

Object Modelling Group



SOLUTA.NET

Versione:	0.1
Data di rilascio:	01/10/06
Stato:	final
Distribuzione:	
Nome del file:	OMGinAneheim.pdf

AUTORI

Name	Role	Date	Signature
Pierfranco Ferronato	Chief Architect	01/10/06	



PREMESSA

Le informazioni contenute in questo documento sono state verificate con attenzione e si ritengono esatte. Tuttavia, Soluta.net non si assume la responsabilità per qualsiasi inesattezza tecnica o per errori tipografici che possono essere contenuti nella presente. In nessun caso Soluta.net sarà ritenuta responsabile per eventuali perdite dirette, indirette, particolari, accidentali o per qualsiasi altro danno causato da errori, omissioni, errori di stampa od interpretazioni che si trovano in questa pubblicazione, anche se esplicitamente avvertita della presenza di errori od omissioni. Soluta.net è esente da ogni responsabilità nei confronti di chiunque per fatti, omissioni ed eventuali conseguenze in relazione al contenuto di questa pubblicazione.

CONTATTI

Soluta.net Via Vignola, 9 31037 Loria (TV) Italy
tel: + 39 0423-915547
info@soluta.net www.soluta.net

LICENZA

Questo lavoro è autorizzato con la "Creative Commons Attribution-ShareAlike License". Per visionare una copia di questa licenza, visitate <http://creativecommons.org/licenses/by-sa/1.0/> oppure inviate una lettera a "Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA".

E' possibile:

- copiare, distribuire, modificare e condividere il presente documento
- estrarne delle parti
- farne uso commerciale

Alle seguenti condizioni:

- Attribuzione. Deve essere citato l'autore del documento originale.
- Stessa condivisione. Se questo lavoro viene modificato, trasformato o utilizzato come base, il lavoro risultante deve sottostare a un Licenza simile a questa.
- Per ogni riutilizzo o distribuzione, devono essere indicati agli utenti i termini della licenza di questo documento.
- Si può rinunciare a ciascuna di queste limitazioni solo con il permesso esplicito dell'Autore.

Ad Anaheim in California si è appena concluso un technical meeting dell'Object Management Group (OMG) che è durato dal 25 al 29 Settembre. OMG, per coloro che non sapessero, è una organizzazione senza fini di lucro costituita nel 1989 dalle più grosse ed importanti industrie mondiali che si occupano di informatica, sia come sviluppatori di soluzioni o prodotti come IBM, sia come utenti evoluti dell'informatica come Boeing, Raytheon o NASA. Per fare una breve introduzione OMG è nata dalla volontà di industrie, teoricamente concorrenti, di definire degli standard comuni di interoperabilità. Standard che permettano ai loro clienti di facilitare l'integrazione tra sistemi host: in questo modo è nato ad esempio CORBA (Common Object Request Broker Architecture). A seguire poi, dalla "donazione" di Rational Rose, OMG si è presa carico anche di UML (Unified Modelling Language).

Oggi OMG è una grossa organizzazione ben strutturata composta da molte centinaia di membri, che conta circa 15 gruppi di lavoro i quali hanno prodotto e producono tutt'ora circa 170 standard¹. E' da tenere presente che ad oggi CORBA è stabile e che rappresenta il solo middleware nato al suo interno. In questo momento gli standard più rilevanti sono essenzialmente -oltre ai già citati CORBA e UML- MDA (Model Driven Architecture), XMI (XML Metadata Interchange), CWM (Common Warehouse Metamodel), SBVR (Semantic for Business Vocabulary and Rules), BPMN (Business Process Modelling Notation) e BPDM (Business Process Definition Metamodel). Recentemente RuleML.org (Rule Modelling Language) è confluita in OMG come anche BPML.org (Business Process Modelling Initiative).

L'atmosfera è brillante, i meeting si susseguono in continuazione tra una pausa caffè e l'altra, si lavora dalle 9 alle 19 e vi sono anche dei meeting alla domenica. Le discussioni dei gruppi di lavoro sono regolamentate da un chair -presidente- ed in alcuni casi agevolate da un co-chair, che si occupa di far rispettare l'agenda con molta precisione. La parte consistente del lavoro di collaborazione dei gruppi OMG si svolge soprattutto nelle mailing list in cui i membri collaborano alla definizione degli standard durante tutto il corso dell'anno, nei meeting invece, che si tengono ogni 3 mesi circa, i gruppi si riuniscono per concordare la chiusura delle versioni. Il processo di definizione degli standard OMG è molto formale e strutturato, prevede l'impegno di diversi gruppi di lavoro, mentre i tipi di documenti che man mano sono pubblicati determinano il grado di avanzamento di una specifica. I comitati di OMG essenzialmente si suddividono in Domain Task Force e Platform Task Force; i primi sono responsabili di standard relativi a specifici settori di business come sanità, finanza, robotica, mentre i secondi sono impegnati con specifiche orizzontali al business, ad esempio UML, SBVR, MDA, XMI, CWM. Gli stessi membri possono essere di queste due tipologie. La struttura di governo in OMG è l'AB (Architecture Board) che sovrintende a tutte le specifiche, ne avvalta l'inizio ed il rilascio della specifica finale. Degli undici componenti l'AB, dieci sono eletti dai membri Domain, Platform e Contributing di OMG, mentre uno è il rappresentante dello staff di OMG (AB chair). Cinque componenti sono eletti ogni anno e ogni membro dell'AB rimane in carica per due anni. OMG è una struttura paritetica, ogni membro ha gli stessi diritti degli altri e può aver diritto di voto (è una scelta del membro); una azienda piccola come la mia Soluta.net, ha lo stesso potere di voto di IBM.

Ma oltre agli aspetti formali in OMG partecipano aziende con diversi interessi. Vi sono le organizzazioni, compresi centri di ricerca ed università, che sono interessati come osservatori ad acquisire informazioni in prima linea sulle specifiche in corso di definizione, ma anche ad influenzarle poiché sono rilevanti al loro lavoro. L'interesse di tutti è ovviamente quello di concorrere ad avere uno standard aperto e riconosciuto sfruttando anche la competenza di tutti i membri. Nel caso di UML, ad esempio, i produttori di strumenti di modellazione, sono in prima linea per la definizione delle specifiche ed IBM lavora con No Magic, SparxSystem ed altri ancora migliorando l'interoperabilità tra loro e uniformando il linguaggio. In altri casi, vi sono organizzazioni che hanno definito degli standard nuovi ed hanno deciso di portarli in OMG per la standardizzazione, come è successo per UML e per xUML (Executable UML). In passato ad esempio Microsoft ha partecipato ai lavori come membro effettivo per definire il mapping CORBA-DCOM.

¹ http://www.omg.org/technology/documents/spec_summary.htm

In modo più prosaico molte aziende partecipano per avere contatti ed estendere il proprio network. Vi sono ad esempio osservatori, semplici membri senza potere di voto e giornalisti. OMG è un'associazione che comprende il valore della comunicazione e del commercio ed infatti collabora con aziende per seminari, forum e corsi. OMG ha, ad esempio, un programma di certificazione su UML, e presto ne avrà uno anche per MDA.

In OMG si sta chiudendo la specifica UML 2.0 che pur se supportata da diversi tool non è ancora ufficialmente rilasciata. UML 2.0 ha finalmente affermato la separazione forte tra modelli behavioural e structural che prima erano solo una classificazione informale. Ha inoltre introdotto nuovi diagrammi con le relative semantiche: component, composite structure, communication, interaction e timing (c'è una vasta letteratura che spiega UML 2.0). Una aggiunta importante anche se ancora in erba sono le actions semantics che intendono colmare la natura ancora dichiarativa di UML e supportare la definizione del behaviour dei metodi: questo dovrebbe avvicinare il gruppo al lavoro sulla standardizzazione di xUML.

Le principali evoluzioni sono tuttavia più interne che esterne, aspetti che non sono visibili agli utenti finali di UML, ma ai vendor che lo implementano, vale la pena a questo punto di aprire una parentesi. Nelle prime versioni di UML la tendenza era di estenderne la semantica aggiungendo primitive che supportassero le esigenze di chi modellava con i sistemi realtime, i database, i componenti, il business, CORBA ed altri ancora. I tres amigos (Runbaugh, Booch e Jacobson) infatti hanno fatto confluire i loro linguaggi in uno solo, fondendo i loro diversi punti di vista nel software engineering. Prima ancora di farlo diventare un mostro di linguaggio omni-comprendivo, OMG ha con molta lungimiranza, preso un'altra strada e definito un linguaggio con cui definire UML: MOF (Meta Object Facility), nome non felice debbo ammettere. L'idea alla base è che UML non sia "il linguaggio" di modellazione per eccellenza, ma uno dei tanti che si possono definire usando MOF, un linguaggio generico e molto più piccolo. Ecco che a fianco di UML sono emersi altri linguaggi specializzati scritti in MOF per definire la notazione per modelli di business (BPMN), per la modellazione del business (BPDM), per i database (CWM) ed altri ancora (questo non è l'unico modo per definire linguaggi di modellazione, vi sono altre tecniche meno brutali come i profili, che non abbiamo spazio qui per approfondire).

Chiarisco: consideriamo ad esempio un utente di un tool UML che crea un Class, un elemento UML, per rappresentare l'entità Cliente che è il modello; un'istanza di Cliente potrà essere ad esempio "Mario Rossi snc". Il metamodello UML descrive -semplificando- che Class è un'istanza di Classifier (un elemento di MOF). Allo stesso modo un Attribute di UML è un Modelling Element di MOF che è in relazione di composizione con Class, istanza dell'elemento Classifier di MOF. Sperando di non confondere, possiamo a ritroso dire che i casi reali sono i dati, descritti da un modello scritto in UML, descritto a sua volta da un metamodello scritto in MOF. E MOF come è scritto? MOF è un meta-metamodello perchè è un linguaggio per scrivere metamodelli, che si auto-describe. Per cui non avremo mai un meta-meta-metamodello.

UML 2.0 è stato soprattutto una riscrittura in MOF (UML è quindi un modello MOF). Pur essendo all'apparenza lo stesso per le entità "vecchie", in realtà è stato riformulato, anche in modo pesante, in altre parole il metamodello UML è meglio strutturato e più facile da riusare.

Si nota, partecipando ai lavori che, OMG per la parte di Platform Domain si occupa solamente di metamodellazione. Chi partecipasse alle riunioni o che seguisse i tutorial di UML resterebbe deluso poiché non trattano di modelli specifici e neppure propongono metodi su come debba essere usato UML, il focus è molto bastato sulla astrazione. Ad esempio un effort chiamato SPEM (Software Process Engineering Metamodel) definisce un linguaggio per modellare i processi di sviluppo software, sulla base del quale i produttori di tool costruiranno i loro strumenti indipendentemente da OMG. In OMG si definiscono linguaggi, cioè meta modelli, con lo scopo di avere una semantica comune nel descrivere poi i casi reali ed i problemi. OMG non si occupa neppure di metodologie, infatti un altro malinteso è che UML sia un metodologia, si tratta in modo più prosaico solo di un linguaggio (pure a sua volta estensibile). Ma una metodologia descrive come usare un linguaggio e seppure la cosa desti delusione, si tratta di un punto a favore di OMG poiché permette di lasciare libertà a chi deve affrontare problemi

diversi, ognuno ha il suo stile. Vi sono aziende che hanno le loro metodologie, noi in Soluta.net abbiamo Chorus(tm) ed esiste Iconix come altro esempio, in ogni caso in nessuna associazione si sarebbe mai trovato un accordo su una metodologia unica.

OMG è stata molto lungimirante nella definizione di MOF poiché questo permette ora di avere una quasi completa interoperabilità tra i tool UML, aspetto che negli anni 90 aveva creato grosse difficoltà. Oggi i principali tool UML “capiscono” MOF, cioè sono in grado di leggere modelli scritti in MOF ed UML è scritto in MOF come dicevamo: la semantica è comune finalmente. Prima di tutto questo era come se ogni tool parlasse una lingua diversa (italiano, tedesco), uno stesso modello quindi poteva essere scritto in ogni lingua. Un tool per poter importare da tutti avrebbe dovuto capire tutti i linguaggi, oggi questo non è un problema, basta convenire su MOF. Resta da ricordare che lo sforzo non è stato la standardizzazione di UML come linguaggio (cosa relativamente più facile), ma la definizione del meta linguaggio MOF. Ecco che oggi due ipotetici tool sono in grado di interscambiarsi modelli UML anche con la versione v2.1 di UML, fatto fermo che MOF non sia cambiato ovviamente.

C'è da evidenziare che MOF ha portato un altro più significativo miglioramento nella vita degli informatici che passa inosservata e trasparente, ed è un bene. Avendo ora questo linguaggio dei linguaggi si possono definire delle regole di mappatura tra MOF ed altri linguaggi come Java, Corba, C++, XML (ed altri a piacere). I lettori più attenti avranno già capito che questo permette la realizzazione di una specie di magia; infatti una volta che sia stato descritto come si può mappare MOF in XML (ad esempio), si può applicare questa regola ad ogni modello scritto in UML che si trova quasi magicamente -è il caso di dirlo- trasformato in XML. Ma questo non vale solo per UML ma per ogni linguaggio scritto in MOF come CWM o BPDM. L'architettura di OMG comincia a pagare i dividendi. Il mapping MOF-XML si chiama XMI ed il mapping MOF-Java JMI che però non è stato definito in OMG ma bensì all'interno dalla Java Community Process.

Conto di aver chiarito alcuni aspetti di OMG, della sua organizzazione e delle strategie che la muovono. In altri interventi si potrà parlare degli altri meccanismi di estensione, dei linguaggi di mapping e di MDA.

--- Fine documento ---