

INTRODUZIONE AD

UML

Unified Modeling Language

**Relazione di Sistemi
Informativi**

*Mantova 13/03/2006
Author : Daniele Caprini*

Cosa è L'UML?

Nel mondo costantemente in fermento ed in evoluzione quale è quello dello sviluppo di applicazioni Object Oriented, diviene sempre più difficile costruire e gestire applicazioni di alta qualità in tempi brevi.

Da tale difficoltà e dall' esigenza di avere un linguaggio universale per modellare gli oggetti che, potesse essere utilizzato da ogni industria produttrice di software, fu inventato il linguaggio **UML**, *Unified Modeling Language*..

La forza dell'UML consiste nel fatto che il processo di disegno del sistema può essere effettuata in modo tale che i clienti, gli analisti, i programmatori e chiunque altro sia coinvolto nel sistema di sviluppo possa capire ed esaminare in modo efficiente il sistema e prendere parte alla sua costruzione in modo attivo.

I diagrammi UML non fanno altro che documentare in modo esauriente e a tutti i livelli le caratteristiche tecniche che dovranno essere implementate.

E' importante avere chiaro fin dall'inizio il punto di vista utilizzato nello sviluppo di un particolare modello del sistema in esame.

Tre sono i differenti punti di vista:

- ✚ **concettuale** : consente di costruire un modello del sistema in esame rappresentando i concetti del dominio dell'utente;
- ✚ **di specificazione** : è il primo passo verso il software, ma ci si occupa del software solo a livello di interfacce. Ci si muove non più nel dominio utente, ma nel dominio applicativo;
- ✚ **implementativo**: tale punto di vista consente di descrivere l'implementazione del software.

UML non distingue tra i tre punti di vista, permettendo di descrivere l'applicazione in ognuno di essi. Uno dei punti di forza sta proprio in questa flessibilità.

Lo svantaggio è comunque ovvio: se non si ha chiara la distinzione tra i tre punti di vista, è possibile costruire un modello derivante dall'analisi non buono, cioè un modello di analisi che, anziché concentrarsi sull'analisi del problema in esame, contiene già elementi di soluzione.

Quando nacque l'UML?

Durante gli anni '90 furono introdotte nel mercato dell'Information Technology parecchie metodologie per il disegno e la progettazione di sistemi software.

Ognuna di queste tecnologie aveva il suo insieme proprio di notazioni e simboli, che differiva, a volte in modo rilevante, dalle altre.

Le tre principali metodologie erano:

- + OMT (Rumbaugh)
- + Booch 1991
- + OOSE (Jacobson)

Ognuno di questi metodi aveva, naturalmente, i suoi punti di forza e i suoi punti deboli. Ad esempio, l'OMT si rivelava ottimo in analisi e debole nel disegno. Booch 1991, al contrario, eccelleva nel disegno e peccava in analisi. Jacobson aveva il suo punto di forza nell'analisi dei requisiti e del comportamento di un sistema ma si mostrava debole in altre aree.

Successivamente, tutti gli autori cercarono di apportare delle modifiche tali da migliorare sempre più la propria metodologia, basandosi sui punti di forza altrui.

In sostanza, i tre metodi stavano convergendo verso un'unica visione che incorporasse le qualità migliori che ognuno di essi aveva mostrato.

Restava era il fatto che ogni metodo portava ancora con sé la propria notazione. Tale problema non era da sottovalutare in quanto l'uso di simbologia differente portava facilmente a confusione sul mercato a causa del fatto che un determinato simbolo poteva avere un significato differente per analisti e disegnatori diversi.

Finalmente, dopo un periodo di tempo in cui andò avanti la cosiddetta "*guerra della metodologia*" ("method war") ci si rese conto che era assolutamente necessario produrre uno standard che unificasse anche la notazione utilizzata.

Fu così che, nell'Ottobre del 1995, nacque la prima bozza dell'**UML**, ovvero l'unificazione delle notazioni e delle idee prodotte da Booch, Rumbaugh e Jacobson per modellare un sistema software. La prima versione ufficiale, prodotta dall'OMG (Object Management Group) fu rilasciata nel Luglio del 1997 e nel Novembre dello stesso anno l'UML venne adottato come standard.

Tra tutti i benefici derivanti dall'UML ricordiamo:

1. Un sistema software grazie al linguaggio UML viene disegnato professionalmente e documentato ancor prima che ne venga scritto il relativo codice, da parte degli sviluppatori. Si potranno così conoscere in anticipo il risultato finale del progetto su cui si sta lavorando.
2. La scrittura del codice stessa è resa più agevole ed efficiente oltre al fatto che in tal modo è più facile scrivere del codice riutilizzabile in futuro. I costi di sviluppo, dunque, si abbassano notevolmente con l'utilizzo del linguaggio UML.
3. È più facile prevedere e anticipare eventuali "bugs" nel sistema. Il software che si scrive, si comporterà esattamente come ci si aspetta senza spiacevoli sorprese finali.
4. L'utilizzo dei diagrammi UML permette di avere una chiara idea, a chiunque sia coinvolto nello sviluppo, di tutto l'insieme che costituisce il sistema. In questo modo, si potranno sfruttare al meglio anche le risorse hardware in termini di memoria ed efficienza, senza sprechi inutili o, al contrario, rischi di sottostima dei requisiti di sistema.
5. Grazie alla documentazione del linguaggio UML diviene ancora più facile effettuare eventuali modifiche future al codice. Questo, ancora, a tutto beneficio dei costi di mantenimento del sistema.

La comunicazione e l'interazione tra tutte le risorse umane che prendono parte allo sviluppo del sistema è molto più efficiente e diretta. Parlare la stessa "lingua" aiuta ad evitare rischi di incomprensioni e quindi sprechi di tempo.

Utilizzo di UML

L'UML può essere usato per:

1. **Pensare.**
Nel corso delle attività di sviluppo o di evoluzione di un sistema, UML può servire agli sviluppatori per ragionare sui problemi e sulle soluzioni.
2. **Comunicare (e documentare).**
Tra soggetti diversi, distanti nello spazio (ad esempio da analisti separati fisicamente dai progettisti e dagli sviluppatori) o nel tempo (posso trovarmi a riprendere in mano modelli che io stesso ho creato in passato). Tra aziende diverse, che devono collaborare su basi contrattuali.

Naturalmente, c'è qualche differenza tra l'utilizzo di UML per pensare e per comunicare. Se lo uso per ragionare, posso permettermi di fare solo i diagrammi che mi servono al momento, al livello di dettaglio e di precisione che mi interessa, senza pormi problemi di comunicazione. Se invece lo uso per comunicare e documentare, è necessario che chiarisca prima di tutto con chi voglio comunicare, che cosa esattamente voglio comunicare, e a quale livello di dettaglio e di precisione.

Utilizzi scorretti di UML

La prima causa è legata alla complessità di UML, ed alla difficoltà di raggiungere un buon livello di competenza. I concetti che costituiscono il linguaggio sono il risultato di decenni di ingegneria del software, ed hanno significati precisi. Utilizzare la notazione senza conoscere bene i concetti rappresentati porta a risultati inefficaci e a perdite di tempo. In molti casi manca una formazione adeguata. Spesso vengono realizzati molti più diagrammi rispetto a quelli che sarebbero stati necessari; altre volte ne mancano di essenziali. Bisogna ricordare che ogni progetto ha le sue caratteristiche specifiche: è importante (tempi, costi, efficacia) che UML venga utilizzato nel modo migliore nello specifico contesto.

Il secondo motivo che provoca utilizzi scorretti è che UML non è, allo stato attuale, verificabile in modo automatico (non è "compilabile"). Gli errori, anche gravi, che un utilizzatore può compiere non sono rilevabili se non a fronte di verifiche condotte da chi conosca bene il linguaggio.

Nell'evoluzione dello standard UML stanno assumendo un peso sempre maggiore le "well-formedness rules" (regole per la definizione di costrutti ben formati), che vincolano e precisano gli aspetti semantici e sintattici che regolano la produzione di modelli corretti. Purtroppo si tratta di aspetti poco considerati, non tanto dagli utilizzatori del linguaggio (che non sono tenuti a conoscerne gli aspetti più sofisticati), quanto dai produttori di strumenti.