

Model Driven Developing (MDE)

Universidad de Carabobo. Ingeniería de Software.

Integrantes

Jesús Pelay,	Gabriel Peraza,	Elio Ruíz
C.I: 26.804.859	C.I: 26.929.687	C.I: 27.877.261

Docente

Dinarle Ortega

Introducción

Ideas preliminares

Codificar resuelve problemas simples.

```
c = 1  
print( c + 1 )
```

Pero los problemas complejos requieren de estructuras flexibles, ajustadas a la necesidades de los clientes.

```
EngineBuilder eng = logic.engineFactoryBuilder();  
eng.build( abstract_model );  
// ... Y muchas cosas más ...
```

Y si trabajamos en equipo... ¿cómo se coordinan las tareas?

Se emplean modelos

¿Qué es un modelo?

Un modelo es una abstracción de algo con el propósito de entenderlo antes de construirlo. - [1]

- Dirigen la **atención** a los aspectos de interés.
- Describen el **contexto** del problema.
- Determinan los **alcances** y **detalles** del problema.
- Definen como se debe **implementar** un sistema.

Es decir, son una ***Abstracción*** del problema.

Modelado para el desarrollo de software.

Modelos como bosquejos

Comunicación + Vistas parciales = Vistas parciales de la especificación del sistema.

Modelos como planos

Descripción detallada de la especificación

Modelos como programas

Modelos en lugar de `código` → Desarrollo de sistemas.

MDSE

Model Driven Software Engineering

Principios de MDSE

Está constituido por:

- Conceptos.
- Notaciones.
- Procesos y Reglas.
- Herramientas.

Modelos + Transformaciones = Software

Muchos acrónimos similares MD*s

La diversidad de MD*s

1. MDD: Model-Driven Development. es un **paradigma** de desarrollo.
2. MDA: Model-Driven Architecture. es un **subconjunto** propuesto por le OMG.
3. **MDE**: Model-Driven Engineering. Es el **superconjunto** de MDD.
4. MBE: Model-Based Engineering. versión más flexible de **MDE**.

Metodología MDSE

Visión general

Objetivo de MDSE

- Dominios.
- plataformas.
- Espacios técnicos.
- Escenarios.

Lenguajes de modelado

- DSLs: Domain-Specific Languages.
por ejemplo → HTML
- GPMLs: General-Purpose Modeling Languages.
el más conocido → UML

Metodología MDSE

Visión general

Metamodelado

Transformaciones

Metodología MDSE

Visión general

Clasificación de los modelos

- **Modelos estáticos**

describen formas estructurales, la arquitectura del sistema y cómo se gestiona la información.

- **Modelos dinámicos**

Muestran la secuencia de ejecución de las acciones y los algoritmos, las colaboraciones entre los componentes, cambios de los estados internos, entre otros.

Adopción de MDSE en la industria.

 Grafico de adopcion

Metodología MDSE

Visión general

Soporte de herramientas

- Herramientas: Model-Based vs Programming-Based

Críticas a MDSE

Casos de uso de MDSE

1. Automatización del desarrollo de software.

- Generación de ``código``.
- Interpretación de *modelos*.
- Combinación de ambos enfoques.

2. Interoperabilidad del sistema

Arquitectura Dirigida por Modelos (MDA)

Propuesta creada por *The Object Magement Group* (OMG) para aplicar las prácticas del MDE dentro del desarrollo de sistemas.

Definiciones y suposiciones de MDA.

Sistema	Dominio del Problema	Espacio de Solución
Modelo	Arquitectura	Plataforma
ViewPoint	Vista	Transformación

Arquitectura Dirigida por Modelos (MDA)

Tres niveles de modelado:

- CIM: Computation-Independent Model.
- PIM: Platform-Independent Model.
- PSM: Plataform-Specific Model.



Mapeos

correspondencias entre los elementos de dos modelos distintos.

Lenguajes dentro del MDA

- DSLs: Domain-Specific Languages.
- GPMLs: General-Purpose Modeling Languages.

Los lenguajes de modelado

- Anatomía de los lenguajes de modelado.
- Lenguajes de Modelado de Propósito General vs Lenguajes de Modelado de Dominio Específico (DSL).
- Modelado de propósito general: El caso de UML.
 - Prácticas de diseño.
 - Diagramas de estructura (o diagramas estáticos).
 - Diagramas de comportamiento (o diagramas dinámicos).
 - Herramientas UML.
 - Críticas y evolución de UML.
- Visión general de las DSL.
 - Principios de DSLs.
 - Algunos ejemplos de DSLs.
- Definición de Restricciones de Modelado (OCL).

Transformaciones de modelo a texto

- Fundamentos de la generación de código basada en modelos.
- Generación de código mediante lenguajes de programación.
- Generación de código mediante lenguajes de transformación MT.
 - Ventajas de los lenguajes de transformación MT.

Gestión de modelos

- Comparación de Modelos.
- Gestión global de modelos.
- Calidad de los modelos.
 - Verificación de modelos.
 - Prueba y validación de modelos.

Referencias

- [1] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen. Object-Oriented Modeling and Design. Prentice Hall, Englewood Cliffs, New Jersey, USA, 1991.
- [2] Jordi Cabot, and Manuel Wimmer. Model-Driven Software Engineering in Practice Marco Brambilla, 2012.