

# Metodologie Object-Oriented in campo legislativo

Le leggi dello Stato italiano soffrono di scarsa efficacia e di difficile leggibilità. Per questo motivo sono aperte a interpretazioni che il più delle volte causano ulteriori problemi. In questo articolo proporremo una panoramica sugli strumenti di analisi informatica e sulla loro applicazione alle leggi dello Stato

di **Andrea Boschin** e **Davide Mauri**

Diversamente da quello che si potrebbe pensare, la parte più impegnativa nella realizzazione di un software non è quella della scrittura del codice eseguibile, che darà vita e corpo alle sue funzionalità, ma piuttosto la definizione stessa di tali funzionalità, e quali sono le modalità con cui esso opererà sui dati che dovrà trattare. L'analista informatico, ed il programmatore stesso, si trovano spesso ad affrontare argomenti che per ovvie ragioni non conoscono. Si pensi ad esempio alla realizzazione di un software gestionale, nel quale il problema da risolvere è sicuramente conosciuto meglio dal committente che dall'esecutore. Per poter enunciare delle "regole" che determinano il funzionamento di una logica di business, è necessario che siano a priori conosciute le interrelazioni che esistono tra le entità trattate da tale logica e gli interventi esterni che apporteranno modifiche a que-

ste entità e ai loro dati. Negli ultimi anni, nel campo dell'ingegneria del software si è venuta affermando una metodologia analitica che ha essenzialmente questo scopo, e che poggia le sue basi su un linguaggio unificato necessario per esprimere una rappresentazione il più possibile efficace della realtà. Questo set di strumenti, noto con la denominazione di UML [1], ha lo scopo primario di descrivere una realtà che dovrà essere regolata da una logica di business da un lato, e dall'altro di descrivere le regole che ne determineranno il funzionamento. In questo modo è possibile verificare che la logica che l'applicazione deve incorporare sia corretta, chiara e senza possibilità di fraintendimento anche in mancanza di un background informatico.

## Informatica e attività legislativa

È proprio durante l'uso di questi strumenti che abbiamo avvertito delle forti analogie con l'attività che deve essere in grado di svolgere il legislatore, quando affronta l'estensione di una legge. È del tutto palese che la realizzazione di un solido ed efficace dispositivo legislativo passa immancabilmente per diverse fasi, riassunte nel **Riquadro 1** che, per semplicità, possono essere ricondotte alla seguente categorizzazione logica: la conoscenza della realtà che si deve normare e la descrizione efficace delle norme che si desidera siano applicate. La prima fase permette di determinare quali sono le tipologie di persone, cittadini ed enti che intervengono nella realtà attuale, mentre nella seconda si cerca di comprendere la realtà come dovrà essere dopo che la legge entrerà in vigore e se ne verifica la correttezza e l'applicabilità. L'idea nasce da una semplice constatazione: una legge può essere considerata come un algoritmo che spieghi cosa deve – o non deve – essere fatto. Partendo

### Andrea Boschin

aboschin@infomedia.it

È stato consulente presso la UNISYS Corporation, dove si è occupato di applicazioni per il web-publishing e ha lavorato presso il gruppo Inferentia DNM. Attualmente lavora come senior IT engineer presso Archeometra S.r.l. a Castelfranco Veneto (TV). È membro attivo dello user group italiano di .NET [www.ugidotnet.org](http://www.ugidotnet.org).

### Davide Mauri

dmauri@infomedia.it

È un consulente freelance che si occupa di SQL Server 2000 e 2005 e di sviluppo di soluzioni web basate sul .NET framework. All'attività di consulenza affianca una cospicua attività di formazione presso Mondatori Informatica. È membro attivo dello user group italiano di .NET [www.ugidotnet.org](http://www.ugidotnet.org).

Contenuti tipici dell'attività legislativa

- valutazione impatto reale
  - I. preventiva: analisi fattibilità
  - II. consuntiva: analisi efficacia
- valutazione impatto normativo
  - I. valutazione coerenza interna
  - II. valutazione coerenza esterna
- valutazione impatto giuridico (interpretazione normativa)
- elaborazione tecnico-legislativa (drafting)

Fasi del processo legislativo

- decisione politica (a, b, c)
- drafting (composto da: d)
- controllo progetto (b, a, c)
- verifica legge (a, b)

## RIQUADRO 1 Contenuti e fasi dell'attività legislativa

da questo assunto si può quindi supporre che gli strumenti analitici utilizzati per sviluppare e testare algoritmi informatici possano essere utilizzati anche in campo legislativo.

Nel corso di questo articolo cercheremo di approfondire quanto finora enunciato, con lo scopo di proporre per l'iter del processo legislativo, una possibile metodologia formalizzata. È evidente che quanto sarà esposto nel corso dei prossimi paragrafi vuole introdurre strumenti che hanno l'unico scopo di semplificare, chiarificare, comunicare e rendere più efficiente lo sviluppo di soluzioni a problematiche informatiche o – e questa è la nostra speranza – politiche e civili. Inizieremo quindi con l'affrontare un sottoinsieme degli strumenti di analisi, andando a descriverne il significato e l'utilizzo, ed in seguito tenteremo una sorta di reverse engineering di una parte di un testo unico dello Stato italiano, allo scopo di dimostrare che l'uso di diagrammi UML è plausibile, e può certamente aiutare a comprendere quali sono le criticità di un sistema/legge nel trovare le soluzioni più adeguate.

## Strumenti di analisi UML

Se è vero che “fatta la legge, trovato l'inganno”, è immediatamente chiaro che il compito di un'analisi legislativa ben fatta è proprio quello di definire quali sono questi possibili “inganni” per evitare che possano essere perpetrati. I testi di legge attuali sono così arzigogolati e ripetitivi proprio allo scopo di coprire tutti i possibili casi legislativi, anche quelli che risultano imprevedibili. Proprio da questo punto nasce l'idea di usare la stessa tecnica adottata dagli ingegneri del software moderni, per scoprire tutti i possibili casi d'uso di un software. Questa tecnica, denominata appunto *use-case diagram*, a prima vista può risultare ovvia e banale, ma all'atto pratico è uno degli strumenti più efficaci nello stimolare la valutazione della materia in analisi durante la sua definizione.

In effetti l'estrema utilità di un diagramma use-case risiede proprio nella sua ovvietà e banalità: grazie ad esso diventa pressochè impossibile fraintendere il

significato della situazione che lo use-case descrive. Questo – a pensarci bene – è già un grande passo avanti: normalmente molti problemi nascono semplicemente dal fatto che le parti in gioco hanno interpretato in modo leggermente diverso il significato di una frase. Dopotutto nessuno si mette mai a discutere del fatto che il cielo sia blu o che il prato sia verde: essendo osservazioni ovvie e banali non c'è spazio per dubbi né fraintendimenti.

Uno use-case permette di ottenere proprio questo importante obiettivo: fissare in modo chiaro e semplice ciò che abbiamo o che dobbiamo ottenere. Un fatto molto interessante da notare, e che ci indica quanto sia necessario evitare qualsiasi tipo di ambiguità nel testo di una legge, è dato proprio dall'articolo 12 delle Disposizioni Preliminari al Codice Civile (*preleggi*):

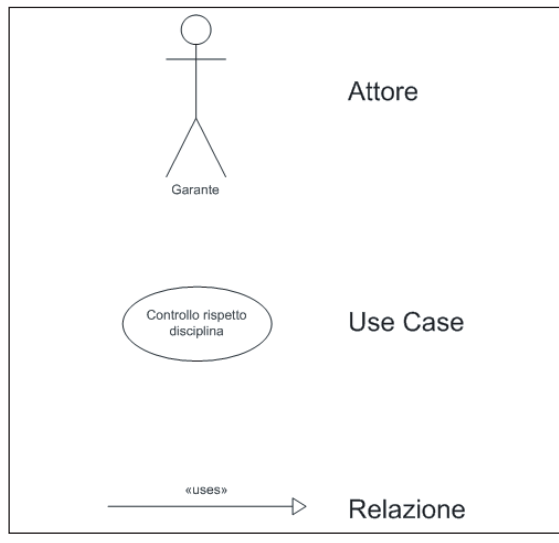
*“Nell'applicare la legge non si può ad essa attribuire altro senso che quello fatto palese dal significato proprio delle parole secondo la connessione di esse, e dalla intenzione del legislatore.”*

Un meccanismo formale permetterebbe quindi di soddisfare in pieno tale direttiva, evitando immediatamente le ambiguità obbligatoriamente introdotte dall'uso di una lingua parlata.

Ma com'è fatto, in soldoni, uno use-case? Prima di vederne un esempio è necessario comprendere quali sono gli elementi propri di questo strumento.

- 1) **Scenario.** La situazione che lo use-case descrive prende il nome di scenario: questo descrive le interazioni tra un attore ed il sistema. Tipicamente lo scenario viene definito dal committente stesso che durante le interviste dell'analista descrive – con parole sue – quello che avviene nella sua situazione e quello che si aspetta dovrebbe essere il comportamento del sistema. Proiettando questo concetto in ambito legislativo lo scenario altro non è che il contesto pratico nel quale la materia legislativa normalmente si snoda.
- 2) **Attore.** Un attore è una qualsiasi entità (un essere umano oppure una macchina) che agisce all'interno dello scenario. Il suo intervento è fondamentale per lo svolgersi degli eventi ed egli apporta modifiche o beneficia dei risultati di una situazione. Talvolta un attore può essere il sistema informatico stesso, ad esempio qualora esso si rapporti con altri sistemi esterni. È evidente che in un testo legislativo l'attore è anch'esso un elemento fondamentale dell'iterazione: attore potrà essere quindi un cittadino, piuttosto che un ente pubblico o privato oppure un organo di esercizio del potere giudiziario.
- 3) **Use-case.** Lo use-case rappresenta l'obiettivo che si deve raggiungere. È composto da una serie di passi che specificano come è possibile raggiungere tale obiettivo. Esso è un testo molto discorsivo ma al tempo stesso molto preciso e formale che si articola nella spiegazione di come avviene l'interazione tra

**FIGURA 1** Gli elementi grafici utilizzati per creare Use Case Diagrams



uno o più attori all'interno di uno scenario, per compiere un dato compito. È interessante rilevare che spesso, durante l'approfondimento dell'analisi, è possibile che un passo di uno use-case diventi esso stesso uno use-case: questo permette di semplificare e chiarificare ulteriormente passaggi complessi.

Prima di passare all'ultimo elemento, applichiamo quanto appena detto prendendo come riferimento il punto n.1 e 1.1 dell'articolo 154 della legge sulla Privacy:

*Oltre a quanto previsto da specifiche disposizioni, il Garante, anche avvalendosi dell'Ufficio e in conformità al presente codice, ha il compito di:*

*a) controllare se i trattamenti sono effettuati nel rispetto della disciplina applicabile e in conformità alla notificazione, anche in caso di loro cessazione;*

Da quanto descritto si può notare che l'attore è il Garante, e lo use-case è il "Controllare il rispetto della disciplina". In questo caso omettiamo gli step (che descriverebbero i passi necessari per portare a compimento il suddetto use-case) in quanto sarebbero i punti riportati in altri articoli della legge stessa, ognuno dei quali potrebbe avere uno suo use-case.

3) **Use-case diagram.** Il Garante citato nell'esempio sarà presente in altri use-case. Questo può portare le cose a complicarsi in modo notevole, in quanto non sarebbe più così facile notare a vista d'occhio, senza leggere tutti gli use-case, in quali entra in gioco il suddetto attore ed in che modo. Per questo può essere molto utile raggruppare gli use-case con una notazione che non si basa sulla lingua parlata, ma sull'uso di semplicissimi diagrammi e di indicazioni che ne descrivono la relazione. Gli elementi del diagramma sono visibili in **Figura 1**; un diagramma di esempio

che invece dimostra come possono essere utilizzati per schematizzare il concetto espresso della legge citata poche righe fa è visibile in **Figura 2**.

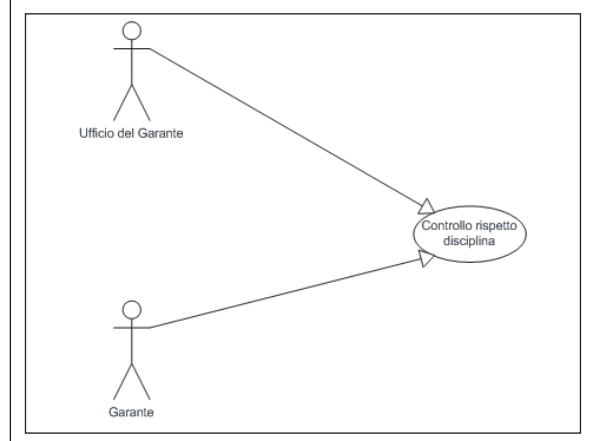
Se state leggendo queste righe dopo aver visto il diagramma di esempio probabilmente state pensando "è uno scherzo?". Come possono dei disegni *naïf* aiutare nella progettazione?

La risposta la si ottiene ricordandosi che un diagramma è utile per mostrare come sono relazionati tra loro molteplici use-case: in questo modo, come visibile in **Figura 3** si può facilmente intuire quali sono i punti in comune tra vari use-case e quindi, nel caso di un testo legislativo, sarà più facilmente determinabile come sono articolati i compiti e le responsabilità, così da poter aver più chiaro il tipo di impatto che la legge avrà sugli attori.

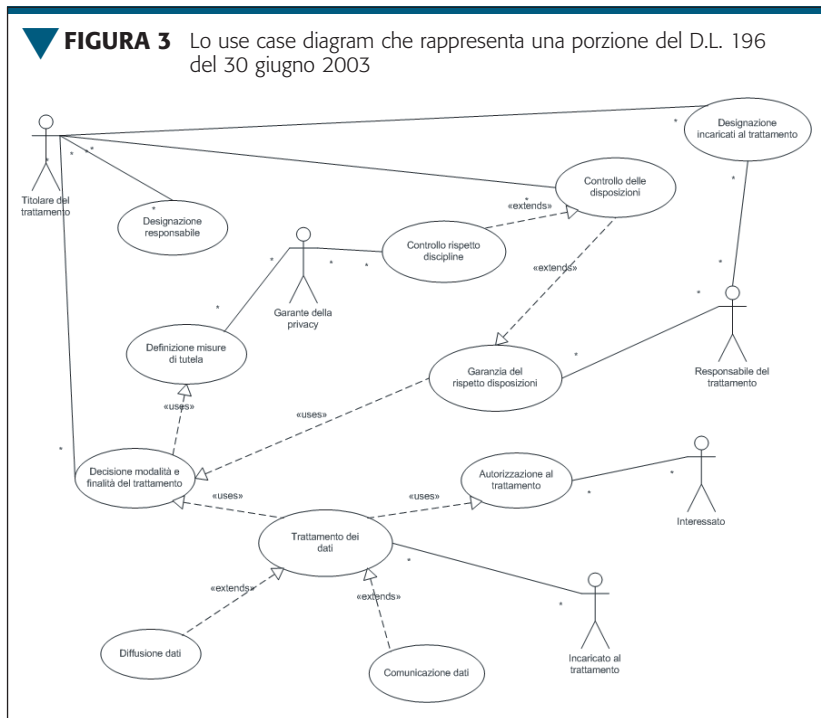
**Analisi legislativa**

Dallo schema proposto in **Figura 3** realizzato elaborando il testo del Titolo I e IV del Decreto legislativo 30 giugno 2003, n. 196 - *Codice in materia di protezione dei dati personali*, si evince innanzitutto che la chiarezza espositiva dell'articolato di una legge è decisamente meno efficace di quella basata su diagrammi use-case. Nello schema si può facilmente notare che vi sono notevoli ridondanze nelle responsabilità da parte dei vari attori e che tali responsabilità non sono ben chiare e potranno dare adito a contenziosi tra gli attori stessi. Tuttavia è importante notare che use-case e testo di legge debbono essere considerati complementari, perché mentre il diagramma dà un'idea di massima della materia regolamentata nel suo assieme, il testo legislativo enuncia una serie di dettagli che non sono evidenziati dallo schema. Quest'ultimo punto mostra ulteriormente come ci sia una notevole vicinanza tra analisi legislativa ed informatica: un diagramma use-case deve essere sempre supportato dal relativo use-case scritto a punti (un esempio è visibile nel **Riquadro 2**) in quanto il primo semplifica la visione globale ed il secondo assicura che ogni elemento sia spiegato per esteso e nei dovuti particolari. Considerando questa collaborazione use-

**FIGURA 2** Un semplice use case diagram di esempio



**FIGURA 3** Lo use case diagram che rappresenta una porzione del D.L. 196 del 30 giugno 2003



case/testo legislativo – indispensabile – si può certamente tentare di adottare un metodo alternativo nell'estensione del testo di legge allo scopo di renderlo più adeguato a quanto viene schematizzato dallo use-case.

Ad esempio, sarebbe interessante realizzare un testo legislativo come codifica esaustiva degli use-case che sono nati da una analisi legislativa. L'attività di legiferazione quindi si dovrebbe concentrare innanzitutto nella definizione dei casi d'uso della normativa, cercando di evidenziare analogie nelle responsabilità, uniformità nelle azioni e gerarchie di attori. In un secondo momento il legislatore si potrebbe concentrare nel dettagliare analiticamente il corpo dello use-case elencando i passi necessari al compimento e con essi tutte le possibili

eccezioni, in modo abbastanza analogo a quello che avviene attualmente in campo informatico. Vantaggio indiscutibile in questo processo, è l'isolamento di problematiche semplici, la cui soluzione è notevolmente agevolata rispetto alla totalità del problema. In questo modo, la schematizzazione degli use-case potrebbe aiutare a trovare i punti ridondanti nella normativa e accentrarli in semplificazioni e quindi in nuovi use-case che coprono settori più specifici. Tale possibilità aiuterebbe anche in modo notevole a vedere quali attori verranno toccati da eventuali modifiche apportate alle leggi, ed eventualmente evitare di dovere successivamente ed in tutta fretta ritoccare le modifiche apportate (citiamo come esempio il Decreto Urbani sul P2P). Un altro vantaggio che si può ottenere grazie a questo approccio, è dato dalla facilità e dal-

l'efficacia con cui esso può essere comunicato, senza che intervengano interpretazioni o dimenticanze. È risaputo che una delle difficoltà imposte dalla legiferazione classica è proprio quella dell'interpretazione delle norme di legge. L'adozione di un linguaggio UML per rappresentare i casi d'uso della legge porterebbe ad una maggiore chiarezza degli intenti della legge.

### Conclusioni

Se quanto riportato finora è convincente ed efficace, ci chiediamo perché non si debba tentare di andare oltre in questo processo. Effettivamente le analogie tra una attività di analisi e sviluppo di un software ci sembrano molte e tutte calzanti. È evidente che ognuno dei paradigmi che introduce l'UML per tentare di descrivere i vari aspetti di un software è a questo punto applicabile in qualche modo al processo legislativo. Sequence diagram potranno quindi descrivere procedure legali e/o giudiziarie, state diagram potranno evidenziare gli stati di un contribuente rispetto al fisco. Ma continuando su questa linea di pensiero si può facilmente tentare di applicare le metodologie di sviluppo agile (agile development/xp) all'iter legislativo. Gli emendamenti ad una legge potranno quindi diventare attività di refactoring e si potrebbe cercare di adottare procedure di unit testing dei singoli commi o titoli di un dispositivo di legge. Fantascienza? 🧩

#### Scenario:

- 1) L'utente cerca nel catalogo il prodotto da acquistare
- 2) L'utente inserisce nel carrello i prodotti selezionati
- 3) L'utente va alla pagina di check-out
- 4) L'utente inserisce dati anagrafici e tipo di spedizione
- 5) Il sistema mostra il dettaglio totale dei costi
- 6) L'utente inserisce il numero di carta di credito
- 7) Il sistema autorizza il pagamento
- 8) Il sistema effettua la transazione
- 9) Il sistema aggiorna il magazzino
- 10) Il sistema invia un e-mail all'utente con i dati

#### Casi particolari:

- Punto 4: L'utente è già registrato
- a) Il sistema mostra i dati già inseriti
  - b) L'utente li accetta oppure specifica un altro indirizzo

#### Punto 7: L'autorizzazione fallisce

- a) L'utente può specificare un'altra carta di credito oppure annullare l'operazione

**RIQUADRO 2** Use case per l'acquisto di un prodotto tramite un sito eCommerce

### RIFERIMENTI

- [1] M. Fowler, K. Scott, "UML Distilled 3rd Edition", Addison Wesley
- [2] P. Mercatali, "Legimatica e redazione delle leggi", <http://www.ittig.cnr.it/Ricerca/Testi/mercatali1995.htm>