Modellazione di architetture software

Lezione 13 di Ingegneria del software

introduzione a UML e MDA

Docente: Giuseppe Scollo

Università di Catania, sede di Comiso (RG)
Facoltà di Scienze Matematiche, Fisiche e Naturali
Corso di Studi in Informatica applicata, AA 2009-10

Indice

1. Modellazione di architetture software
2. architettura nel progetto software
3. UML: cosa è, cosa non è
4. cenni preistorici: origini di UML
5. cenni storici: evoluzione di UML
6. pragmatica di UML, "prospettive"
7. MDA, *Executable* UML
8. modelli UML, metamodello di UML
9. tipi di diagrammi UML
0. valore di modelli UML:
1. subito all'opera: *quick start*
2. strumenti: dove cercare
3. strumenti: come scegliere
4. alcuni strumenti *open source*
5. una proposta di lavoro

architettura nel progetto software

- **progetto dell'architettura:** fase dello sviluppo, ma ...
- ... l'astrazione architetturale è **iterabile** a parti del sistema
- **utilità** delle architetture software:
    - *riconoscimento di **tratti comuni** a sistemi diversi*
    - *supporto ad **analisi**, valutazione di alternative, **decisioni***
    - ***riusabilità** per implementazioni differenti*
    - ***documentazione** ad un livello di astrazione appropriato*

UML: cosa è, cosa non è

- notazione "**grafica**" per la modellazione di sistemi "**software**" ad "**oggetti**"
  "..." = non solo ...
- linguaggio di modellazione **unificato** : uno **standard**
- **modelli UML** : **descrizioni** di sistemi, da realizzare o esistenti
- UML è uno **strumento di comunicazione**
- UML **non è** una metodologia di sviluppo né di documentazione del software
  *ma può ben essere adoperato in tali **contesti***
- UML + metodologie → strumenti

cenni preistorici: origini di UML

- anni '60: **classe, oggetto** in **Simula 67**
- anni '70: **incapsulamento, ADT**
- anni '80: linguaggi OO "crescono": **SmallTalk, C++**
- anni '90: **proliferazione** dei "metodi OO"

  *ci si metta d'accordo almeno sulla **notazione!***
- **OOPSLA'95**

  OOAD (Booch) + OMT (Rumbaugh) + Objectory (Jacobson)
  → UML 0.8 (Rational Inc., ora IBM)

cenni storici: evoluzione di UML

1997: **OMG (Object Management Group)**:

UML 1.0 (proposta Rational) →
UML 1.1 (standard, OCL ⊆ UML 1.1)

uno standard è "buono" quando rende tutti "egualmente infelici" (!?)
dicembre 1998: UML 1.2 (cosmesi)
giugno 1999: UML 1.3 (semantica dei diagrammi di attività)
maggio 2001: UML 1.4 (profili, elaborati (componenti))
marzo 2003: UML 1.5 (semantica delle azioni)
luglio 2005: UML 2.0 (molte innovazioni)
febbraio 2009: UML 2.2 (la versione ufficiale più recente)

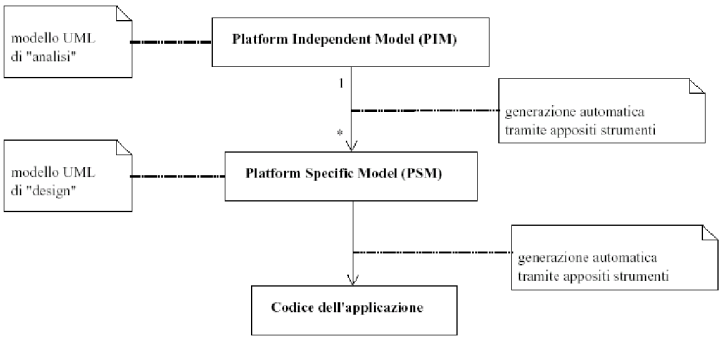
pragmatica di UML, "prospettive"

perché si usa UML:

- rappresentazione del sistema a vari **livelli di astrazione:**
  bozza, *progetto dettagliato*, *eseguibile*
- **"prospettive":**
  concettuale, *specifica*, *implementazione*
- >→ **forward engineering** (costruzione)
- ←< **reverse engineering** (documentazione)
- ←→ **round-trip engineering** (evoluzione)

MDA, *Executable* UML

MDA: *Model Driven Architecture* (standard OMG)
scopo: costruzione automatica del software da modelli UML



tratta da: Adriano Comai (2006) Introduzione a UML, http://www.analisi-disegno.com

l'uso di UML **non** implica quello di MDA
***Executable* UML:** no PSM, sottolinguaggio compilabile di UML

modelli UML, metamodello di UML

UML definisce:

- una **notazione** (grafica):

  *elementi di **sintassi** dei modelli*

- un **metamodello** di UML stesso:

  ***sintassi astratta** del linguaggio*

  +

  ***regole** (semantiche?) in OCL che la vincolano*

  +

  ***rappresentazioni** grafiche degli elementi astratti*

tipi di diagrammi UML

diagrammi **strutturali:**

delle classi
degli oggetti
dei componenti
di struttura composita
di deployment
di package

diagrammi **comportamentali:**

dei casi d'uso
delle attività
di macchina a stati
*di **interazione:***

di sequenza
di comunicazione
di interazione generale
di temporizzazione

valore di modelli UML

un modello UML può aver valore: descrittivo, prescrittivo, normativo

nel **forward engineering** la **prospettiva** semantica di un modello UML ne determina approssimativamente il **valore**:

concettuale ⇒ *descrittivo*

di specifica o implementazione ⇒ *prescrittivo*

nel **reverse engineering** il valore è **descrittivo**

qual è il valore dello standard UML?

normativo = prescrittivo & universale

sì, però con molti contenuti descrittivi e norme non vincolanti, ad esempio vale la regola generale che in un modello UML

qualsiasi informazione può essere soppressa

N. B.: spesso, questa regola può dar luogo ad **ambiguità**, ad es. quando sono definiti dei valori di **default**: l'assenza significa soppressione o presenza del valore di default?

subito all'opera: *quick start*

perché adoperare un testo tanto "striminzito" che l'autore lo intitola "distillato" dello standard?

una risposta non unica: la **legge di Pareto**

esercizio: determinare altre buone ragioni per tale scelta

oltre alla documentazione ufficiale (http://www.uml.org) e ai testi consigliati, può essere utile consultare:
Linee Guida UML - Modelli Architetturali
http://www.analisi-disegno.com/StrumentiOperativi.htm
di Adriano Comai, di orientamento pragmatico, rivolto alla pratica progettuale

il laboratorio vi aspetta...

BOUML è già installato! Per saperne di più: http://bouml.free.fr/doc

strumenti: dove cercare

catalogazione e valutazione comparativa di strumenti di sviluppo di modelli UML:

http://www.objectsbydesign.com/tools/umltools_byPrice.html
N.B. la lista è ordinabile anche in base ad altri criteri, oltre al costo

http://uml-directory.omg.org/vendor/list.htm
catalogo di "fornitori UML" curato dall'OMG

http://www.oose.de/service/uml-werkzeuge.html
pagina bilingue, tavola sinottica del supporto dei tipi di diagrammi UML

http://www.cetus-links.org/oo_ooa_ood_tools.html
sito ricco di informazione ma un po' "datato"

http://www.jeckle.de/umltools.html
tavola sinottica in tedesco, ma con un articolo (in inglese) di valutazione analitica di alcuni degli strumenti considerati:
http://www.jeckle.de/mad_about_modelling.html

strumenti: come scegliere

occorre determinare i **criteri di valutazione**, che naturalmente dipendono da vari fattori: finalità di progetto, contesto operativo, vincoli economici, etc.

un buon punto di partenza può essere offerto da una **checklist**, dalla quale selezionare i criteri più rilevanti al proprio caso, v. ad esempio:

http://www.objectsbydesign.com/tools/checklist.html

un **caso concreto**, la scelta di uno o più strumenti di sviluppo di modelli UML, installabili in laboratorio **per questo corso:**

finalità: *formativa*
contesto operativo: *s.o. GNU Linux/Debian*
vincolo di costo: *0*

per la finalità formativa si richiede:

supporto di tutti i diagrammi di UML 2.0
buona disponibilità di documentazione

alcuni strumenti open source

per quanto estese, le liste di strumenti di supporto allo sviluppo di modelli UML indicate sopra non sono complete

ad esempio, in nessuna di esse si fa menzione di **Gaphor**, uno strumento *open source* (in via di sviluppo): http://gaphor.sourceforge.net

> **N.B.** il testo di (Damiani et al., 2007) lo valuta "interessante e completo", però non tutti i diagrammi UML sono supportati allo stato attuale

inoltre, sebbene la maggior parte degli strumenti *open source* siano disponibili per Unix/Linux o multi-piattaforma, ne esistono anche di specifici per altri sistemi operativi, ad esempio **StarUML** (MS Windows):

> http://staruml.sourceforge.net
>
> **N.B.** non è incluso nella lista di strumenti al sito Object by Design, tuttavia ha un *thread* nel Forum UML Tools mantenuto in tal sito

una proposta di lavoro

la valutazione di strumenti di sviluppo per modelli UML è

> non solo un necessario preliminare alla scelta per lo sviluppo di modelli UML
> ma anche un utile esercizio di formazione professionale

si avanza quindi la seguente proposta di lavoro, oggetto di valutazione in itinere ***(bonus ;)***

1. compilazione di una **lista di criteri di selezione e di valutazione** di strumenti di sviluppo di modelli UML, motivata dalle proprie finalità formative e condizioni operative di sviluppo progettuale
2. **selezione** di uno o più strumenti *open source* di sviluppo di modelli UML, in base ai criteri di selezione di cui al punto precedente, per la successiva valutazione
3. produzione di una **scheda di valutazione** per ciascuno degli strumenti selezionati, adoperando i criteri di valutazione precedentemente definiti

buon lavoro!