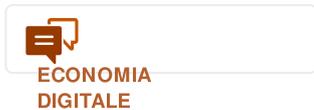


 Contenuto archiviato il 2024-04-23

## Articoli di approfondimento - Obiettivo semplificazione: controllare la complessità dei software

"Gli stupidi ignorano la complessità, mentre i pragmatici la temono. Alcuni la evitano, i geni la eliminano". Queste sono le parole di Alan Perlis, scienziato pioniere dell'informatica che storcerebbe il naso, sgomento, di fronte alla complessità dei moderni sistemi di software. Benché non possano descriversi come geni, i membri di un team di ricercatori finanziati dall'UE si stanno occupando del controllo della complessità, utilizzando un approccio alternativo all'ingegneria del software per la creazione di sistemi più sicuri, economici e resistenti.



I software non sono diventati solo più complessi, ma anche più pervasivi: la loro presenza si fa sentire in vari luoghi, tra cui automobili, contatori dell'elettricità e centrali elettriche, aerei, treni, banche e ospedali. I software fanno funzionare sistemi critici in cui un solo errore può avere conseguenze devastanti. Inoltre, l'aumento del rischio di malfunzionamenti dovuto alla maggiore complessità dei software non rappresenta

l'unico problema, in quanto quest'ultimo aspetto determina anche un aumento dei costi: il collaudo dei software finalizzato alla ricerca di bug rappresenta attualmente circa la metà dei costi di pre-rilascio e il 70 % dei costi di post-rilascio.

Il cuore del problema è rappresentato dal fatto che i processi tradizionali di ingegneria del software non sono progettati in modo adeguato alla gestione dell'enorme complessità e diversità dei moderni sistemi di software.

"Se osserviamo ad esempio un'auto moderna, ci accorgiamo della presenza di centinaia di elementi software coesistenti, dal controllo della qualità del suono al controllo della velocità di crociera. Con un numero sempre crescente di elementi e di funzionalità integrati nei sistemi, gli ingegneri sostengono di aver perso il controllo della complessità e temono di non essere in grado di fornire le garanzie necessarie in termini di qualità", spiega Alexander Romanovsky, docente di informatica presso Università di Newcastle nel Regno Unito. "La complessità si trova ovunque ed è insita in qualsiasi cosa: le aziende che costruiscono auto, treni, aerei, ecc. investono effettivamente buona parte del tempo e delle risorse a disposizione, probabilmente la metà, nello sviluppo di software: in definitiva, tutti i settori producono software".

La domanda è: "L'industria è in grado di trovare metodi migliori per la creazione di software, specialmente nel caso di sistemi critici?".

I ricercatori coinvolti nel progetto ([Deploy](#) ("Industrial deployment of advanced system engineering methods for high productivity and dependability"), che ha ricevuto un finanziamento di 12,4 milioni di euro dalla Commissione europea ed è coordinato dal prof. Romanovsky, ritengono che ciò sia possibile. L'approccio degli esperti si basa su metodi di ingegneria formale che, grazie specialmente al contributo del team, stanno iniziando a essere accettati da aziende e ingegneri, come strumenti più efficaci e pratici per lo sviluppo di complessi sistemi di software. I processi Deploy sono già stati utilizzati per lo sviluppo di software critici per la sicurezza delle linee metropolitane e delle navette aeroportuali.

Diversamente dagli approcci tradizionali dell'ingegneria del software, i metodi formali, incentrati su modellizzazione e analisi matematica, sostengono il ragionamento su vari livelli di astrazione ai fini dell'attivazione di un flusso ingegneristico sistematico che va dalla specifica dei requisiti, mediante la modellizzazione dell'architettura e la progettazione dei dettagli, all'implementazione, al collaudo e all'utilizzo. Per il progetto, il team ha proseguito l'evoluzione di "Event-B", un metodo formale per la modellizzazione e l'analisi a livello di sistema, sviluppato per la prima volta nell'ambito dell'iniziativa Rodin ("Rigorous open development environment for complex systems"), finanziata dall'UE e sostenuta da "Rodin Tools", un ambiente di sviluppo integrato.

Incoraggiare gli ingegneri a pensare diversamente

"Per gli ingegneri, si tratta di un cambiamento piuttosto radicale. Tipicamente, quando si lanciano in un progetto, gli esperti sono abituati a iniziare con lo sviluppo del software, mediante aggiunte ed espansioni, e a lavorare duramente solo alla fine, dedicandosi al processo di implementazione, di verifica e di collaudo. L'ingegneria formale rovescia questa tendenza: il duro lavoro, che parte da una sintesi dei requisiti di sistema e procede in base ad un rigoroso processo di sviluppo logico-matematico, viene effettuato all'inizio", spiega il prof. Romanovsky.

Il vantaggio consiste nel fatto che gli errori sono individuati nelle prime fasi dello sviluppo e il rigoroso approccio logico fa sì che la complessità sia contenuta entro limiti prefissati che riducono il bisogno di collaudo nelle fasi finali. In definitiva, tale approccio conduce a sistemi più resistenti e affidabili, potenzialmente a costi nettamente inferiori rispetto a quelli implicati nell'utilizzo di metodi ingegneristici tradizionali.

Sebbene possa sembrare una situazione vantaggiosa per tutti, rimane ancora in ballo una sfida enorme: il fatto, cioè, che solitamente gli ingegneri non sono dei matematici.

"I metodi su cui si basa l'ingegneria formale richiedono conoscenze matematiche non possedute dagli ingegneri, abituati invece a utilizzare approcci tradizionali e, poiché si tratta di un processo così diverso, è necessario un cambiamento di mentalità. Inoltre, è difficile convincere un'azienda che sviluppa software da 10-30 anni in un certo modo ad adottare metodi completamente diversi", spiega il prof. Romanovsky.

Il progetto Deploy, che rappresenta probabilmente il più ampio utilizzo di ingegneria formale del software a livello industriale mai realizzato, ha fatto molti progressi nel superamento di questi ostacoli.

Ad esempio, grazie alla collaborazione con il partner del progetto SAP, il team ha implementato in modo trasparente processi formali che consentono agli ingegneri di continuare a utilizzare linguaggi di programmazione specifici dei domini, allo scopo di sviluppare software critici per le attività commerciali, tradotti quindi automaticamente nel metodo Event-B, sostenuto da esperti di ingegneria formale.

Siemens, i cui ingegneri possiedono già una notevole esperienza nella modellizzazione formale, ha adottato il metodo Event-B per sviluppare software attualmente utilizzati nei sistemi metropolitani di Barcellona e di Parigi e altrove per il controllo e la segnalazione dei treni. Bosch ha utilizzato questo approccio per sviluppare sistemi di controllo della velocità di crociera e di start e stop per le auto. Inoltre, Space Systems Finland, un altro partner del progetto, ha utilizzato il processo per i componenti della sonda spaziale BepiColombo dell'Agenzia spaziale europea e per i sistemi di controllo dell'altitudine e delle orbite.

Al contempo, il partner francese Systeme S sta utilizzando il metodo Event-B per una gamma di sistemi ferroviari e aerospaziali. I ricercatori del progetto Deploy hanno creato due imprese spin-off, Rodin Tools e Formal Mind, per la commercializzazione e la promozione dell'utilizzo degli strumenti creati. Grazie ai finanziamenti dell'UE gli esperti si sono altresì lanciati in un progetto di follow-up chiamato Advance ("Advanced design and verification environment for cyber-physical system engineering").

L'utilizzo efficace e ininterrotto di metodi di ingegneria formale nelle aziende coinvolte offre un chiaro esempio delle strategie che le altre imprese dovrebbero seguire per ricercare nuovi metodi di gestione della complessità.

"Stiamo certamente offrendo all'industria spunti di riflessione sugli attuali metodi ingegneristici e sui vantaggi che il passaggio ad un approccio formale comporta in alcune circostanze", osserva il prof. Romanovsky.

Gli strumenti, l'ampia documentazione su Event-B e il sostegno offerto dai partner Deploy potrebbero altresì contribuire a convincere gli ingegneri a lanciarsi in progetti di ingegneria formale.

"Nel lungo termine i sistemi di software non faranno altro che diventare sempre più complessi. L'ingegneria formale, come abbiamo dimostrato, rappresenta uno strumento per affrontare questo problema", afferma il prof. Romanovsky.

Il progetto Deploy ha ricevuto finanziamenti per la ricerca nell'ambito del Settimo programma quadro (7° PQ) dell'UE.

Collegamento al progetto su CORDIS:

- [7° PQ su CORDIS](#) 
- [Scheda informativa del progetto Deploy su CORDIS](#) 

Collegamento al sito web del progetto:

- [Sito web "Industrial deployment of advanced system engineering methods for high productivity and dependability"](#) 

Altri collegamenti:

- [Sito web dell'Agenda digitale della Commissione europea](#) 

## **Progetti correlati**



ARCHIVED

## Industrial Deployment of Advanced System Engineering Methods for High Productivity and Dependability

DEPLOY

15 Luglio 2019

PROGETTO

### Questo articolo è contenuto in...

RIVISTA RESEARCH\*EU



**Le donne nella scienza - e la ricerca per migliorare la vita delle donne**

**Ultimo aggiornamento:** 28 Dicembre 2012

**Permalink:** <https://cordis.europa.eu/article/id/90012-feature-stories-keep-it-simple-bring-software-complexity-under-control/it>

European Union, 2025