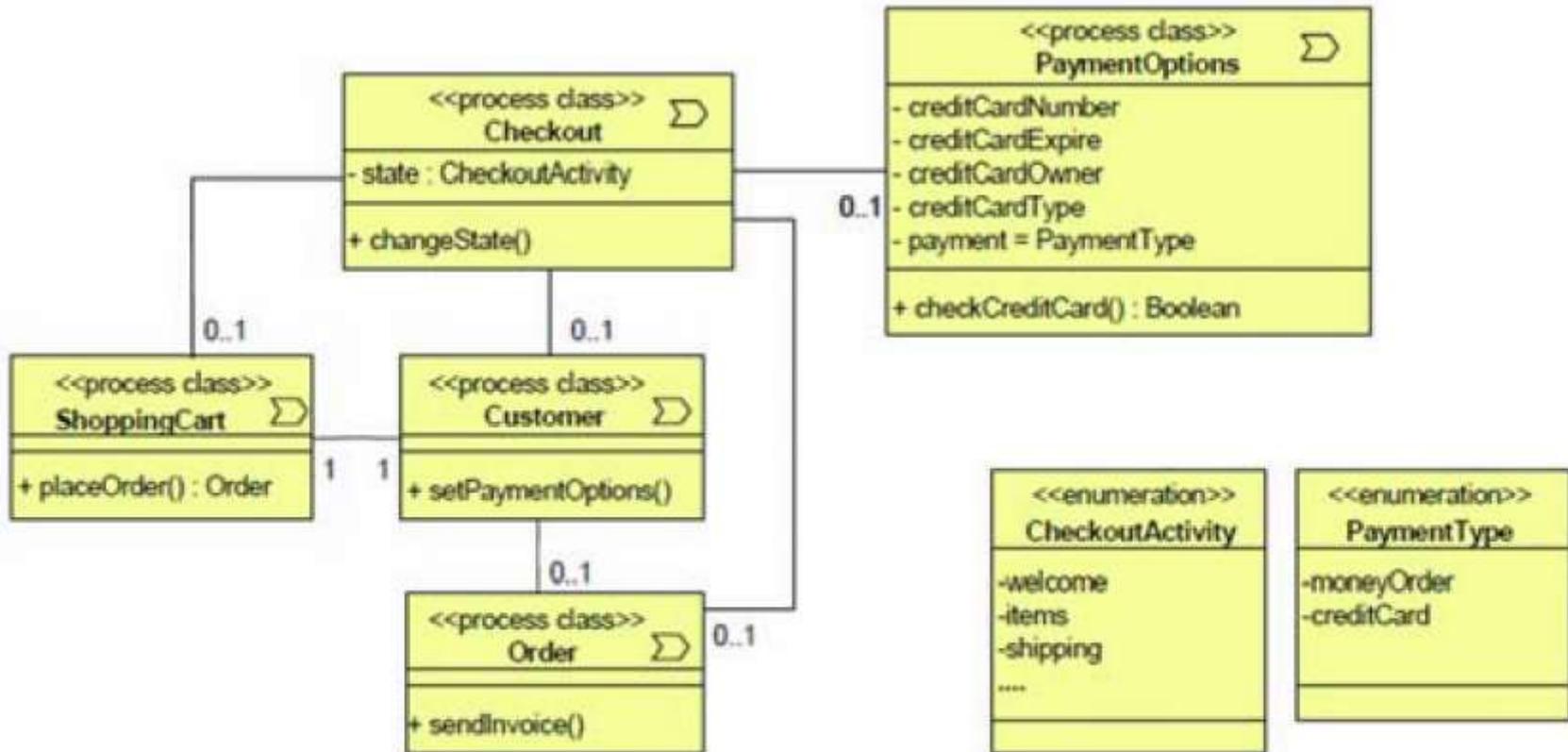
Modelo de estructura de navegación UWE

## UWE

### 6 • Modelo estructural de proceso

- Define las clases involucradas en la implementación de un proceso de negocio

UWE



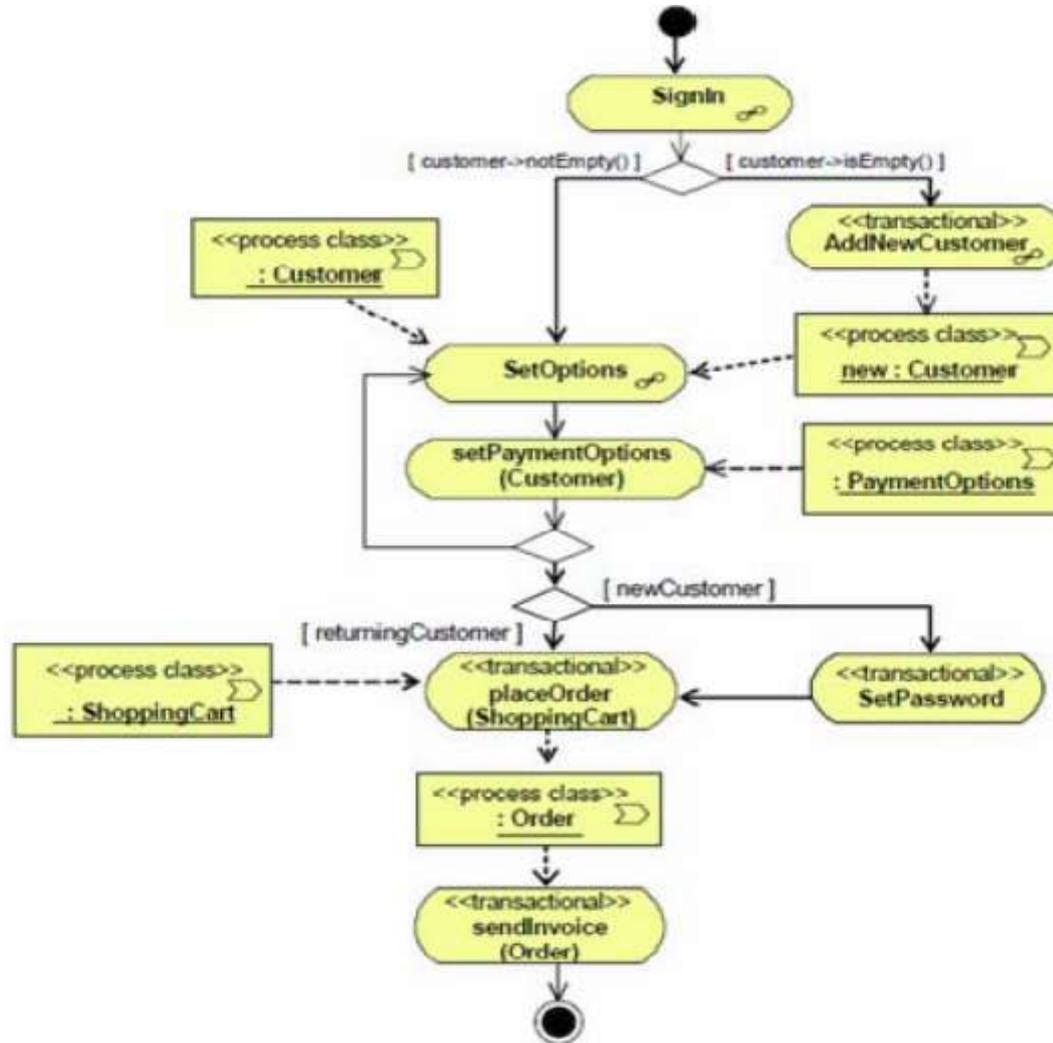
Modelo estructural de proceso UWE

## UWE

### 7 • Modelo de flujo de proceso

- Especifica el flujo de invocación de los objetos que implementan los procesos de negocio
- Utiliza un diagrama de actividades en vez de la aproximación clásica UML basada en diagramas de interacción (secuencia o comunicación)

Ingeniería Web



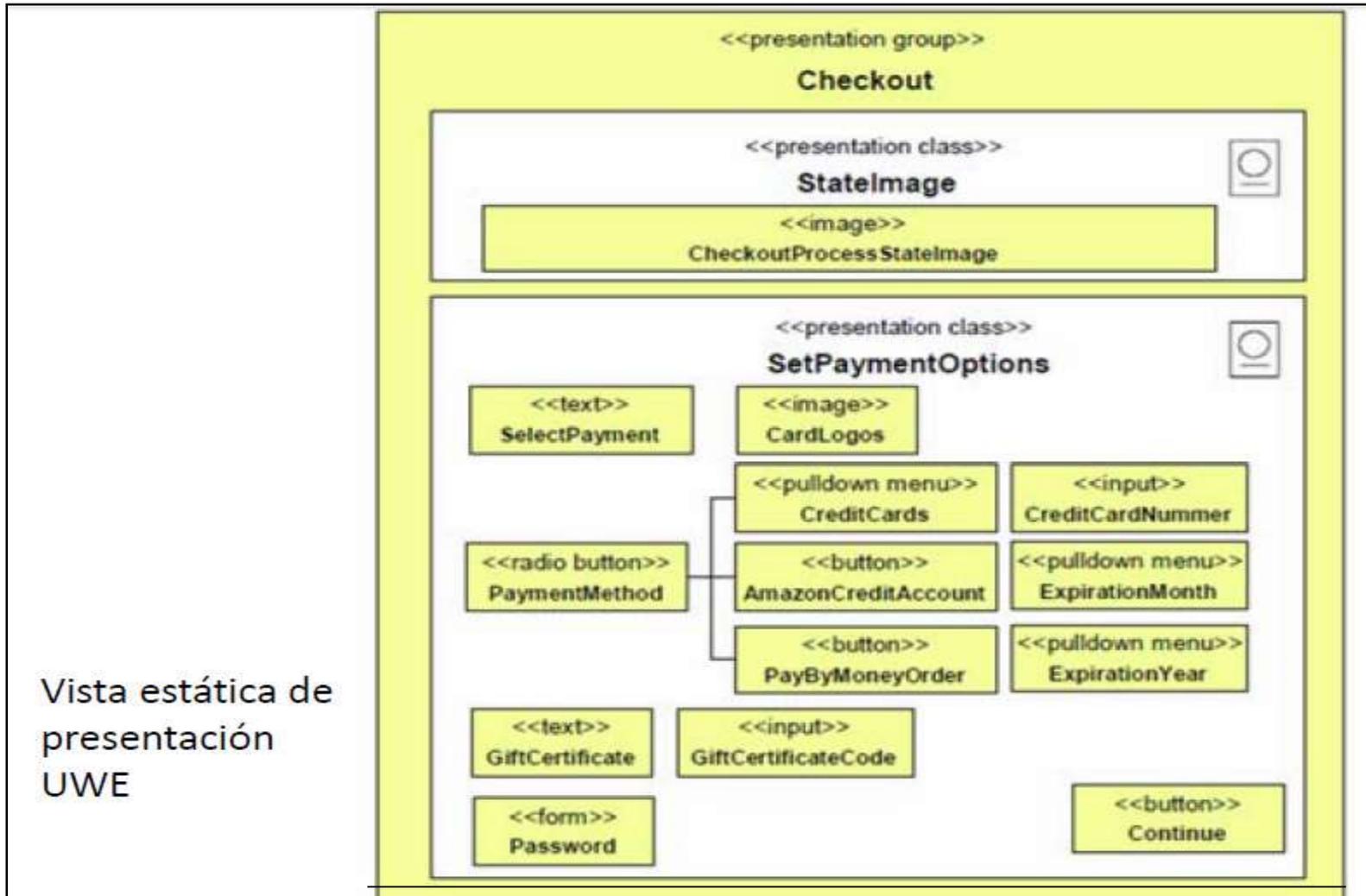
Modelo de Flujo de Proceso para el ckeckout

## UWE

### 8 • Modelo de estructura de presentación

- Define los elementos estructurales de la interfaz de usuario
- Utiliza estereotipos:
  - `presentational class`
  - `frameset`
  - `text`
  - `audio`
  - `video`
  - `anchor`
  - **etc.**

UWE

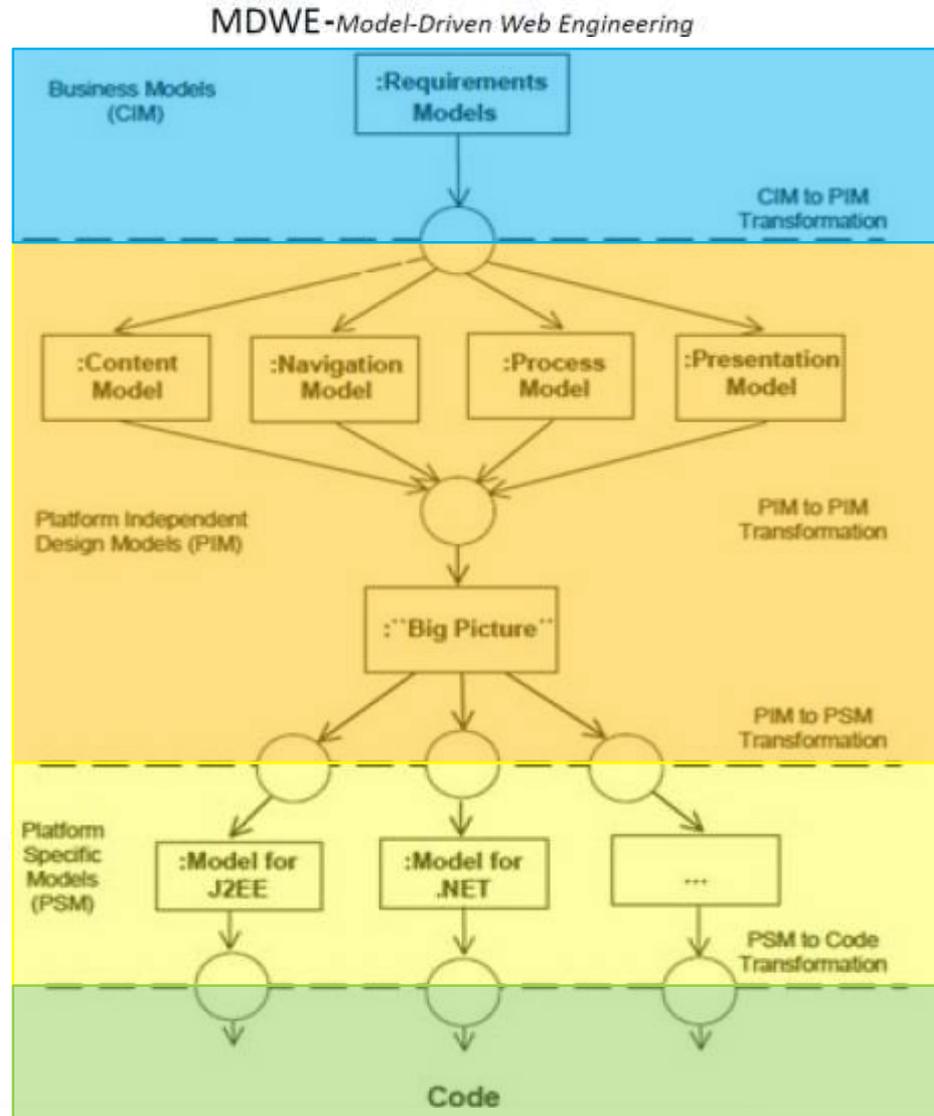


Vista estática de  
presentación  
UWE

## MDWE - *Model-Driven Web Engineering*

- La *Model-Driven Web Engineering* es un ejemplo de MDD
- Como vimos MDD es un nuevo paradigma que **da más importancia a los conceptos que a la forma de representarlos.**
- La idea es **detectar los conceptos y representarlos de manera**
- **abstracta.**
- Luego se definen las “**relaciones**” entre estos conceptos.
- **Basándose en los conceptos** se pueden definir “**transformaciones**”.

Ingeniería Web



El Modelo propuesto por MDWE

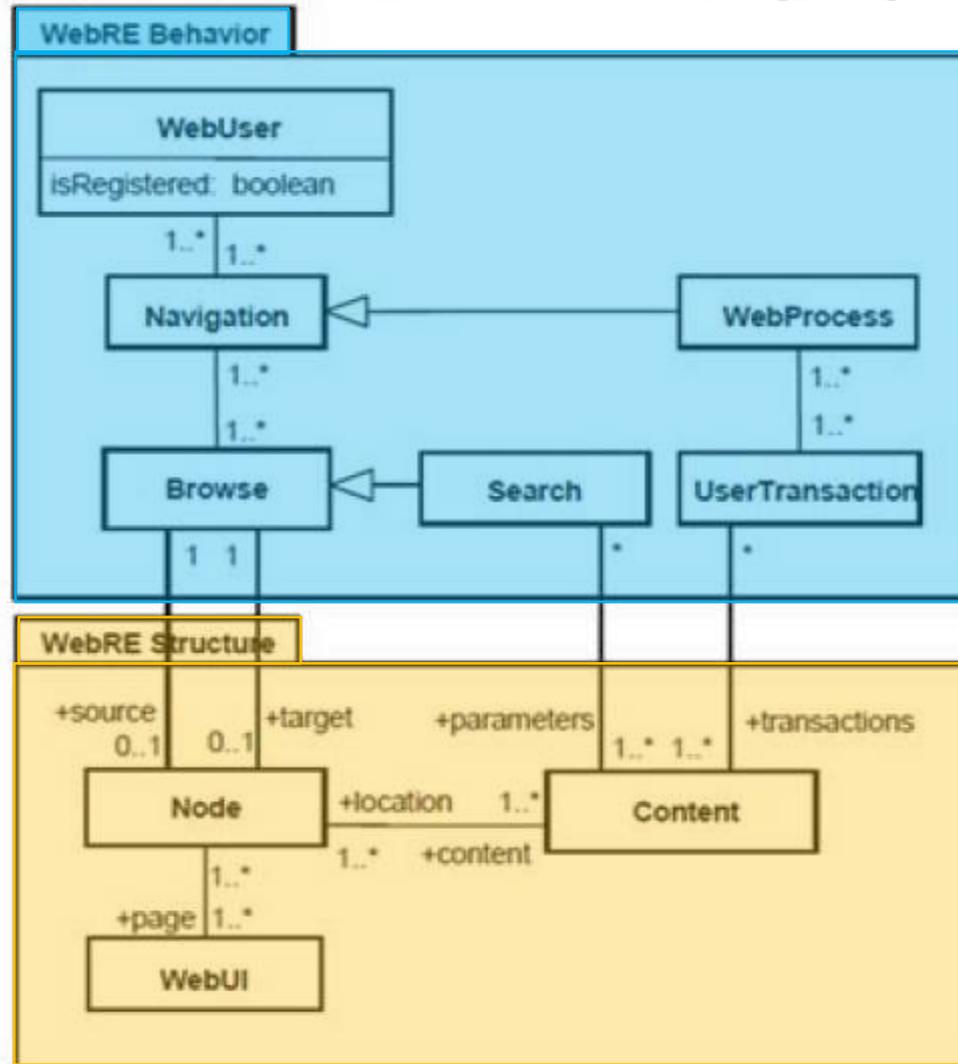
MDWE - *Model-Driven Web Engineering*

## Un ejemplo de MDWE

- WebRE es una aproximación para la representación de requisitos
  - Tiene un metamodelo que lo define
  - Tiene un perfil UML para su uso

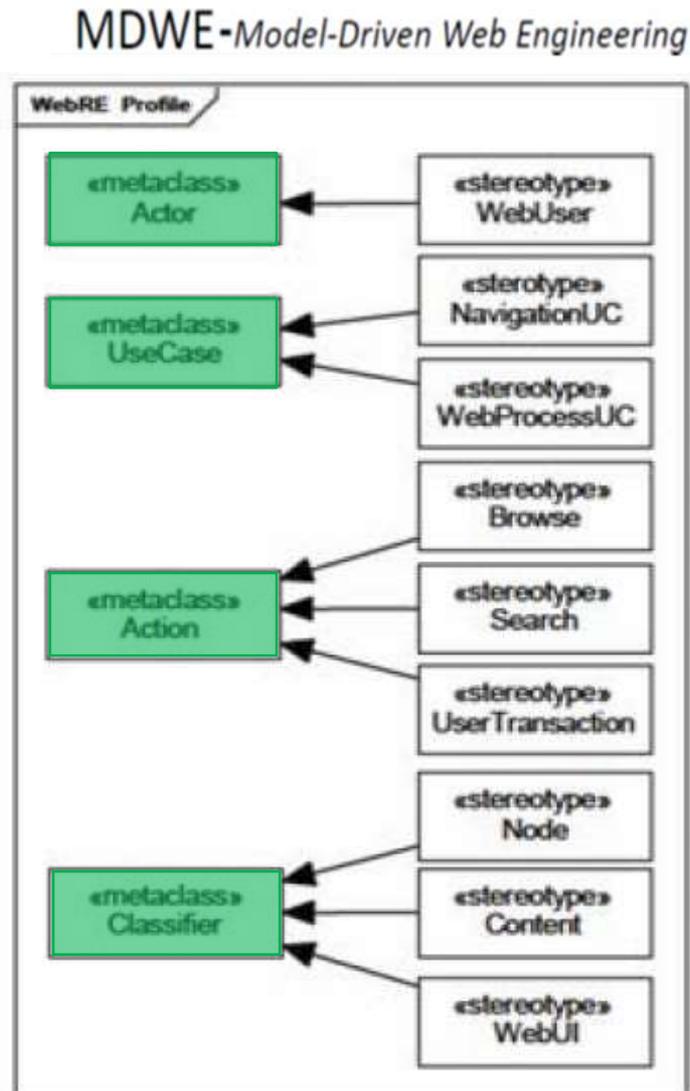
1° -Definir los conceptos

MDWE - Model-Driven Web Engineering



Metamodelo WebRE

## 1° -Definir los conceptos



Perfil UML para el **Metamodelo WebRE**

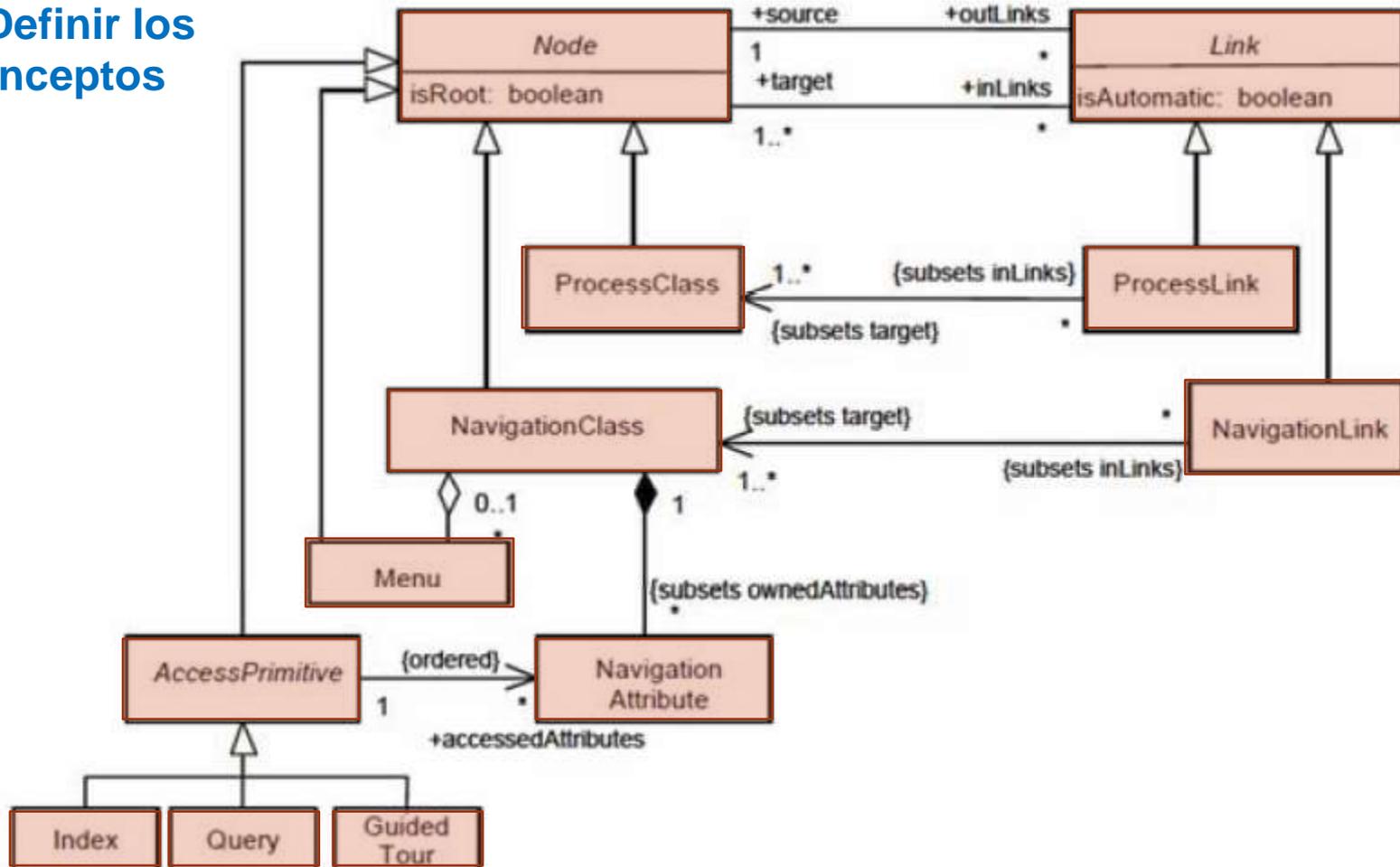
*MDWE-Model-Driven Web Engineering*

## Un ejemplo de MDWE

- UWE también tiene distintos metamodelos que lo caracterizan

MDWE - Model-Driven Web Engineering

1° - Definir los conceptos



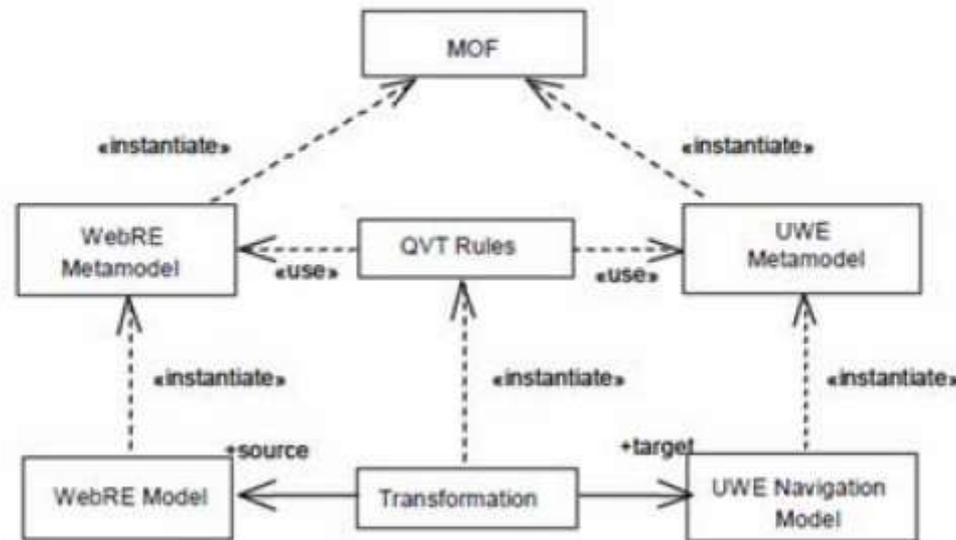
**Metamodelo UWE** para el modelo de estructura de navegación

## 2° -Definir las transformaciones

MDWE-Model-Driven Web Engineering

Un ejemplo de MDWE

- Se han definido transformaciones para relacionar ambas aproximaciones



Transformaciones entre WebRE y UWE

## 2° -Definir las transformaciones

### MDWE-*Model-Driven Web Engineering*

#### Un ejemplo de MDWE

```
transformation Content2Class (webre:WebRE, uwe:UWE) {  
  top relation R1 {  
    n: String;  
    checkonly domain webre c:Content { name = n };  
    enforce domain uwe cc: Class { name = n };  
  }  
  relation R2 {  
    cn: String;  
    checkonly domain webre p: Property { namespace=c:  
      Content {}, name = cn};  
    enforce domain uwe p1:Property { namespace = cc: Class{};  
      name = cn}  
    when {R1 (c,cc); }  
  }  
}
```

Transformación que convierte contenidos de requisitos WebRE en clases del dominio UWE

## 2° -Definir las transformaciones

### MDWE-*Model-Driven Web Engineering*

#### Un ejemplo de MDWE

```
transformation Browse2NavClass&Link (webre: WebRE, uwe: UWE) {
  top relation R3{
    c: String;
    checkonly domain webre b:Browse {target = no:Node
      {content = c} };
    enforce domain uwe nc: NavigationClass {};
    enforce domain uwe link: Link { isAutomatic = false,
      target = nc; };
    where { R4 (c,nc); }
  }
  relation R4 {
    n: String;
    checkonly domain webre c: Content {name = n };
    enforce domain uwe nc: NavigationClass {name = n };
  }
}
```

Transformación que convierte relaciones de navegación WebRE en relaciones de navegación UWE

MDWE-*Model-Driven Web Engineering*

Un ejemplo de MDWE

- Lo importante son los conceptos . La forma de “representar” los conceptos ya no es tan relevante.
- Las relaciones permiten “controlar” la consistencia de los modelos.
- Las transformaciones son una vía que se pueden traducir en “procesos” de transformación.