Normativa ed informatica: un matrimonio difficile

Dott. Arch. Roberto Spagnuolo
Softing srl

Convegno AIST

Bologna, 6 ottobre 2011

E' mia convinzione, maturata in quasi un trentennio di attività, che vi sia un "equivoco" di fondo tra ingegneria ed informatica che impedisce una "sana" e proficua informatizzazione nel campo delle costruzioni ed una sana informatizzazione delle norme tecniche per le costruzioni.

Per dare una ragione con basi quanto più possibile scientifiche di questa, che non vuol essere solo una personale opinione, ritengo che la teoria della complessità possa interpretare bene i problemi che vi sono.

Li interpreta efficacemente sotto i due profili più rilevanti del fenomeno: quello sociale e quello tecnico.

Per capire quanto sostengo, occorre comprendere quanto asserisce la teoria della complessità e cioè che:

- Il comportamento di un sistema complesso **non è facilmente predicibile** anche se conosciamo il comportamento di ciascuno dei suoi componenti.

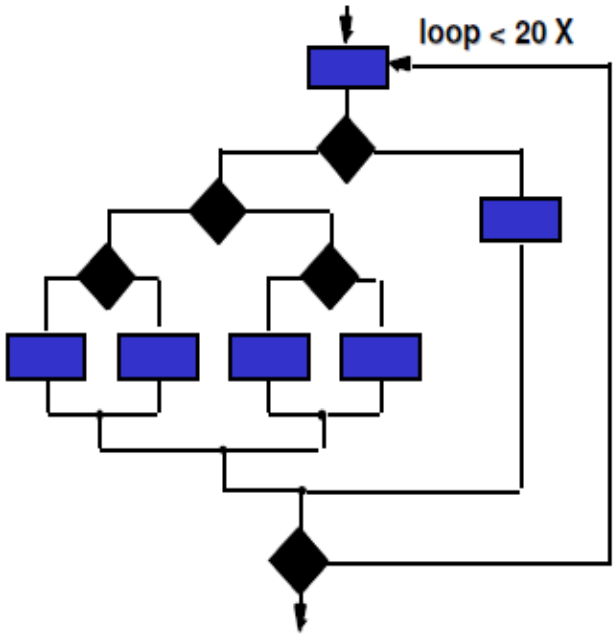
- I sistemi complessi non sono soltanto più complicati dei sistemi semplici, s**ono qualitativamente diversi**.

Uso una metafora di questo fenomeno che trovo divertente: se prendiamo un gatto e lo facciamo crescere non otteniamo un gatto grande ma una tigre e se trattassimo affettuosamente il gatto "grande" così ottenuto rischieremmo di esserne sbranati.

Sotto il profilo culturale ciò comporta che tutti coloro i quali, e nel mondo dei tecnici è facile che accada, hanno avuto occasione di cimentarsi con un semplice problema di programmazione, ritengano che con un procedimento semplicemente additivo si possa passare dal semplice al complesso per cui hanno della informatica una visione piuttosto semplicistica: gli informatici sono più muratori che non architetti, devono cioè ripetere più volte una operazione semplice. Ma così non è.

Sotto il profilo più tecnico, ed è questo il caso della normativa, si trascura il fatto che una formulazione complessa, invece di una semplice, conduce inevitabilmente a comportamenti del software meno prevedibili.

Per fare un esempio, riporto uno schema a blocchi da un testo dell' Ing. Antonio Cornato dell'Università Federico II di Napoli, ma di esempi simili ve ne siano centinaia sui vari testi.



- Semplice programma C di 100 linee di codice
- Due cicli annidati con 4 costrutti if

- 10^{14} possibili path!
- Se se ne eseguissero uno ogni millisecondi ci vorrebbero 3170 anni

Questo esempio evidenzia perfettamente che il comportamento di un piccolo pezzo di codice complesso è sostanzialmente inconoscibile. E si badi, non perché vi siano degli errori in esso, ma perché è logicamente molto complesso.

Una delle metriche della complessità del software è la complessità ciclomatica dovuta a McCabe che si basa, come nel precedente esempio, sui i possibili percorsi che può fare un programma tra dati e risultati. McCabe e il NIST (National Institute of Standard and Technology) ritengono che la complessità ciclomatica 10 non dovrebbe essere sorpassata per evitare appunto comportamenti del software non perfettamente predicibili.

Se però consideriamo la normativa ed esaminiamo la procedura per la verifica di ammissibilità di una analisi con spettro di risposta elastico di strutture esistenti in calcestruzzo, rileviamo che la complessità ciclomatica del procedimento è 11 e cioè supera la soglia consigliata da McCabe.

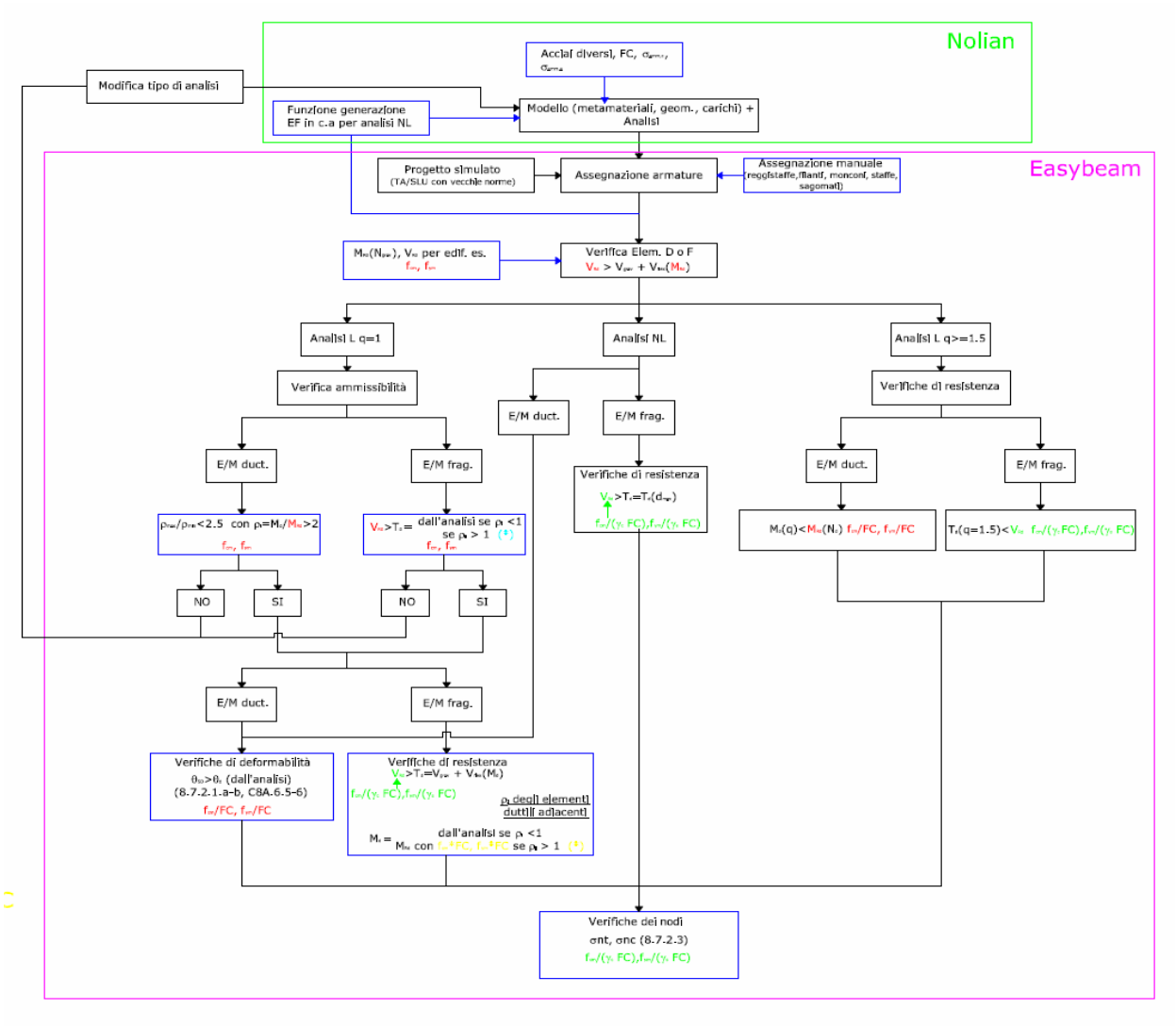


Diagramma logico della procedura di analisi e verifica di edifici esistenti in calcestruzzo redatto dall' Ing. Cristiano Bilello, per sua gentile concessione.

Con questo non si vuol dire che tale procedura sia scorretta o pericolosa ma si evidenzia come il normatore non si sia posto il problema del rapporto tra costi e benefici (in termini di qualità della soluzione) tra un sistema numericamente più compatto e logicamente meno complesso e quello "classificatorio" adottato. Si rischia cioè per spaccare la pagliuzza, di non tagliare neanche il famoso ramo di evangelica memoria.

In conclusione, una informatizzazione produttiva e "sana" presuppone che la procedura da informatizzare (che poi è la specifica del software) sia pensata in termini tali da poter essere al meglio informatizzata, e non ritenendo un fatto solo da opera di portatori d'acqua quello di informatizzare poi, a posteriori, una qualsiasi procedura.

Come si può vedere, nel mio intervento non ci sono formule e ciò può deludere i più, ma sono convinto che spesso l'esposizione cavillosa di un modello matematico in contesti non specialistici serva solo a nascondersi in modo sacerdotale per non voler affrontare i problemi veri. Cito a questo proposito Bertrand Russel che, nella sua Storia della filosofia occidentale, sottolinea come nei momenti più tragici dell'Impero (così simili alla attuale decadenza della nostra civiltà), mentre gli Unni saccheggiavano Aquileia e i legionari Goti deponevano un imperatore dopo l'altro, le maggiori menti dell'epoca impiegavano il loro tempo a scrivere accalorate lettere sulle gioie della verginità e ad accapigliarsi su problemi essenziali come la transustanziazione.