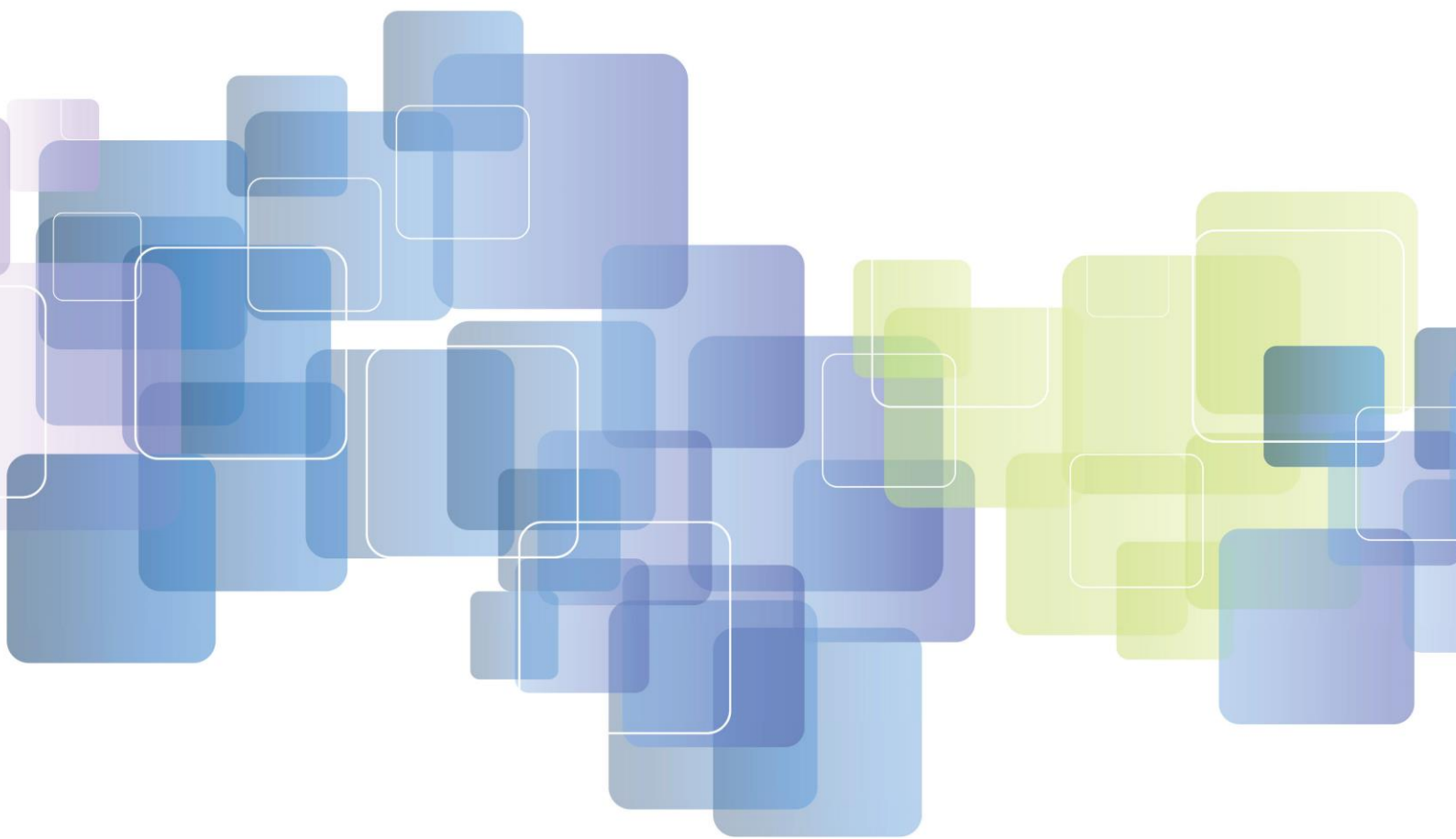


 **TecnetDati**

UML

un linguaggio universale per la modellazione del software

Adriano Comai



Finalmente uno standard per l'analisi e disegno OO?

L'obiettivo è ambizioso. Lo Unified Modeling Language (UML) vuole essere, secondo le intenzioni dei suoi autori, Booch Jacobson e Rumbaugh, "il linguaggio per la creazione di modelli software". Un linguaggio, cioè, "universale per la progettazione (modeling) dei sistemi, nel senso che può esprimere modelli di varia tipologia e creati per obiettivi diversi, proprio come un linguaggio di programmazione o un linguaggio naturale possono essere usati in molti modi diversi".

Obiettivo ambizioso. Ma probabilmente realistico. Perché Grady Booch, Ivar Jacobson e Jim Rumbaugh, gli autori dell'Unified Modeling Language, sono fra i più quotati metodologi Object Oriented, e il loro sforzo congiunto per la definizione dell'UML ha funzionato come un catalizzatore, attirando i contributi di altri metodologi di primaria importanza come Coad, Yourdon, Odell.

Perché l'UML?

Perché nel campo delle metodologie di analisi e disegno object oriented la confusione regna ancora sovrana. Qualcuno ne aveva contate più di cinquanta, ciascuna con i propri modelli ed i propri formalismi. Sebbene le più diffuse non superino la decina, non è facile per le aziende intenzionate a muoversi nel settore OO scegliere la più adeguata. La confusione esistente a livello metodologico si ripercuote poi a livello di strumenti CASE per la progettazione object oriented: sebbene l'orientamento prevalente tra i produttori sia quello di supportare notazioni diverse, l'assenza di uno standard riconosciuto rende rischiosa ogni valutazione di investimento in prodotti di questo genere. In questo confuso panorama, le proposte metodologiche avanzate individualmente da Booch, Rumbaugh e Jacobson risultavano già tra le più diffuse, il che costituisce un fattore positivo per le prospettive di accettazione dell'Unified Modeling Language. Il "progetto UML" nasce nel 1994, con l'ingresso di Jim Rumbaugh nella società di Grady Booch, la Rational, e si perfeziona con l'acquisizione da parte della Rational, a fine 1995, dell'azienda svedese Objectory, in cui opera Ivar Jacobson.

Caratteristiche dell'UML

La prima versione dell'UML (0.8), resa pubblica da Booch e Rumbaugh nell'ottobre 1995, portava il titolo di "Unified Method". La versione attuale (0.9), uscita nel luglio 1996, a cui ha contribuito anche Jacobson, porta la nuova denominazione. Perché questo cambiamento? Perché l'UML intende essere, appunto, un linguaggio universale mediante il quale descrivere i diversi modelli prodotti nel corso dello sviluppo software, a prescindere dalle procedure utilizzate per realizzarli. Booch Jacobson e Rumbaugh precisano che definire "un unico processo per tutti gli stili di sviluppo non sembra possibile e neppure desiderabile. [...] Comunque, l'UML può venire utilizzato per esprimere gli output di tutti i diversi processi, cioè i modelli che vengono prodotti".

Le fondamenta dell'UML sono costituite da un metamodello, che definisce le caratteristiche e le relazioni esistenti tra le diverse componenti di un progetto software

(es. classi, attributi, procedure, moduli, ...). Il metamodello costituisce la base per l'implementazione dell'UML da parte dei produttori di strumenti per lo sviluppo (CASE tools, ambienti visuali), e per l'interoperabilità tra i diversi strumenti. L'intenzione degli autori è di produrre una specifica formale del metamodello al momento della sottomissione dell'Unified Modeling Language all'OMG per la sua approvazione come standard (fine '96 - inizio '97).

Sulla base di quanto definito nel metamodello, vengono poi proposti alcuni modelli diagrammatici:

Livello LOGICO	Diagramma dei casi d'uso (Use Case Diagram)
	Diagramma delle classi (Class Diagram)
	Diagramma di sequenza (Sequence Diagram)
	Diagramma di collaborazione tra oggetti (Collaboration Diagram)
	Diagramma di transizione di stato (State Diagram)
Livello FISICO	Diagramma delle componenti software (Component Diagram)
	Diagramma di allocazione delle componenti (Deployment Diagram)

I modelli diagrammatici presentano ovviamente una notazione unificata, frutto di un compromesso tra quelle originarie proposte singolarmente da Booch Jacobson e Rumbaugh, e già adottate dalla maggior parte degli strumenti CASE Object Oriented.

In realtà, il progetto dei tre metodologi non limita l'utilizzo dell'UML al solo ambito Object Oriented. Il loro obiettivo era quello di definire un linguaggio di progettazione che fosse in grado di supportare anche linguaggi e componenti non OO, come file e moduli software tradizionali, ed ogni tipo di componente software a livello fisico. Anche in questo caso i risultati non si sono fatti attendere, in quanto Microsoft e Hewlett Packard hanno aderito al "progetto UML", che acquista così una ulteriore capacità di attrazione anche per le realtà, praticamente tutte, non ancora strettamente object oriented.

Quale sarà l'impatto dell'UML?

Il primo risultato dell'UML, già raggiunto almeno in parte, è la riduzione della confusione nel campo delle metodologie di sviluppo object oriented. E' estremamente probabile che l'UML diventi un vero e proprio standard, visto che l'Object Management Group (OMG), un consorzio che raggruppa tutti i produttori di software OO, ha chiesto alla Rational (la società di Booch Jacobson e Rumbaugh) di sottoporre l'UML ad una valutazione mirata a farlo diventare standard OMG. Un secondo risultato prevedibile è un'accelerazione nella diffusione degli strumenti CASE

object oriented, come conseguenza dell'emergere di modelli e di una notazione ampiamente accettati come standard.

E' la fine dei flow chart?

Ma cosa cambierà nel mondo reale delle aziende che producono software? Naturalmente è ancora presto per dirlo. In molte realtà la produzione di software è ancora basata su una definizione delle specifiche di analisi in formato testuale (o addirittura vocale...), e sulla loro immediata implementazione in righe di codice. In altre, il flow chart costituisce da sempre l'unica forma di documentazione. In altre ancora, l'avvento e l'utilizzo dei linguaggi visuali ha fatto pensare che la documentazione del sistema fosse un'attività ormai trascurabile. Ma per tutte le aziende in cui esiste la consapevolezza del fatto che lo sviluppo e la manutenzione del software sono attività complesse, la definizione e la condivisione di modelli formali per le specifiche di analisi e disegno è un requisito preliminare a ogni produzione di codice. Le aziende più attente, che da anni documentano i propri sistemi utilizzando modelli derivati dall'approccio strutturato, erano rimaste un po' disorientate, nel passaggio a tecniche di sviluppo orientate agli oggetti, dalla confusione riguardante le scelte metodologiche e gli aspetti di documentazione. In questi contesti, la standardizzazione portata dall'UML, supportata dai futuri strumenti di sviluppo, costituirà senza dubbio un elemento positivo di razionalizzazione.



Tecnet Dati s.r.l.
C.so Svizzera 185 -
10149 - Torino (TO), Italia
Tel.: +39 011 7718090 Fax.: +39 011 7718092
P.I. 05793500017 C.F. 09205650154
www.tecnetdati.com

