



Kirkegaard Winkler (Kirkegaard)

FAST & FURIOUS

21 December 2011

"Se nella prima mezzora non capisci chi è il pollo, allora il pollo sei tu" - [Rounders](#)

Concludere un lavoro nei tempi previsti è diventata una vera e propria chimera all'interno dei processi industriali, un'utopica idea posseduta solo da tronfie entità aliene che, all'interno di queste destrutturate, sottocapitalizzate e sottodimensionate aziende private, allietano gli uffici amministrativi con colorati diagrammi di **Gantt**, **PERT** o **CPM**. Sovente le riunioni per la definizione degli incarichi avvengono in gioviali e sorridenti meeting, dove piacenti amministratori di joint venture aziendali si guardano come il lupo guardò cappuccetto rosso o come Bruto guardò Cesare, aspettando appunto, il momento buono per rifilarti una coltellata. Ora tu ti chiedi cosa avranno mai da essere così contenti, o meglio ti chiedi: perché loro sono contenti e a te girano le balle come una turbina elettrica ? Questa sorta di anomalia che purtroppo solo tu percepisci, deriva dal fatto che in quel meeting, ci sarà solo uno che lo prende nel ... e quel qualcuno sarai tu. Il bello è che tutti lo sanno, ne sono consapevoli, tranne te caro il mio ingegnerotto tecnico col pallino della programmazione.

Questi meeting hanno, in primo luogo, lo scopo di sfoggiare il più ampio ventaglio di ipocriti luoghi comuni, ma la cosa davvero bella, il clou della situazione, che accade sempre puntuale come un orologio verso la fine dell'incontro, è che tutti si fermano, come comandati da una voce superiore e, perfettamente sincronizzati, si girano verso di te. "Perché mi guardano ?" Ti chiedi un po' perplesso, ma non farai in tempo a riflettere perché da lì a poco arriverà la risposta. Ti guardano per vedere la tua faccia e osservare come si comporta uno mentre l'amministratore delegato gli chiede. "Bene veniamo al sodo: quanto tempo impieghi a terminare il progetto?"

La sera prima presi la macchina aziendale per partire presto ma quella mattina mi accorsi che nella borsa del pc mancava l'alimentatore e il piccolo portatile della ditta aveva la batteria scoppiata.

"Ciao Rud ... dobbiamo tornare in ditta perchè ho lasciato il cavo di alimentazione del portatile..."

"Si ti stavo chiamando per la stessa cosa... ho dimenticato l'interfaccia ottica..."

-Ho trovato chi è peggio di me- pensavo io, ma sicuramente lo pensava anche lui...

Raggiunsi la casa di RUD; erano le 4,30 della mattina visto che dovevamo ritornare in azienda per prendere il materiale mancante e poi di nuovo partire per raggiungere l'ente certificatore il più presto possibile.

Operativamente si doveva ancora ultimare la fase di download del firmware, necessaria per ricevere il certificato funzione che, ovviamente, ancora non era stata provata; meglio dire che la sera prima l'avevamo provata e non aveva funzionato. Colpa mia colpa sua, l'imperativo era mettere a posto i programmi in quei due giorni prima di eseguire la prova finale per la certificazione del prodotto.

La macchina correva veloce sopra quel viadotto; camion di materia oscura attraversavano regioni per raggiungere siti di deposito altrettanto oscuri. Abbiamo qualche centinaio di chilometri da percorrere, ma sapere di essere in ritardo prima di partire ha creato uno strano stato d'ansia per quella che sarebbe diventata una lunga trasferta contro il tempo. La frenesia era diventata una consuetudine giornaliera per una serie di situazioni contingenti extraziendali.

Ma veniamo al sistema embedded. Questa volta la scheda era composta con un doppio processore **STM32F103RBT6**, un classico **Cortex M3**.

La serie **Cortex** è una super serie che deriva dal processore **ARM** ad alte prestazioni prevista per il mercato embedded mobile. Ha il pregio di consumare meno rispetto al micro della serie tradizionale. La classe ARM era la classe forte dei sistemi embedded più pretenziosi essendo di natura a 32 bit e quindi idonea per gestire grosse masse di dati.

Noi però ci stavamo occupando di un apparato sviluppato per la misura dei consumi del gas e che veniva richiesto dalle nuove normative sulle linee di distribuzione e quindi sui gasdotti principali. Questo sistema viene chiamato correttore ed esegue, così come il nome lascia supporre, una correzione sulla lettura del gas consumato. In pratica al consumo volumetrico bisogna applicarci sopra una [formuletta](#) che tiene conto di diversi parametri tipo la temperatura, la pressione ma anche alcune costanti dipendenti dal tipo di gas (quello libico per esempio è molto diverso da quello russo e se non ricordo male di maggior pregio...). In questo caso non ero coinvolto nel sistema embedded ma nello sviluppo di una interfaccia software per la configurazione della macchina.

Ho scoperto la programmazione ad oggetti qualche anno fa e mi sono intrigato fra overloading ed eredità incominciando a implementare interfacce Windows. Dagli oggetti sono passato ai thread e così ho progettato sistemi ATE per il collaudo della linea di produzione dei nostri macchinari e altre interfacce per la configurazione degli stessi. I programmi windows venivano sviluppati con un vecchio visual studio 6.0 sfruttando le MFC. Parlando delle MFC con queste librerie, a suo tempo, ci si illudeva di bypassare alcuni aspetti profondi del kernel di windows come ad esempio la gestione delle code dei messaggi. Questo software sviluppato internamente per il nuovo correttore era come dicevamo una applicazione mfc dialog based, che aveva al suo interno diverse cartelle per suddividere i parametri.

Nel 1983 nacque il [c++](#) come espansione del c e fu una vera rivoluzione. Non era solo funzionale e quindi non aveva solo migliorato tutti gli aspetti della programmazione

tradizionale, ma era anche una rivoluzione strutturale nello sviluppo del software. I progettisti del compilatore fecero fare un deciso salto di qualità al C che i linguaggi concorrenti non seppero implementare (vedi l'agguerritissimo Pascal che non solo ha anche lui la **tipizzazione statica** ma ha anche il controllo del tipo).

Questo salto di qualità possiamo riassumerlo in un unico aspetto importante, che possiamo considerare l'elemento importante di un vero nuovo approccio alla programmazione. Questo nuovo aspetto nasce dalla creazione di una nuova struttura dati chiamata "oggetto".

Programmare ad oggetti necessita, come dicevamo, di un modo diverso di sviluppare la stesura di un software. I sistemi embedded a 8/16 bit che ho finora preso in considerazione però sono poco idonei per implementarci una programmazione ad oggetti, quindi direi che per questo tipo di sistemi il C standard ancora adesso è ampiamente sovrabbondante.

C'è una sorta di filo conduttore che lega le macro dell'**assembler** alle **librerie** (.lib) per passare alle **dll** e finire con gli oggetti **COM**. Mi sono convinto ci sia stato un processo di evoluzione nello sviluppo dei linguaggi di programmazione che è strettamente legato al riutilizzo del codice, ma mi spiegherò meglio in seguito.

Squilla il telefono è Mangusta.

"Pronto Rud ... sono Mangusta"

"Buon giorno Mangusta", siamo proprio sopra il viadotto appenninico il sole è ancora dietro le montagne, l'aria è tersa leggermente pungente.

"Si senti Rud dimmi riusciamo entro venerdì ad ottenere il certificato?"

Rud strabuzza gli occhi e pensa -mamma che idiota... gli ho già detto che non lo sapevo ieri sera ... -

"Non saprei Mangusta, perché molto dipende anche da loro .." (loro è l'ente certificatore ...)

Rud fa il cenno della pistola col dito e si spara sui maroni.

" Perché sapete... è abbastanza importante che otteniamo quel certificato, mi raccomando..."

Sapevamo, sapevamo... Sapevamo bene che la più grande azienda municipalizzata d'Italia aveva fatto il concorso per l'appalto degli apparati che dovevamo produrre (per circa una decina di milioni di euro...) e che abbiamo vinto sbaragliando la concorrenza. Il problemino era che ci siamo un po' aiutati dichiarando certificati metrici che purtroppo non esistevano e che appunto adesso dovevamo ottenere entro venerdì...

“Guardi stiamo facendo tutto il possibile... siamo partiti presto proprio per arrivare in sede dell’ente certificatore, così almeno ci organizziamo per questi giorni...”

“Bravi ragazzi ... ma a proposito che ora è ”

“Sono le 6:30 di mattina ingegnere ...”

Abbiamo superato il viadotto e un tir ci rallenta... Rud sfanala e chiudendo il cellulare sacramenta...

Nella [Teoria della comunicazione](#) si prevede che esistano 3 elementi fondamentali.

1. **Un trasmittente**
2. **Il mezzo**
3. **Un ricevente**

Questi elementi fisici necessitano di un quarto elemento non fisico che a me piace definire **protocollo**.

Il protocollo può essere complesso e quindi su più livelli e deve essere condiviso dal trasmittente e dal ricevente. Queste caratteristiche definite da [Shannon](#) (Claude Shannon 1916 2001), valgono per tutti i processi di comunicazione.

Ad esempio: quando Zeno scrive un articolo abbiamo un processo di comunicazione dove l'elemento trasmittente viene impersonificato da Admin Zeno, come mezzo il blog ElectroYou e come ricevente i vari lettori. Il protocollo è dato dalle regole grammaticali dell’italiano scritto che condivise dal trasmittente e dai riceventi permettono di garantire le finalità del processo di comunicazione.

Nel nostro caso Mangusta è il trasmittente, il mezzo è il telefono, Rud il ricevente. Il protocollo spesso è fatto da più livelli che diciamo si **incapsulano** uno dentro l'altro; così per continuare nell'esempio della telefonata, in uno dei livelli del protocollo si poteva dedurre che venerdì sarebbe stato l’ultimo giorno utile per ottenere il certificato, in caso contrario, evidentemente, sarebbe saltata la commessa principale e di conseguenza tutto il progetto e di conseguenza tutti gli investimenti quindi ancora una volta no produzione no stipendio.

La macchina correva veloce e nella prima mattinata raggiungemmo il grattacielo dell’ente certificatore.

Scendemmo prendendo tutto il materiale che non avevamo caricato in sede aziendale. Così in una ampia scatola di lavoro avevamo un paio di portatili per lo sviluppo del codice con tanto di alimentatori, interfacce Jtag per la programmazione e il debug dell’Arm, convertitori usb seriale, prolunghe seriali, perché non si sa mai, poi un paio di periferiche ufficiali che avrebbero dovuto essere usate per i test che l’ente certificatore avrebbe dovuto implementare. Tutto era accelerato, anche i tempi per riceverci ed era evidente il contrasto della nostra frenesia in quei mausolei burocratici dove le cose, si prendono con calma, con molta calma....

Appena arrivati mi guardo nelle tasche; il cellulare questa mattina l'ho tolto dal carica batterie spento. Così lo accendo, scrivo il pin e non faccio in tempo a riposarlo nella tasca quando sento la vibrazione e l'inconfondibile doppio squillo dell'sms.

Guardo è il feticchione di Phil che ieri sera mi aveva scritto il messaggio.

"Ciao Kirk. Ho un nuovo gioco. Prendi il cronometro e fallo partire. Adesso rispondi al seguente quesito prima che la lancetta arrivi a 30 secondi.

- Una mela pesa un etto più mezza mela. Quanto pesando due mele ? " Il cronometro era partito 1, 2, 3... il mio regno a chi mi toglie di impiccio.

Estratto da "<http://www.electroyou.it/mediawiki/index.php?title=UsersPages:Kirkegaard:fast-furious>"