

## SOFTWARE & RIVOLUZIONE

Aniello Cimitile

*Quando la filosofia dipinge il suo chiaroscuro, allora una figura della vita è diventata vecchia, e con il chiaroscuro essa non si lascia ringiovanire, ma soltanto conoscere; la nottola di Minerva comincia il suo volo soltanto sul far del crepuscolo.*

G.W. Friedrich Hegel, Prefazione a  
"Lineamenti di Filosofia del Diritto"

**Gennaro: Stètteme tre giorni senza mangiare e senza bévere nu surzo d'acqua, sette persone con due cadaveri sfracellati dalle schegge ... A nu certo punto...**

**Federico: Va buò, don Genna' .....**

Eduardo De Filippo, "Napoli Milionaria"



UNIVERSITÀ DEGLI STUDI  
DEL SANNIO Benevento

LECTIO  
MAGISTRALIS  
SOFTWARE E RIVOLUZIONE

*Prof.*  
*Aniello Cimitile*



Anche per me è arrivato il crepuscolo, che in verità già volge a sera. Parafrasando Hegel, credo che non solo per la grande storia, ma anche per una piccola storia individuale, questa sia l'ora in cui una personale nottola di Minerva comincia il suo volo. Dipinge il suo chiaroscuro offuscando e seppellendo per sempre fatti e vicende di una vita, evidenziandone o lasciandone altri in chiaro, nella speranza che possano essere utili per produrre conoscenza. Quando mi son chiesto cosa mai debba essere un'ultima lezione, ho pensato che possa risolversi nel seguire la nottola per qualche istante, osservando e raccontando.

Lo so, nel fare questo, come direbbe l'amico Bolognani, c'è sempre il rischio di soffrire la sindrome di Napoli Milionaria, di sentirti come il protagonista di questa commedia, Gennaro che, reduce dalla guerra, prova a raccontare esperienze drammatiche registrando intorno a se solo spazientito disinteresse e garbato ma fastidioso invito a parlar d'altro.

Luciano Canfora, riprendendolo dal Diario di Pietro Nenni, ha recentemente ricordato l'aneddoto secondo cui Mao Tse Tung, sul letto di morte, disse ai dirigenti del suo partito: *"Raccomandate ai giovani cinesi di ricordarsi di Yu Kung"*. E' un contadino che si era dedicato per tutta la vita all'incredibile impresa di spianare una montagna a colpi di zappa. A chi gli faceva notare che non ce l'avrebbe fatta mai, rispondeva: *"Io morirò ma rimarranno i miei figli. Moriranno i miei figli, ma resteranno i miei nipoti e così le generazioni si susseguiranno all'infinito. Le montagne sono alte ma non possono diventare ancora più alte. Ad ogni colpo di zappa esse diventeranno più basse"*.

La mia montagna didattica e scientifica si chiama "Software". Le generazioni si susseguiranno! Se penso all'Italia, io appartengo alla terza. La quarta è nel pieno della sua maturità e la quinta è già in campo. Generazioni che hanno prodotto ed inseguito uno sviluppo impetuoso e permanente di quella che appare come una tecnologia, una tecnologia inedita ed originale, perché il software è una tecnologia immateriale. Non si vede, non si tocca, non si sente, non ha odore, non ha sapore. La nostra conoscenza origina in ciò che i nostri sensi catturano e trasmettono, e, quindi, il software non si lascia conoscere direttamente. Abbiamo costruito un mondo fondato su pilastri di software, e non basta; il software invade ed occupa ogni spazio della nostra realtà, e non basta; il software evoca ed occupa nuovi spazi che vanno oltre la nostra realtà. In ogni momento della nostra vita, anche quando dormiamo, c'è del software che lavora per noi, costruisce, decide e risolve per noi, raccoglie ed elabora dati su di noi. Eppure dell'esistenza del software neppure ti accorgi, a volte anche quando difetta o fallisce.

La prima generazione è partita dalla idea che la parola software fosse solo un "prank", un modo di dire per indicare un insieme di programmi che venivano "forniti insieme ad un calcolatore" per agevolarne l'uso. Nell'Ingegneria accademica campana quella prima generazione era costituita da Bruno Fadini, capostipite e maestro indimenticabile, e da Carlo Savy. Ancora nell'edizione del 1970 del loro libro "Calcolatori Elettronici", su cui hanno cominciato i loro studi anche quelli della seconda e terza generazione, la parola Software è usata solo nelle pagine 32 e 33 e ridotta ad indicare poco più di un caricatore (loader), un assemblatore (un assembler, mentre un Compilatore veniva già considerato un programma aggiuntivo "di servizio") e un collegatore per la mappatura di eseguibili (linkage editor). Il resto era Programmazione, una attività di utenti, con qualche collezione di programmi e sottoprogrammi, ancora una volta di "corredo" ed utile a non rifare continuamente programmi frequenti o comunque già fatti. In sostanza, il software, era poco più di un vago vichiano sentire, concepito come ancillare e marginale complemento di un "calcolatore", la macchina meravigliosa, generale ed universale, che dominava la scena.

Anche nell'ambito industriale, la visione del software come componente secondario e servente accessorio di un computer era totalmente dominante. Il mitico ELEA 9003 era programmabile solo in linguaggio macchina, e cioè con ermetiche sequenze di 0 e di 1. Quella prima generazione industriale della ingegneria informatica che ebbe in Mario Tchou, prima guida e maestro, e in Giulio Sacerdoti i suoi esponenti di punta, capì molto

tardi che le loro macchine stupende (ELEA 9003 era un computer a soli transistor) non potevano competere senza un minimale corredo di Software. Il primo Assemblatore (PSICO) e il primo Compilatore (PALGO) ideati in Olivetti (a cui avevano cominciato a lavorare Mauro Pacelli e Marisa Bellisario) non videro mai la luce (una prima e provvisoria versione dell'Assemblatore fu portata a termine ma mai perfezionata). A parte la valutazione sulla scelta di puntare sul linguaggio Algol (Luigi Dadda, prima generazione accademica del Politecnico milanese, ricordò a più riprese che nessuno avrebbe comprato ELEA senza un compilatore Fortran), oggi c'è ampia convergenza nel ritenere che la causa primaria del dislivello fra Olivetti e i suoi competitors stava nella "scarsa sensibilità" del valore del software e sul conseguente scarso investimento in competenze e ricerca in tale settore.

Il ritardo nella comprensione del ruolo del software era in verità generalizzato, ma non c'è dubbio che in Italia fosse molto più accentuato. Il 7 Ottobre del 1968, a Garmish in Germania, spinta dall'enorme crescita del fabbisogno di software e dalla grande difficoltà quantitativa e qualitativa di rispondere a tale domanda, la NATO sponsorizzò la prima conferenza sulla Ingegneria del Software. Fu Fritz Bauer, in tale sede, a darne la prima definizione e ad indicarne le prime sfide. A questa conferenza non c'erano italiani. La seconda si tenne a Roma ed era presente Paolo Ercoli (prima generazione accademica, Università di Roma – Sapienza), inizialmente fortemente spinto anche da una sorta di "dovere di ospitalità".

A determinare la rottura della visione ancillare del software come componente marginale e inscindibilmente legata al suo computer, fu la seconda generazione. A Napoli fu Lucio Sansone, negli anni '70, a indicarci la strada della separazione concettuale, logica ed operativa del software dal computer e con lui ci apriamo ad una visione che non si limitava al mondo dei "linguaggi e compilatori" e neppure a quella dei Sistemi Operativi, ma includeva l'universo in espansione delle applicazioni. Fu con lui che seguimmo la rivoluzione della programmazione strutturata (Dijkstra e Wirth), che affrontammo i capisaldi dell'astrazione e dei livelli di astrazione (Barbara Liskov), della modularizzazione e dell'Information Hiding (Parnas); e fu sempre con lui che ci apriamo alla nuova visione dei dati con il modello E-R ed i Data Base relazionali (Codd e Chen). Fu un lungo, intenso ed entusiasmante cammino che ci fece capire che il Software non era solo flow-chart ed istruzioni, ma era qualcosa che richiedeva un suo autonomo patrimonio di teorie, metodi, modelli e tecniche. Sul Software, dal vichiano sentire, si passava al vichiano avvertire ("con animo perturbato e commosso") e finalmente al vichiano riflettere ("con mente pura"). La zappa di Yu Kung affondò i suoi colpi sul grande capitolo della Progettazione del Software dove, per affrontarne la complessità, cominciò l'impetuoso susseguirsi di metodi e modelli (Jackson, De Marco, Yourdon & Costantine) che non conosce un traguardo finale.

La terza generazione è stata quella che ha allontanato definitivamente il baricentro del software dai computer (software di base e di sistema), portandolo negli ambienti reali di produzione di beni e servizi (software applicativi). Anche l'importante corredo di software che accompagnava i computer cominciò a diventare sempre più vasto e complesso, "meno proprietario" e ricco di nuove sfide, e tuttavia la grande massa del software si spostò nei domini applicativi, con una domanda che ormai assumeva caratteristiche e dimensioni di livello industriale. E' con la terza generazione che arrivò l'approdo alla Ingegneria del Software, ai Cicli di Produzione ed ai Cicli di Vita del Software, ai complessi e numerosi processi che essi richiedono e che riconducono "l'antica e dominante Programmazione" ad essere una delle fasi o delle attività più tecniche ed a minor valore. Fra i napoletani di quella terza generazione si costituì un gruppo (oltre a me, c'erano Ugo De Carlini e Gianni Cantone, mentre altri, come Nino Mazzeo, si dedicarono a settori diversi) che spinse con decisione in questa direzione, valorizzando i processi di Manutenzione e Testing, la definizione e la progettazione di metodi, modelli e strumenti software per l'analisi, la reingegnerizzazione ed il riuso di software. Fu anche questa generazione a partecipare prima alla formazione di una comunità nazionale della IS (anche grazie alla spinta di un Progetto Finalizzato CNR dedicato al software) e poi ad attuare un processo di internazionalizzazione (che vide l'approdo in Campania di primarie conferenze IEEE e l'arrivo, per la prima volta in Italia, della IEEE Conference on Software Engineering, massima conferenza mondiale del settore, la

cui 16<sup>a</sup> edizione si tenne a Sorrento nel 1994). E' questa generazione che prese atto che la scena era ormai popolata e sarebbe stata dominata da Software che si collocano fra i sistemi più complessi che l'uomo abbia mai costruito, fatti di centinaia di milioni di linee di codice (...altro che il centinaio scarso di istruzioni dei loader di prima generazione). E' questa generazione che, di fronte alla esplosione della domanda di software, si proiettò verso visioni industriali e processi da Fabbrica del Software (Vic Basili), e approdò alla Programmazione e alle metodologie di Analisi e Progettazione ad Oggetti (Booch, Objectory di Jacobson, OMT di Rumbaugh) che confluirono poi nella proposta del "linguaggio di modellazione unificante UML.

Il Software aveva ora una sua completa autonomia; il processo di "liberazione" dal suo vecchio padrone, il computer, si era completato e il sistema di elaborazione dati che lo eseguiva era tutt'al più equiparabile ad uno dei tanti stakeholder e delle tante entità a cui il software deve conformarsi. E' stata una rivoluzione frutto di un cambiamento continuo del software, un cambiamento che va ben oltre la legge di Belady e Lehman che si applica a sistemi e componenti software. Insieme alla complessità, l'invisibilità, la conformità, all'essere discreto, la "changeability" è una delle proprietà essenziali del software che quindi "o cambia o muore". Ma i cambiamenti a cui voglio ora riferirmi sono quelli "rivoluzionari" che vedono il software essere insieme soggetto ed oggetto di una metamorfosi più generale ed importante. Ogni generazione nasce e si forma lavorando con le due che la precedono, poi raggiunge il vertice e corre sul ramo discendente della propria parabola lavorando con le due che la seguono. Io credo che la prima e la seconda generazione, anche con i suoi esponenti più visionari, non abbiano capito né potevano intuire le rivoluzioni in cui sarebbero state immerse la terza e la quarta generazione e la nuova realtà in cui la quinta si troverà a compiere il suo cammino. Da noi, la quarta generazione accademica nata dall'Ingegneria napoletana non è più solo napoletana: penso a Gerardo Canfora ed Umberto Villano a Benevento; a Mario Vento, Andrea De Lucia e Angelo Marcelli a Salerno; a Beniamino De Martino a Caserta; a Giorgio Ventre, Nicola Mazzocca, Stefano Russo e Angelo Chianese a Napoli. E' con questa generazione (in particolare con Gerardo Canfora e, per un lungo tratto, con Andrea De Lucia), che ho vissuto, spesso anche senza averne coscienza, la nascita, la convergenza e la indissolubile congiunzione di Software&Rivoluzione.

Il 2018, che si è appena concluso, oltre a segnare il 50° anniversario della nascita della IS, segna anche quello del noto movimento sociale che aveva l'ambizione di cambiare il mondo. So che l'essere stato un "Sessantottino" ed un attivo partecipante del "Movimento Studentesco", porta a pensare che cambiamento e rivoluzione siano parte del mio DNA, e che quindi io possa essere portatore di una prevenuta tendenza ad usarli come categorie interpretative in ogni "ministra". Proprio per questo, vale la pena di introdurre qualche preliminare chiarimento. Credo che il termine "rivoluzione" sia fra i più complessi, i più abusati e conseguentemente più polisemico dei nostri tempi. A seguire la retorica politica, mi perderei nel contare quante rivoluzioni al giorno vengono annunciate o fatte; quanto alla pubblicità, non c'è prodotto che non ci rivoluzioni l'esistenza. Ovviamente, non è in questa direzione che mi muovo, né in quella delle Rivoluzioni politiche, come la Rivoluzione Francese o la Rivoluzione Russa, a cui Hanna Arendt associa la proprietà essenziale di essere puntuali nel quadrante del tempo, veloci e violente. Mi riferisco invece a rivoluzioni legate all'attività primaria e dominante dell'uomo e/o a quelle scientifiche e tecnologiche; rivoluzioni più lente e a cadenza plurisecolari le prime, forse meno lente e a cadenze in accelerazione crescente le seconde.

Per quanto riguarda la prima tipologia mi riferisco innanzitutto alla Rivoluzione dell'Agricoltura, che a partire dall'inizio del primo millennio sviluppò la Società Contadina (o Rurale); poi alla Rivoluzione Industriale che a partire dalla seconda metà del 18° secolo ci portò nella Società Industriale; ed infine alla Rivoluzione della Informazione, che a partire dal volgere del terzo Millennio ci porta in quella che Manuel Castells, nella trilogia sull'età dell'informazione, ha chiamato Società Informazionale. La società industriale ebbe come suo motore propulsivo l'energia, la meccanica e le tecnologie ad esse associate (un noto rivoluzionario russo all'inizio del 20° secolo coniò l'equazione sviluppo= energia + soviet); la società informazionale ha come suo motore un paradigma scientifico e tecnologico fondato sull'informazione e sulle tecnologie dell'informazione, dall'elettronica alle telecomunicazioni e all'informatica (diremmo oggi: sviluppo= informazione +

democrazia). Non mi soffermo sugli elementi emergenti e sensibili di uso quotidiano prodotti da questo paradigma (come la rete, il web, il cellulare, etc.); e mi limito solo a segnalare (o ricordare) il ruolo straordinario di componenti strutturali che nella società dell'informazione hanno assunto i dati e la conoscenza (i primi legati ai valori delle informazioni note e la seconda legata alla loro elaborazione ed alla produzione di nuove informazioni). Respiriamo dati e siamo quotidianamente sommersi da uno tsunami di nuovi dati che vengono prodotti, memorizzati, trasmessi e usati in quantità e velocità strabilianti se paragonate al passato: bit, byte, Kbyte e Kbit/sec della prima generazione, impallidiscono rispetto agli zettabyte ( $10^{21}$  byte), ai brontobyte ( $10^{27}$  byte) e ai Mega, ai Giga e ai Tera ( $10^{12}$ ) bit/sec della quinta generazione. Ed è strabiliante, rispetto al passato, sia per quantità che per velocità la produzione e l'uso di conoscenza. Come parte della punta di questo iceberg e solo come esempio, lasciatemi segnalare che sono più di 45 milioni il numero di articoli scientifici (citabili e raggiungibili) pubblicati in un anno (2017), centinaia di migliaia i brevetti rilasciati (circa centomila solo in Europa nel 2016), e più di 80000 all'anno le idee innovative che hanno concretamente prodotto cambiamenti e applicazioni nella produzione di beni e servizi. Fra i cambiamenti strutturali che la rivoluzione informazionale ha introdotto assume sicuramente grande rilevanza quello della Metamorfosi del Modo di Produzione.

Dal modo di produzione industriale centrato sulla valorizzazione di grandi masse di capitale fisso materiale siamo passati rapidamente ad un modo di produzione centrato sulla valorizzazione di capitale immateriale (capitale umano, capitale conoscenza, capitale intelligenza). Nella società informazionale, il valore aggiunto è prodotto non dal "lavoro materiale" ma dalla "conoscenza" che è cristallizzata in un prodotto o in un servizio. Insieme a questo si è sviluppato anche un profondo processo di Metamorfosi del Lavoro. Come direbbe André Gorz, dalla centralità del lavoro materiale e semplice siamo passati a quella del lavoro immateriale e complesso. Benchè il primo resti indispensabile e, sia pure in misura minore e calante, ancora prevalente, il cuore della creazione del valore è il lavoro immateriale. Come ci racconta l'Economia della Conoscenza, penso a Rullani, ora "tutti gli oggetti e tutti i servizi sono ad alto contenuto di conoscenza, ed il lavoro dominante è quello cognitivo". In questo trionfo dell'immateriale il Software assume addirittura un ruolo strategico perché è la Tecnologia dell'Immateriale. Non solo è una tecnologia ICT, ma è quella che più di ogni altra, in modo straordinariamente flessibile e veloce, consente di incapsulare e cristallizzare nei prodotti lavoro cognitivo e conoscenza. In effetti, ogni prodotto software, dalla più semplice sequenza di istruzioni al più complesso sistema software, è esso stesso pura conoscenza, ovvero conoscenza oggettivata, quella che Michael Polanyi chiamerebbe conoscenza esplicita, con la caratteristica di essere immediatamente usabile, riproducibile e trasferibile, adattabile ed integrabile. La conclusione è che, come tecnologia, il Software è un componente essenziale, ovvero caratterizzante e strutturale della Società dell'Informazione.

Quanto ai dati, una diffusa affermazione in stile twitter, molto popolare anche fra i politici, racconta che i dati sono il petrolio del nostro tempo. Restando nell'ambito di questa metafora, diciamo che sono le ICT a trasformare questo petrolio in potenziale ricchezza. Ad esempio, è l'elettronica a consentire intercettazione e emissione di dati; sono le telecomunicazioni a consentirne la trasmissione e, soprattutto, è il software che ne consente l'organizzazione e la gestione in "bacini di dati", il loro raffinamento, la loro trasformazione in informazioni, la loro elaborazione. Uscendo dalla metafora, il petrolio si forma e si accumula naturalmente e in tempi biblici rispetto a quelli del suo uso, uso che ne implica la distruzione, per cui è una risorsa destinata all'esaurimento. Nella società dell'informazione la produzione dei dati è come abbiamo detto, uno tsunami permanente e crescente, non siamo di fronte ad una risorsa esauribile, peraltro, il loro uso non implica la loro distruzione. La stragrande maggioranza di questi dati non si accumula naturalmente e di essa solo una parte modesta viene raccolta e conservata, il resto si perde. Se si chiede alle ICT quali di questi dati possiamo catturare, organizzare, gestire ed usare, la risposta è che lo possiamo fare per qualsiasi dato; se si chiede se è possibile farlo per tutti, la risposta è no, non avremo mai il software necessario per farlo perché i dati sono un universo che già ci appare come infinito ed è pure in espansione. Ciò ci dice, tuttavia, quale è la dimensione della domanda di software che viene dal mondo dei dati e ci lascia capire anche che Big Data e associati

Analytics sono solo nell'infanzia del loro cammino e, forse, come accaduto con le Basi di dati, sono solo una tappa di fase della gestione ed elaborazione di dati nell'evoluzione della società informazionale. Venti anni fa, nella relazione invitata tenuta in occasione dell'inaugurazione dell'anno accademico della nostra università, ebbi a dire che era tecnicamente possibile costruire il software necessario per l'attuazione di Echelon, ovvero la costruzione del sistema di intelligence congiunto di 5 paesi per la intercettazione delle comunicazioni pubbliche e private. Echelon appariva allora come un fatto inquietante, pericoloso e per taluni un inverosimile ipotesi ai confini della fantascienza. Oggi, non solo sappiamo che allora quella valutazione di fattibilità era corretta, ma che ora la intercettazione, la collezione e l'uso di dati privati e pubblici è non solo possibile ma anche largamente diffusa e potenzialmente estendibile. Inoltre, è palesamente infondata quella affermazione secondo cui, a differenza del petrolio, i dati non inquinano. E' un inquinamento diverso fatto di fakes e dati falsi. Oggi il problema più scottante che abbiamo di fronte non è quello della intercettazione e collezione di dati, ma è caso mai quello della loro sicurezza e difesa da accessi indesiderati (dolosi o non), a cui si aggiunge il grande problema della quarta V, ovvero della Veridicità dei dati (che si unisce a quello del Volume, della Velocità, della Varietà). Una nuova sfida per le ICT ed in modo particolare per il software, che trova e continuerà a trovare soluzioni per sistemi ed assistenti ausiliari per la sicurezza ed il filtraggio di dati.

Credo, tuttavia che quanto detto, sia ancora insufficiente per metter a fuoco il ruolo del software e le sfide che deve affrontare nella rivoluzione in corso. Da tale punto di vista aiuta la visione proposta da Luciano Floridi nel primo contributo in assoluto che la filosofia ci ha dato sull'informazione. Secondo tale visione, questa nostra società si caratterizza per la nascita di un nuovo ambiente informazionale che le ICT stanno creando ed espandendo: è una Infosfera nella quale le generazioni future trascorreranno la maggior parte del loro tempo (negli USA, già ora, il tempo medio speso dall'uomo nell'infosfera ha superato i 5 mesi all'anno). L'infosfera è definibile come una rete di attori e di oggetti informazionali, gli InfoOrg, che sono sorgenti e pozzi di informazione e/o conoscenza. Noi siamo una parte di questi Inforg, di cui fanno parte tanti altri InfOrg di tipo biologico e naturale; ma la loro maggioranza è costituita da IT-entity, ovvero entità che incorporano o sono fatte da tecnologie dell'informazione. Nell'Infosfera, la rete consente di sviluppare una quantità teoricamente infinita di processi informativi fondati su collegamenti A2A (anything to anything), ovvero connessione di ogni InfoOrg a qualsiasi altro, e A4A (anywhere for anytime), ovvero collegamenti possibili in ogni luogo ed in ogni momento. Nella Società Informazionale l'espansione dell'Infosfera è tale che presto tutto il mondo materiale, tutto il mondo reale, sarà parte dell'Infosfera, che non sarà il mondo dell'Informazione ma la realtà. Quando questa migrazione nella nuova realtà sarà completata, essere staccati dall' Infosfera sarà come vivere da pesci fuor d'acqua o in sopravvissute tribù primitive dimenticate e disperse in qualche foresta (ovvero, pezzi di una vecchia e scomparsa realtà che, per isolamento, sono rimasti tali). Di fronte a ciò, il software, come tecnologia dell'immateriale, assume l'enorme ruolo di tecnologia portante della costruzione ed espansione dell'infosfera, dove, poi, dovrà invadere i suoi nuovi, inediti ed imprevedibili spazi.

Dobbiamo aggiungere che nella società informazionale e nella sua infosfera si sta realizzando un'altra rivoluzione scientifica, una rivoluzione appartenente alla classe della rivoluzione copernicana, di quella darwiniana e così via. Come la terra non è più il centro del sistema solare intorno a cui ruota tutto il resto, così oggi l'intelligenza umana non è più sola sulla terra e non è sulla sua centralità che si sviluppano attività e relazioni. L'insieme dei beni, dei servizi, delle relazioni e di ogni altra attività che l'uomo produce non è più frutto unicamente della sua intelligenza ma anche di un insieme di altre e diverse intelligenze che si integrano, sommandosi e moltiplicandosi, interagendo e cooperando fra di loro. Nell'Infosfera nascono e proliferano IT-entity che selezionano, valutano, trasformano e producono dati e conoscenza, che hanno capacità di apprendimento ed intelligenza, capacità di fornire feed-forward (non solo i tradizionali feed-back) e assumere decisioni e comportamenti su eventi futuri. Come accade nell'uomo e come direbbe Gardner, è una intelligenza fatta da molteplici tipi di intelligenza; non solo Intelligenza logico-matematica, ma anche spaziale (come dimostrano, ad esempio, realtà virtuale e aumentata), cinestetica (come dimostrano ad

esempio robot e robot cooperativi, stampanti 3D), o interrelazionale (capacità di interfacciamento, collegamento e relazione con altri InfoOrg, a cominciare da uomini e donne) e così via. Appare evidente che tutta questa intelligenza artificiale è fornita alle IT-entity quasi totalmente dal software. Nell'infosfera non esisterebbe intelligenza artificiale senza software, che ha dunque il compito di fornire tutta la IA dell'infosfera che sarà necessaria oggi e in tutta la evoluzione della società informazionale.

In questo quadro, l'Ingegneria del Software ha evidentemente davanti a sé nuove ed enormi sfide da affrontare. Non sono solo le storiche sfide quantitative e qualitative. Per quanto riguarda la quantità di software necessaria, dobbiamo produrne per miliardi di macchine, miliardi di utenti, milioni e milioni di applicazioni. Mi viene in mente quello che Butler Lampson raccontò nel 2001, quando la IEEE gli assegnò la medaglia Von Newman, e cioè che da più di 25 anni egli aggiornava e mostrava uno stesso grafico nel quale il fabbisogno di sviluppatori software superava l'ammontare della popolazione mondiale. Oggi la punta dell'iceberg degli sviluppatori software è costituita da 21 Milioni di addetti; se ipotizziamo grossolanamente che ciascuno di essi produca mediamente 10 istruzioni al giorno, se ne deduce che vengono prodotte più di 70 Miliardi di istruzioni all'anno (naturalmente non tutto si trasforma in prodotti di mercato o free software, e bisogna ricordare che quelli che chiamiamo "gli addetti" sono solo una parte di quelli che producono software); e tuttavia più se ne produce e più cresce il suo fabbisogno che, quindi, nell'Infosfera e nella sua espansione, appare praticamente come infinito. Per quanto riguarda la qualità, restano le battaglie di sempre (come la sfida dei tempi e dei costi, la capacità di rispondere a richieste ed aspettative, la presenza di errori, e così via); oggi come ieri, resta valida la vecchia lezione di Fred Brooks: "there is no silver bullet". Forse, oggi più che mai, crescono l'attenzione e le domande sui comportamenti imprevedibili del software; ora che si moltiplicano IT-entity intelligenti sono domande che nascono da inquietudini e paura dell'autonomia dell'automa. Noi sappiamo, da un punto di vista teorico e generale, come il problema della individuazione di tutte le possibili esecuzioni di un programma e del testing esaustivo dello stesso sia un problema indecidibile. Sappiamo anche l'effetto di esplosione combinatoria di tali possibili esecuzioni quando, come accade sempre più nella "programmazione odierna", un software è costruito sulla integrazione di altri software (o una nuova IT-entity è costruita con l'integrazione di altre IT-entity). Sappiamo, ad esempio, che quelle dieci istruzioni giornaliere di uno sviluppatore possono produrre migliaia e migliaia di possibili esecuzioni, non tutte volute o controllate. E' vero, dunque, che il software contiene errori e può produrre comportamenti inattesi, come abbiamo sperimentato in questi 50 anni, anche con clamorosi fallimenti. E' vero quindi che il software può produrre comportamenti inattesi nelle IT-entity, nella rete dell'Infosfera, nei sistemi cyber-fisici che abbiamo e che verranno. Sappiamo, però, che proprio dove si annidano indecidibilità teoriche l'Ingegneria si esalta costruendo e garantendo, nella concretezza e nei vincoli della specifica applicazione, livelli di affidabilità, garanzia, prevenzione, recovery. E' quello che sta accadendo e continuerà ad accadere anche nel futuro e nell'Infosfera. Non mancheranno errori e "failures", ma non si verificheranno scenari apocalittici, di impazzimento delle intelligenze artificiali, né è, allo stato, possibile o prevedibile, una loro incontrollata contrapposizione alle intelligenze umane (nessuna di tali contrapposizioni è possibile se non diretta rappresentazione o deriva delle contrapposizioni uomo-uomo). E' vero, però, che con riferimento ai nuovi scenari, la IS dovrà arricchire e potenziare attività di testing e soluzioni di reliability che, per quanto riguarda i comportamenti terminali, si troveranno di fronte a problemi inediti e originali (si pensi ai sistemi ad autoapprendimento, destinati ad arricchirsi di comportamenti non noti a priori). Ma più in generale la IS si troverà ad affrontare una nuova fase di profondo rinnovamento che non è riconducibile unicamente alle tecniche, ma riguarda teorie, metodi e modelli. Ad esempio, anche i più recenti cicli di produzione fondati su metodi agili appaiono palesemente insufficienti per la maggioranza delle esigenze della infosfera, dove, ad esempio, le fasi di definizione del problema (elicitazione e specifica dei requisiti) e di progettazione assumono complessità sistemiche e di integrazione sistemica non riconducibili al rapporto "sviluppatore-utente/proprietario". L'anno scorso, nel 50° anniversario della IS, Grady Booch ci ha ricordato come, nei grandi cambiamenti che la IS subirà, resteranno fermi ed irrinunciabili alcuni principi e primo far tutti, il Principio della Astrazione. Ebbene, è proprio nel bagaglio delle astrazioni che la IS dovrà rinnovarsi

profondamente, sia nei processi di produzione delle astrazioni che nei componenti di una astrazione. Come direbbe Floridi, nella transizione dalla realtà industriale alla infosfera, ad ogni livello di astrazione usato, sono cambiati e cambiano “gli osservabili”, ovvero le entità concettuali che caratterizzano e definiscono una astrazione. E’ un fenomeno che coinvolge ogni livello della clessidra delle astrazione della IS, dal primo livello (quello dei dettagli della infosfera) alla strozzatura della definizione dettagliata del sistema software (progettazione esecutiva LLD), ed infine all’ultimo livello (quello dei dettagli della IT-entity che eseguirà il software o in cui lo stesso sarà incapsulato). Un profondo rinnovamento dovrà poi riguardare la strumentazione, ovvero i tools da usare nei processi software. Cito, a puro titolo di esempio, che anche qui, come in tutta la IS, è ormai indispensabile e perfino urgente importare ed usare metodi e tecniche di IA; anche nei nostri processi abbiamo sempre più bisogno di tools e assistenti sempre più intelligenti e sempre più capaci di incorporare e rendere fruibile la conoscenza esplicita e quindi oggettivabile. Profondamente mutata è anche la più vecchia attività dei processi software, ovvero la programmazione; non solo perché la criticità di questa fase si è fortemente ridotta rispetto ad altre, ma anche perché non dobbiamo più programmare computer ma IT-entity e sistemi cyber-fisici. Qui, come in tutte le fasi, l’Ingegneria del software deve proporre nuovi approcci sistematici fondati su una spina dorsale concettuale in cui tutti gli artifatti della IS e le IT-entity (con i loro software embedded) sono integrati ed integrabili, relazionabili, adattabili, modificabili, migliorabili ed evolvibili. Abbiamo bisogno dei concetti di questa spina dorsale, delle astrazioni e dei metodi che debbono caratterizzarla.

In questa divenire della Società dell’Informazione, il software ha davanti a se ruolo e sfide importanti. Ma il software da solo non basta. Ha sicuramente bisogno della Ingegneria del Software per dominare la definizione, la costruzione e la tenuta in esercizio di sistemi complessi, per definire approcci strutturati e sistematici ai processi software, per costruire gli ambienti di lavoro intelligenti (reti e IT-entity) in cui sviluppare tali processi. Ha sicuramente bisogno del contributo che hanno dato e daranno le Scienze del software e dei dati. Appare ovvio e scontato che ha bisogno di tutte le altre ingegnerie delle ICT (dall’elettronica alle telecomunicazioni, dalle misure alla automatica e alla robotica) con le quali condivide la responsabilità della costruzione della infosfera. E’ sempre più evidente che ha bisogno di tutte le Ingegnerie e di tutte le Scienze Applicate con le quali deve integrarsi per la transizione di tutta la realtà attuale nella infosfera. In questo momento il lancio dei programmi di Manifattura 4.0 (o di I4.0) nei paesi europei ed in Italia ne danno una plastica e concreta evidenza. Qualcuno ha detto che questi programmi segnano la rivincita del materiale e del lavoro materiale; credo invece che sia esattamente il contrario, perché siamo di fronte alla invasione dell’Immateriale nelle produzioni di beni materiali. E’ il trascinarsi dei processi industriali e manifatturieri nell’Infosfera, la disseminazione in tali processi, dalla logistica alla manutenzione, di reti ed IT-entity, dove anche le macchine utensili diventano entità in rete oltre che basate su sistemi informatici di controllo e gestione remota. Stiamo sperimentando qui che I4.0 significa innanzitutto integrazione di discipline e competenze diverse perché qualsiasi problema richiede approcci e soluzioni fondate sulla fusione di software, ICT, tecnologie e conoscenze del dominio applicativo. Dovunque, non solo nell’industria (penso all’agricoltura, all’ambiente, e a ogni cosa) il trascinarsi-assorbimento nell’Infosfera implica questo tipo di integrazione. Ma tuttavia tutto questo ancora non basta. Concordo profondamente con quanti sostengono che la società informazionale è cresciuta molto più rapidamente della capacità dell’uomo di svilupparne solide radici concettuali, etiche e culturali. Credo che anche in questa breve relazione di un informatico, emerga con chiarezza il grande fabbisogno di economia (da quella politica a quella gestionale ed organizzativa), di sociologia, e, su tutto, di Filosofia. Il software e le altre tecnologie sono strumenti per realizzare una infosfera che deve essere definita dalle scienze della prassi. Abbiamo bisogno di una metafisica dell’infosfera, di passare da una metafisica materialistica ad una metafisica informazionale. Abbiamo bisogno di categorie e costrutti filosofici che ci consentano di capire ed interpretare l’infosfera, il suo divenire e ciò che in essa accade. Abbiamo bisogno di capire cos’è un uomo “on line”, che non è e non può vivere “off line”, parola che potrebbe presto diventare incomprensibile a livello di massa. Appare incredibile che siamo dentro la società dell’informazione e il primo contributo per una filosofia

dell'informazione sia arrivato, grazie a Luciano Floridi, solo nel 2010; o, ancora, che solo con questo primo contributo sia arrivato la prima moderna riflessione filosofica sul concetto di astrazione che ovviamente è fondamento centrale dell'immateriale. Ma abbiamo, più in generale, bisogno di una Ecologia dell'Infosfera, vale a dire Etica, Diritto, Cultura ...E' tutto questo che definirà cosa il software dovrà far fare alle IT-entity, è tutto questo che dovrà definire le sfide delle Ingegnerie. Perché se è vero che il Software non è più servo di un computer o di altre tecnologie, è pur sempre una tecnologia ...Un anno fa, in occasione di una mia partecipazione alla presentazione di un suo libro, Guido Trombetti mi chiese se potevo rendere creativo il software (e quindi le entità che lo incorporano); gli risposi che avrei potuto farlo non appena lui mi avesse fornito la matematica della creatività.

Ho scelto, in questa breve relazione, di dare spazio alla principale percezione che si ha del software, ovvero quella di una tecnologia. Naturalmente il software è anche un fondamentale comparto produttivo, un relevantissima area scientifico disciplinare, un settore strategico della formazione e della didattica. Non ho spazio per parlarvi di tutto questo; eppure vorrei parlarvi delle storie dei ritardi e degli errori del sistema produttivo italiano e ancora di più dei ritardi e degli errori della politica, incapace di concepire adeguate politiche industriali e interventi pubblici per definire e sostenere un autonomo e stabile ruolo italiano ed europeo nella produzione di software.

Voglio, invece, avviarmi alla conclusione osservando che nel mondo software, e forse non solo, fra le metamorfosi in corso, ci sia, come direbbe Pekka Himanen, la transizione dall'etica protestante del lavoro all'etica hacker del lavoro dove generare software è un'attività appassionata e libera dagli schemi. Io non so dire se e quando questa transizione sarà totalmente completata, o se, ad esempio, si stabilizzerà un equilibrio di coesistenza. So che sicuramente nel nostro mondo accademico l'etica dell'hacker deve essere dominante, che il nostro lavoro deve essere fondato sulla passione e sulla scelta di farlo per il piacere di farlo, che lavorare al software significa ricondurre nel lavoro componenti caratteristiche del tempo libero, come il divertimento. So che nell'Università viviamo un inverno duro e difficile, fatto di carenze ed addirittura di diminuzione delle risorse, di lacci e laccioli, di incrostazioni burocratiche che offuscano o falsificano missione e veri obiettivi, e perfino di mortificazione delle persone e quindi dell'irrinunciabile e strategico lavoro cognitivo che in essa si svolge. Ma è proprio e più che mai in queste fasi, che solo l'etica dell'hacker ci può portare in primavera. Grady Booch ha recentemente detto che noi "softwaristi" siamo "storytellers", inventori e sceneggiatori di storie, narratori e cantastorie, e ci ha ricordato che: "software is the invisible writing that whispers the stories of possibility to our hardware". Il software non solo non è più "schiavo del computer" ma è quello che decide ciò che può fare un computer o l'entità in cui è incorporato. Io credo che sia un lavoro esaltante, di grande privilegio e responsabilità, dare senso e storie a cose, spesso complesse e meravigliose, che senza software quel senso e quelle storie non avrebbero. Nella società dell'informazione, nel dispiegarsi dell'Infosfera, nella rivoluzione dell'intelligenza, noi siamo più che mai cercatori di futuro, raccoglitori, creatori, narratori e scrittori di storie di vita che a loro volta cambieranno la vita delle generazioni a venire.

